IMPLEMENTASI METODE *BOOSTING* UNTUK PREDIKSI JENIS TANAMAN BERDASARKAN KONDISI TANAH

LAPORAN TUGAS AKHIR

Laporan Ini Disusun Untuk Memenuhi Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri

Disusun Oleh:

RAFIF HUDA ADITYA

NIM 32602100109

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025


UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

IMPLEMENTATION OF THE BOOSTING METHOD FOR PREDICTING PLANT TYPES BASED ON SOIL CONDITIONS

FINAL PROJECT

Proposed to complete the requirement to obtain a bachelor's degree (S1) at Informatics Engineering Departement of Industrial Technology Faculty Sultan Agung Islamic University

Arrange by :

RAFIF HUDA ADITYA

32602100109

MAJORING OF INFORMATICS ENGINEERING

INDUSTRIAL TECHNOLOGY FACULTY

SULTAN AGUNG ISLAMIC UNIVERSITY

SEMARANG

2025


UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

LEMBAR PENGESAHAN

TUGAS AKHIR

IMPLEMENTASI METODE *BOOSTING* UNTUK PREDIKSI JENIS TANAMAN BERDASARKAN KONDISI TANAH

RAFIF HUDA ADITYA
NIM 32602100109

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 12 Agustus 2025

TIM PENGUJI UJIAN SARJANA :

Imam M.I.S, S.T, M.Sc, PhD	.....	27-8-2025
NIK. 210600017	.....	.....
(Penguji 1)		
Badie'ah, S.T, M.Kom	.....	20-8-2025
NIK. 210615044	.....	.....
(Penguji 2)		
Ghufron, S.T, M.kom	.....	25.8.2025
NIK. 210622056	.....	.....
(Pembimbing)		

Semarang, 27 Agustus 2025
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, S.T, M.IT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Rafif Huda Aditya

NIM : 32602100109

Judul Tugas Akhir : IMPLEMENTASI METODE *BOOSTING* UNTUK PREDIKSI JENIS TANAMAN BERDASARKAN KONDISI TANAH.

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 1/9/2025

...takan,

Rafif Huda Aditya

SURAT PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: Rafif Huda Aditya
NIM	: 32602100109
Program Studi	: Teknik Informatika
Fakultas	: Teknologi industri
Alamat Asal	: Jl. Merpati Kranggan, Kec. Pati, Kab. Pati

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : ***IMPLEMENTASI METODE BOOSTING UNTUK PREDIKSI JENIS TANAMAN BERDASARKAN KONDISI TANAH.***

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiatisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

KATA PENGANTAR

Dengan mengucap syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Implementasi Metode *Boosting* Untuk Prediksi Jenis Tanaman Berdasarkan Kondisi Tanah" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Ghufron, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu. Serta memberikan banyak nasehat dan saran.
4. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini.
5. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini dan masa mendatang.

Semarang, 1/9/2025

Rafif Huda Aditya

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Pertanian merupakan sektor strategis dalam mendukung ketahanan pangan dan pembangunan berkelanjutan, khususnya terkait SDG 2 (*Zero Hunger*) dan SDG 9 (*Industry, Innovation, and Infrastructure*). Tantangan global seperti degradasi tanah dan perubahan iklim menuntut adanya inovasi berbasis teknologi. Penelitian ini bertujuan mengembangkan sistem prediksi jenis tanaman berdasarkan parameter tanah (N, P, K, suhu, kelembaban, pH) dengan membandingkan algoritma Gradient Boosting Machine (GBM) dan Light Gradient Boosting Machine (LightGBM). Dataset diperoleh dari Kaggle dan diproses melalui normalisasi serta *label encoding*, kemudian dilakukan *hyperparameter tuning* dengan *RandomizedSearch*. Hasil menunjukkan GBM memiliki akurasi 96,14% dan LightGBM 96,82%. Sistem ini diimplementasikan dalam antarmuka web menggunakan Streamlit. Temuan penelitian menunjukkan bahwa metode boosting efektif digunakan pada sistem rekomendasi tanaman berbasis kondisi tanah, sekaligus mendukung penerapan pertanian presisi yang berkelanjutan.

Kata kunci: Pertanian Presisi, *Machine Learning*, *Gradient Boosting*, LightGBM, Prediksi Tanaman

ABSTRAK

Agriculture is a strategic sector in supporting food security and sustainable development, particularly in achieving SDG 2 (Zero Hunger) and SDG 9 (Industry, Innovation, and Infrastructure). Global challenges such as soil degradation and climate change require technological innovations. This study aims to develop a crop prediction system based on soil parameters (N, P, K, temperature, humidity, pH) by comparing Gradient Boosting Machine (GBM) and Light Gradient Boosting Machine (LightGBM). The dataset was obtained from Kaggle, processed using normalization and label encoding, and optimized through RandomizedSearch for hyperparameter tuning. Results show that GBM achieved 96.14% accuracy, while LightGBM achieved 96.82%. The system was implemented into a web-based interface using Streamlit. The findings highlight that boosting methods are effective for crop recommendation systems based on soil conditions while supporting the adoption of sustainable precision agriculture.

Keywords: Precision Agriculture, Machine Learning, Gradient Boosting, LightGBM, Crop Prediction

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sektor pertanian memegang peranan fundamental dalam upaya mencapai beberapa Tujuan Pembangunan Berkelanjutan (SDGs), terutama SDG 2: Tanpa Kelaparan (*Zero Hunger*), dengan menopang ketahanan pangan dan stabilitas ekonomi dunia, sebagaimana ditegaskan dalam berbagai studi yang menyatakan bahwa pertanian merupakan fondasi bagi keberlangsungan hidup manusia sekaligus pilar utama pembangunan ekonomi dan ketahanan pangan global (Allahyari dan Poursaeed 2021). Seiring dengan proyeksi pertumbuhan populasi global, tuntutan untuk meningkatkan produktivitas agrikultur secara berkelanjutan menjadi semakin mendesak. Namun, sektor ini menghadapi tantangan signifikan berskala global, seperti degradasi tanah, cuaca ekstrem, dan penurunan kesuburan lahan, yang berisiko menurunkan produktivitas pertanian hingga 14% pada tahun 2050 jika tidak ada upaya adaptasi (Farah dkk. 2025). Kondisi ini diperparah oleh fakta bahwa 33% tanah dunia mengalami degradasi sedang hingga berat (Smith dkk. 2024), mengancam ketahanan pangan dan keberlanjutan ekosistem. Untuk mengatasi kompleksitas ini, inovasi yang sejalan dengan SDG 9: Industri, Inovasi, dan Infrastruktur menjadi krusial, di mana pendekatan pertanian presisi (*precision agriculture*) menjadi kunci untuk optimalisasi hasil panen yang efisien dan berkelanjutan. Selain itu, pemanfaatan data terbuka dan perangkat open-source memungkinkan praktik pertanian presisi berbasis bukti (Jeppesen dkk. 2022). Kondisi degradasi tanah dan perubahan iklim tersebut secara langsung memengaruhi ketersediaan unsur hara utama seperti nitrogen (N), fosfor (P), dan kalium (K), serta parameter lingkungan seperti suhu dan kelembaban tanah. Variabel-variabel ini sangat menentukan kesesuaian suatu lahan untuk jenis tanaman tertentu, sehingga analisis berbasis data terhadap parameter tersebut menjadi langkah strategis dalam mengoptimalkan pemilihan tanaman dan meningkatkan produktivitas pertanian.

Dalam ekosistem pertanian presisi, *machine learning* muncul sebagai teknologi transformatif yang mampu memberikan solusi berbasis data. Berbagai studi menunjukkan bahwa *machine learning* efektif dalam menganalisis variabel lingkungan yang kompleks seperti komposisi nutrien tanah (N, P, K), suhu, kelembaban, dan tingkat pH untuk memprediksi jenis tanaman yang paling cocok untuk suatu lokasi (Singh Mohan dkk. 2023). Dengan mendeteksi pola yang sulit dikenali secara manual, teknologi ini memungkinkan petani dan praktisi agrikultur untuk membuat keputusan yang lebih akurat, mengurangi risiko kegagalan panen, dan meningkatkan efisiensi penggunaan lahan (Meshram dkk. 2021; Muhammad dkk. 2023).

Di antara beragam teknik *machine learning*, metode *ensemble boosting* seperti *Gradient Boosting Machine* (GBM) dan *Light Gradient Boosting Machine* (LightGBM) telah menunjukkan performa yang superior untuk berbagai tugas klasifikasi. Sebuah studi relevan oleh (Wardhana dkk. 2022) menyoroti adanya *trade-off* antara kedua model ini, di mana satu model mungkin unggul dalam kecepatan sementara yang lain menawarkan stabilitas akurasi. Temuan ini membuka celah penelitian (*research gap*) yang penting: perlunya analisis perbandingan komprehensif untuk menentukan model mana yang memberikan keseimbangan terbaik antara akurasi prediksi dan efisiensi komputasi saat dihadapkan pada dataset agrikultur yang beragam dan mencakup berbagai kondisi iklim. Lebih jauh, eksplorasi mengenai dampak *hyperparameter tuning* terhadap performa masing-masing model dalam konteks global ini masih perlu didalami.

Oleh karena itu, penelitian ini bertujuan untuk melakukan analisis perbandingan kinerja dan performa antara model GBM dan LightGBM untuk prediksi jenis tanaman menggunakan dataset yang representatif secara global. Penelitian akan berfokus pada evaluasi pengaruh *hyperparameter tuning* terhadap akurasi dan efisiensi komputasi. Hasilnya diharapkan dapat memberikan wawasan berharga bagi komunitas ilmiah dan praktisi *precision agriculture*. Dengan demikian, penelitian ini tidak hanya berkontribusi secara teknis, tetapi juga secara langsung mendukung pencapaian SDG 2 dengan mendorong produktivitas

pertanian, serta sejalan dengan semangat inovasi pada SDG 9 untuk membangun solusi teknologi yang andal bagi tantangan global.

1.2 Perumusan masalah

1. Bagaimana performa model *boosting* (GBM, LightGBM) dalam sistem prediksi jenis tanaman?
2. Penerapan *hyperparameter tuning* untuk meningkatkan akurasi masing-masing model?

1.3 Pembatasan masalah

Laporan penelitian ini memiliki beberapa batasan, antara lain:

1. Cakupan Model:

   Penelitian ini hanya akan fokus pada model-model berbasis *boosting*, yaitu *Gradient Boosting*, *LightGBM*. Model atau teknik lain dan metode linear tidak akan dibandingkan.
2. Dataset

   Analisis dilakukan hanya dengan menggunakan dataset yang sudah tersedia secara publik yang bersumber dari Kaggle. Dataset dianggap sudah melalui proses *preprocessing* dasar seperti pembersihan data dan hanya mencakup fitur numerik seperti N, P, K, suhu, kelembaban, pH.
3. Evaluasi dan Parameter
   - Evaluasi model dengan metrik akurasi, *precision, recall, F1-score*.
   - *Hyperparameter tuning* dilakukan pada parameter yang telah ditentukan tanpa eksplorasi parameter yang terlalu luas.
4. Implementasi
   - Penerapan dan analisis model dilakukan dalam lingkungan *Python*.
   - Fokus penelitian adalah pada analisis kinerja model dalam hal akurasi.
5. Analisis Data
   - Penelitian ini tidak membahas secara mendalam mengenai interpretabilitas model atau analisis fitur *feature importance* secara lebih kompleks.
   - Data dianggap sudah dalam kondisi yang representatif sehingga tidak difokuskan pada eksplorasi atau pembersihan data lebih lanjut.

1.4 Tujuan

Tujuan dari penelitian ini adalah untuk mengimplementasikan dan mengevaluasi model boosting (GBM dan LightGBM) dalam sistem prediksi jenis tanaman serta menganalisis pengaruh *hyperparameter tuning* terhadap performa model.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat dalam pengembangan sistem prediksi berbasis *machine learning*, khususnya di bidang pertanian. Dengan menerapkan algoritma *Gradient Boosting Machine* (GBM) dan LightGBM yang telah melalui proses *hyperparameter tuning*, sistem yang dibangun mampu memberikan rekomendasi jenis tanaman yang sesuai berdasarkan parameter kondisi tanah seperti kandungan nitrogen, fosfor, kalium, suhu, kelembaban, dan pH. Sistem ini tidak hanya bermanfaat untuk meningkatkan efisiensi pertanian, tetapi juga memberikan dasar ilmiah bagi pengembangan sistem pertanian presisi berbasis data. Implementasi sistem ke dalam antarmuka web menggunakan *Streamlit* juga memudahkan penggunaan secara langsung oleh petani atau pihak terkait tanpa memerlukan pemahaman teknis mendalam.

1.6 Sistematika Penulisan

Sistem penulisan yang akan Penulis gunakan dalam laporan tugas akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Dalam BAB I ini, penulis membahas batasan-batasan penelitian, rumusan masalah, tujuan penelitian, metodologi, dan sistem penulisan.

BAB II: TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II memuat tentang penelitian terdahulu dan landasan teori yang berkaitan untuk membantu memahami tentang metode *boosting* yang akan dipakai.

BAB III : METODE PENELITIAN

Pada BAB III menjelaskan proses penelitian yang dimulai dari pengumpulan data hingga evalusi performa dan efisiensi dalam analisis.

BAB IV : HASIL DAN PEMBAHASAN

Pada BAB IV berisi pemaparan hasil penelitian yang mencakup hasil akhir sistem, klasifikasi data uji, evaluasi performa dan efisiensi model, serta analisis keunggulan model dalam melakukan klasifikasi

BAB V : KESIMPULAN DAN SARAN

Bab ini Penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian sebelumnya telah menggunakan pendekatan *machine learning* dengan memanfaatkan metode *boosting*. Penelitian oleh Wardhana dkk. (2022) mengkaji performa model *Gradient Boosting Machine* (GBM) dan *LightGBM* dalam klasifikasi jenis kacang kering. Hasilnya menunjukkan bahwa *LightGBM* memberikan akurasi pelatihan tertinggi 99% namun validasi hanya 91%, sedangkan GBM memberikan akurasi yang relatif stabil dengan kesalahan prediksi lebih kecil pada beberapa kelas. Pendekatan serupa juga digunakan dalam penelitian ini, di mana GBM dan *LightGBM* diimplementasikan untuk mengklasifikasikan jenis tanaman berdasarkan kondisi tanah, serta dilakukan *tuning* parameter untuk memperoleh model terbaik.

Penelitian sebelumnya menunjukkan bahwa *Light Gradient Boosting Machine* (LGBM) menghasilkan akurasi sebesar 83% dalam klasifikasi rumah sewa berdasarkan fitur-fitur seperti luas bangunan, jumlah kamar, dan lokasi, sedikit lebih rendah dibandingkan *Gradient Boosting Machine* (GBM) dengan akurasi 85%, namun LGBM unggul dalam efisiensi komputasi untuk dataset besar (Dahlia dan Agustyaningrum 2022).

Penelitian yang dilakukan oleh Rizka Dahlia dkk menganalisis kinerja algoritma *Gradient Boosting* dalam memprediksi harga sewa rumah menggunakan dataset dari Kaggle. Dalam penelitian tersebut, *Gradient Boosting* diterapkan untuk membangun model prediktif berdasarkan fitur-fitur seperti jumlah kamar, lokasi, ukuran rumah, dan status furnitur. Hasil penelitian menunjukkan bahwa algoritma ini mampu memberikan performa yang tinggi dengan akurasi mencapai 84,38% dan nilai AUC sebesar 92,65%, yang menandakan kemampuan model dalam mengklasifikasikan data positif dan negatif dengan akurasi tinggi (Dahlia Rizka, Fitriana Lady Agustin 2025).

Pada penelitian sebelumnya yang dilakukan oleh Atlantic dkk 2024, menunjukkan bahwa penelitian ini menggunakan teknik *boosting* pada klasifikasi

kelulusan mahasiswa dengan populasi 181 dan sampel 128 data. Setelah menggunakan nilai akurasi dalam mengevaluasi model prediksi yang diperoleh dari model GBM. Berdasarkan penelitian, didapat nilai akurasi model GBM yaitu 71,09% yang lebih besar dibanding model *CART* yaitu 67,97% (Atlantic dkk 2024).

Penelitian sebelumnya membandingkan metode *ensemble learning* salah satunya yaitu metode *boosting* pada klasifikasi penyakit diabetes dengan 3 buah dataset didapatkan akurasi tertinggi sebesar 81.82% dengan model *Gradient Boosting*. Pada dataset 2, didapatkan akurasi tertinggi sebesar 99.25% dengan menggunakan model *Light Gradient Boosting*. Sedangkan akurasi tertinggi pada dataset ketiga adalah 100% dengan menggunakan model *Light Gradient Boosting* (Cendani dan Wibowo 2022).

2.2 Dasar Teori

Berikut adalah dasar teori untuk penelitian dengan judul "Implementasi Metode *Boosting* untuk Prediksi Jenis Tanaman Berdasarkan Kondisi Tanah":

2.2.1 Kecerdasan Buatan

Kecerdasan buatan (*Articial Intelligence*) merupakan cabang ilmu komputer yang menekankan pada pembuatan sistem yang mampu melakukan tugas-tugas seperti pengambilan keputusan, klasifikasi, dan prediksi, menyerupai kecerdasan manusia. Dalam bidang klasifikasi dan prediksi, kecerdasan buatan banyak dimanfaatkan melalui pendekatan *machine learning* dan *deep learning* yang memungkinkan sistem mempelajari pola dari data historis untuk menghasilkan keputusan otomatis. Menurut Kaur (2023), teknik klasifikasi berbasis kecerdasan buatan telah digunakan secara luas dalam berbagai domain seperti pengenalan citra, teks, dan suara, serta terbukti meningkatkan akurasi sistem dalam konteks analitik prediktif dan aplikasi klinis(Kaur 2023). Dengan kemampuan untuk mengenali pola-pola kompleks dan menghasilkan prediksi berdasarkan data, kecerdasan buatan menjadi landasan penting dalam pengembangan sistem cerdas di berbagai bidang seperti pertanian, pendidikan, kesehatan, dan industri lainnya (Dhivya dan Bazilabanu 2023). (Rakuasa, dkk 2024) menjelaskan bahwa kecerdasan buatan dapat menjadi katalis dalam transformasi pendidikan dengan memberikan solusi pembelajaran yang adaptif dan personal, serta mampu menjangkau daerah terpencil

melalui platform digital. Hal ini menunjukkan bahwa penerapan kecerdasan buatan dalam suatu sistem, baik pendidikan maupun pertanian, memiliki potensi besar dalam meningkatkan efektivitas pengambilan keputusan berbasis data. Lebih lanjut, kecerdasan buatan memungkinkan monitoring dan evaluasi yang lebih akurat karena mampu menganalisis data dalam jumlah besar secara real-time. Dalam penelitian lain, kemampuan kecerdasan buatan untuk mengoptimalkan proses analitik, klasifikasi, hingga prediksi, menjadi landasan kuat dalam pengembangan sistem pendukung keputusan, seperti prediksi jenis tanaman berdasarkan kondisi tanah.

Dengan demikian, kecerdasan buatan tidak hanya relevan dalam konteks pendidikan, tetapi juga menjadi teknologi utama dalam bidang pertanian presisi, sistem rekomendasi, dan pengelolaan sumber daya secara efisien, yang semuanya bertujuan untuk mendukung pengambilan keputusan yang lebih tepat dan berkelanjutan.

2.2.2 *Machine learning*

Machine learning merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pengembangan algoritma dan model matematis yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa perlu diprogram secara eksplisit. Secara umum, *machine learning* bertujuan untuk mengembangkan sistem yang dapat meningkatkan kinerjanya seiring waktu berdasarkan pengalaman atau data yang diperoleh sebelumnya. Dengan kata lain, *machine learning* tidak hanya mencakup implementasi algoritma, tetapi juga membutuhkan pemahaman teoretis yang kuat untuk memastikan bahwa sistem yang dikembangkan mampu belajar secara efektif, efisien, dan dapat diandalkan dalam berbagai konteks aplikasi.

Dalam konteks penelitian ini, tugas utama adalah memprediksi menentukan jenis tanaman yang tepat untuk ditanam berdasarkan kondisi tanah berdasarkan sekumpulan fitur numerik seperti kandungan *Nitrogen* (N), *Fosfor* (P), Kalium (K), suhu, kelembaban, pH tanah.

2.2.3 *Boosting*

Boosting adalah salah satu metode *ensemble learning* yang bekerja dengan membangun model secara berurutan. Pada setiap iterasi, model baru berusaha untuk memperbaiki kesalahan yang dibuat oleh model sebelumnya. Dengan cara ini, *boosting* secara bertahap menurunkan *bias* dan meningkatkan akurasi prediksi.

Boosting pendekatan yang populer dalam *machine learning* yang digunakan untuk meningkatkan akurasi model prediktif dengan menggabungkan beberapa model lemah menjadi satu model yang kuat. Kontribusi teoretis yang penting berasal dari pendekatan gradient *boosting* yang sepenuhnya korektif, yang memperkenalkan pembaruan model yang komprehensif pada setiap iterasi menggunakan fungsi kerugian hinge kuadrat. Fungsi ini memiliki keunggulan tahan terhadap outlier dan memungkinkan analisis teoretis yang kuat terhadap kinerja model (Zeng, dkk 2022).

Boosting akan menghasilkan akurasi yang baik apabila dalam pengulangan *classifier* yang terbentuk mendekati nilai akurasi 50% (Nanda Putri Cintari 2024). Semua algoritma tersebut menggunakan dasar yang sama, yaitu meminimalkan fungsi kerugian secara iteratif sehingga model yang dihasilkan lebih kuat. Secara keseluruhan, teori *boosting* tidak hanya menunjukkan kekuatannya dalam meningkatkan performa model, tetapi juga memberikan kerangka kerja matematis yang kaya untuk pemahaman dan pengembangan algoritma yang efisien dan robust di berbagai domain aplikasi pembelajaran mesin.

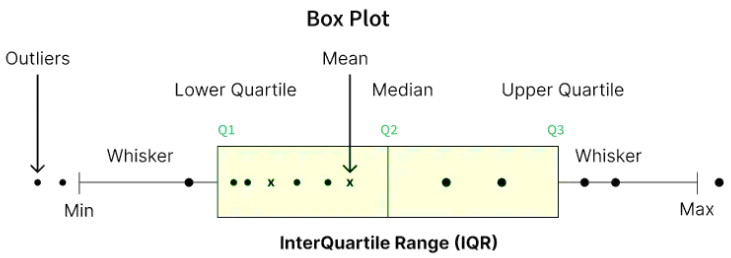
2.2.4 *Hyperparamater Tuning*

Setiap model dalam pembelajaran mesin memiliki sejumlah *hyperparameter* yang perlu ditentukan sebelum proses pelatihan. Optimasi *hyperparamater* adalah teknik yang melibatkan pencarian melalui berbagai nilai untuk menemukan subset hasil yang mencapai kinerja terbaik pada dataset tertentu. Nilai *hyperparameter* yang optimal bersifat relatif, tergantung pada karakteristik dataset yang digunakan serta tujuan dari model tersebut (Setyarini dkk. 2024). *Hyperparameter* mempunyai peran yang sangat penting dalam mengoptimalkan kinerja dari algoritma *machine learning* (Nugraha dan Sasongko 2022). Peningkatan suatu model adalah salah satu keberhasilan dari *hyperparameter tuning*. Berdasarkan

studi oleh (González-Castro dkk. 2024), *hyperparameter tuning* terbukti meningkatkan performa model seperti GBM dan LightGBM secara signifikan pada dataset tabular dengan fitur numerik. Karena dataset penelitian saya memiliki struktur yang sangat mirip (fitur numerik dan tugas klasifikasi), maka *tuning hyperparameter* juga sangat relevan dan efektif diterapkan. Selain itu (Handayani dkk. 2024), juga membuktikan bahwa *tuning hyperparameter* dapat meningkatkan akurasi dan performa model *Gradient Boosting* secara signifikan pada data tabular, yang relevan dengan struktur dataset dalam penelitian ini. Selanjutnya, berdasarkan penelitian (Bengio dan Bergstra 2022), pendekatan pencarian acak seperti *Randomized Search* terbukti lebih efisien dibandingkan *Grid Search*, terutama pada ruang parameter yang besar dan kompleks. Mereka menyatakan bahwa hanya sebagian kecil hyperparameter yang berdampak besar terhadap performa model, sehingga pencarian acak memiliki peluang lebih tinggi untuk menemukan kombinasi optimal dengan biaya komputasi yang lebih rendah.

2.2.5 *Boxplot*

Deteksi outlier pada analisis data merupakan faktor penting untuk mengetahui distribusi data, *Boxplot* yang juga dikenal sebagai *box-and-whisker plot*, adalah sebuah representasi grafis untuk menampilkan distribusi data numerik berdasarkan kuartilnya. Metode ini pertama kali diperkenalkan oleh seorang ahli statistik bernama John W. Tukey pada tahun 1977. Dasar teori *boxplot* bertumpu pada ringkasan lima serangkai (*five-number summary*) yang terdiri dari nilai minimum, kuartil pertama (*Q1*), median (*Q2*), kuartil ketiga (*Q3*), dan nilai maksimum.



Gambar 2. 1 *boxplot* grafik(Geeksforgeeks 2025)

Kotak (box) pada Gambar 2.1 *boxplot* merepresentasikan rentang antar kuartil, sedangkan garis median membagi data menjadi dua bagian. Jarak antara

kuartil ketiga dan kuartil pertama disebut *Interquartile Range (IQR)* yang dirumuskan sebagai berikut:

$$IQR = Q3 - Q1 \quad (1)$$

Keterangan:

- *Q1* = persentil ke-25 (nilai tengah dari separuh bawah data).
- *Q2* = median (persentil ke-50).
- *Q3* = persentil ke-75 (nilai tengah dari separuh atas data).

Deteksi outlier dapat dilakukan dengan memanfaatkan nilai *Interquartile Range (IQR)*. Suatu data dianggap sebagai outlier apabila nilainya berada di luar rentang batas bawah dan batas atas, yang dirumuskan sebagai berikut:

$$Batas\ Bawah\ (Lower\ Bound) = Q1 - 1.5 \times IQR \quad (2)$$

$$Batas\ Atas\ (Upper\ Bound) = Q3 - 1.5 \times IQR \quad (3)$$

Data yang memiliki nilai lebih kecil dari batas bawah atau lebih besar dari batas atas dikategorikan sebagai outlier. Dengan demikian, *boxplot* tidak hanya berfungsi sebagai ringkasan distribusi data secara grafis, tetapi juga sebagai alat penting untuk mengidentifikasi pencilan yang dapat memengaruhi hasil analisis statistik maupun pemodelan lebih lanjut.

2.2.6 *Gradient Boosting Machine* (GBM)

Gradient Boosting adalah algoritma yang diketahui efektif dalam halnya klasifikasi dan prediksi, algortima ini merupakan salah satu algoritma yang masuk dalam kategori *ensemble learning*, dengan menggabungkan beberapa model yang telah dibuat untuk membuat model yang lebih kuat. Proses ini secara umum membentuk sebuah model prediktif yang sangat akurat karena mampu menggabungkan sejumlah besar *weak learners* seperti decision tree dalam cara yang terkontrol dan efisien (Fafalios, dkk 2020).

Konsep utama dari GBM berasal dari interpretasi *boosting* sebagai bentuk optimisasi dalam ruang fungsi, di mana setiap iterasi bertindak seperti langkah gradient descent terhadap fungsi loss yang dipilih. Misalnya, jika loss yang digunakan adalah squared error, maka setiap langkah berfungsi untuk menyesuaikan model dengan arah negatif gradien kesalahan sebelumnya. Pendekatan ini memberikan fleksibilitas tinggi dalam memilih fungsi loss yang

sesuai dengan konteks masalah, baik regresi maupun klasifikasi (Hosen dan Amin 2021).

Selain keunggulan akurasi, penelitian terbaru menunjukkan bahwa GBM juga dapat dimodifikasi untuk menghadirkan karakteristik tambahan seperti *individual fairness*, dengan mengoptimalkan loss function yang robust terhadap *bias* algoritmik, bahkan pada model non-linear seperti decision tree (Vargo dkk. 2021). Pendekatan ini mempertahankan kekuatan prediksi GBM sekaligus memastikan bahwa hasilnya tidak diskriminatif terhadap individu atau kelompok tertentu.

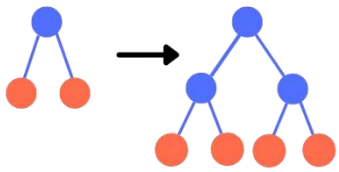
Menurut Rangkuti dkk. (2025), proses awal pada *GBM* dengan menentukan prediksi dasar yang meminimalkan total loss (kerugian), dirumuskan sebagai berikut:

$$F_o(x) = arg\min_{Y} \sum_{i=1}^{n} L(y_i, Y) \quad (1)$$

Keterangan:

- $F_o(x)$ = Fungsi prediksi awal sebelum model mulai melakukan boosting. Biasanya berupa fungsi konstan yang dipilih untuk meminimalkan total loss awal pada dataset.
- $arg\min_{Y}$ = Operasi untuk memilih nilai Y yang membuat jumlah total loss sekecil mungkin.
- $\sum_{i=1}^{n}$ = Penjumlahan atas semua data pelatihan, dari indeks $i = 1$ hingga n.
- $L(y_i, Y)$ = Fungsi loss yang mengukur seberapa buruk estimasi Y dibanding label asli y_i. Bentuk L bisa beragam, seperti squared error, log-loss, exponential loss, atau 0–1 loss.

Di mana $\sum_{i=1}^{n} L(y_i, Y)$ adalah fungsi loss untuk observasi ke-iii dengan target y_i dan prediksi awal Y.



Gambar 2. 2 level-wise GBM(Septiana Rizky, dkk 2022)

Seperti yang terlihat pada Gambar 2.1, model pohon yang terorganisir dibentuk melalui akumulasi berulang dari pembelajaran atau prediktor yang lemah kemudian diubah menjadi pembelajar yang kuat. Proses ini melibatkan pelatihan model-model berikutnya secara iteratif dengan tujuan untuk mengurangi kesalahan yang dibuat oleh model-model sebelumnya (Alawee dkk. 2024). Keuntungan dari GB adalah memiliki akurasi yang akurat dalam memprediksi dan proses yang cepat (Malek dkk. 2023).

2.2.7 *Light Gradient Boosting Machine* (*LGBM*)

Light Gradient Boosting merupakan salah satu algoritma *boosting* yang dikembangkan oleh Microsoft sebagai bentuk algoritma yang lebih efisien dari beberapa algortima *boosting* lainnya. Algoritma *LightGBM* ini menggunakan teknik histogram-based *learning*, yaitu dengan mengelompokkan nilai fitur ke dalam *bucket-bucket* diskrit, yang membuat proses pembelajaran menjadi lebih cepat dan efisien dalam pengguaan memori. *Light Gradient Boosting Machine* (LightGBM) adalah algoritma *boosting* berbasis pohon keputusan yang dirancang untuk efisiensi dan kecepatan dalam menangani data berukuran besar dan berdimensi tinggi. LightGBM merupakan pengembangan dari algoritma Gradient *Boosting* Machine (GBM) dengan beberapa inovasi utama yang meningkatkan performa komputasi, seperti *Gradient-based One-Side Sampling (GOSS)* dan *Exclusive Feature Bundling (EFB)* (Kriuchkova, dkk 2024). GOSS memungkinkan algoritma memfokuskan pelatihan pada sampel dengan gradien besar, sementara EFB menggabungkan fitur yang saling eksklusif untuk mengurangi dimensi data tanpa kehilangan informasi penting.

Menurut Rizky dkk. (2022), algoritma *LightGBM* membangun model prediksi secara iteratif berdasarkan formula:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (2)$$

Keterangan:

- $\hat{y}_i^{(t)}$ = nilai prediksi sampel ke- i pada iterasi ke- t.
- $\hat{y}_i^{(t-1)}$ = nilai prediksi pada iterasi sebelumnya.
- $f_t(x_i)$ = model baru (pohon keputusan) yang ditambahkan pada iterasi ke- t.

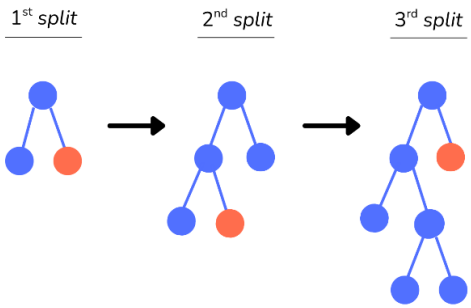
Di mana $\hat{y}_i^{(t)}$ adalah prediksi pada iterasi ke-t, dan $f_t(x_i)$ adalah model pohon baru yang dibentuk untuk memperbaiki kesalahan prediksi sebelumnya. Selain itu, fungsi objektif dalam *LightGBM* menggabungkan fungsi kerugian dan regularisasi:

$$\mathrm{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \sum_{t=1}^{T} \Omega(f_t) \quad (3)$$

Keterangan:

- y_i = nilai target sebenarnya.
- $l()$ = fungsi loss (misalnya squared loss atau log-loss).
- $\sum_{i=1}^{n}$ = penjumlahan atas semua data pelatihan, dari indeks $i = 1$ hingga n.
- $f_t(x_i)$ = fungsi prediksi dari pohon pada iterasi ke- t.
- $\Omega(f_t)$ = istilah regularisasi untuk mengendalikan kompleksitas model.
- T = jumlah total pohon yang dibangun.

Dengan fungsi regularisasi digunakan untuk mengontrol kompleksitas model dan menghindari *overfitting* (Rizky Septiana dkk. 2022).



Gambar 2. 3 leaf-wise LGBM(Septiana Rizky, dkk 2022)

Seperti yang ditunjukkan pada gambar 2.2 algoritma ini menggunakan pendekatan inovatif seperti pembagian *leaf-wise* dalam pembuatan pohon

keputusan, yang memungkinkan pengurangan kerugian lebih optimal dibandingkan dengan metode pembagian *level-wise* tradisional, serta algoritma ini mampu menangani fitur kategori tanpa proses *encoding* manual (H Yabes Dwi Nugroho, Zakiyabarsi Furqan 2025). Secara teoritis, LightGBM menggunakan strategi pembentukan pohon berbasis *leaf-wise* dibanding *level-wise* seperti pada GBM klasik. Strategi ini memungkinkan pertumbuhan pohon yang lebih dalam dan lebih akurat, meskipun rentan terhadap *overfitting* jika tidak disertai regularisasi yang tepat (Hajihosseinlou, dkk 2023). Secara keseluruhan, *LightGBM* merupakan algoritma *boosting* modern yang menggabungkan kecepatan pelatihan tinggi, efisiensi memori, dan fleksibilitas tinggi dalam menangani berbagai jenis data, menjadikannya pilihan utama dalam banyak aplikasi *machine learning* kontemporer.

2.2.8 Kondisi Tanah

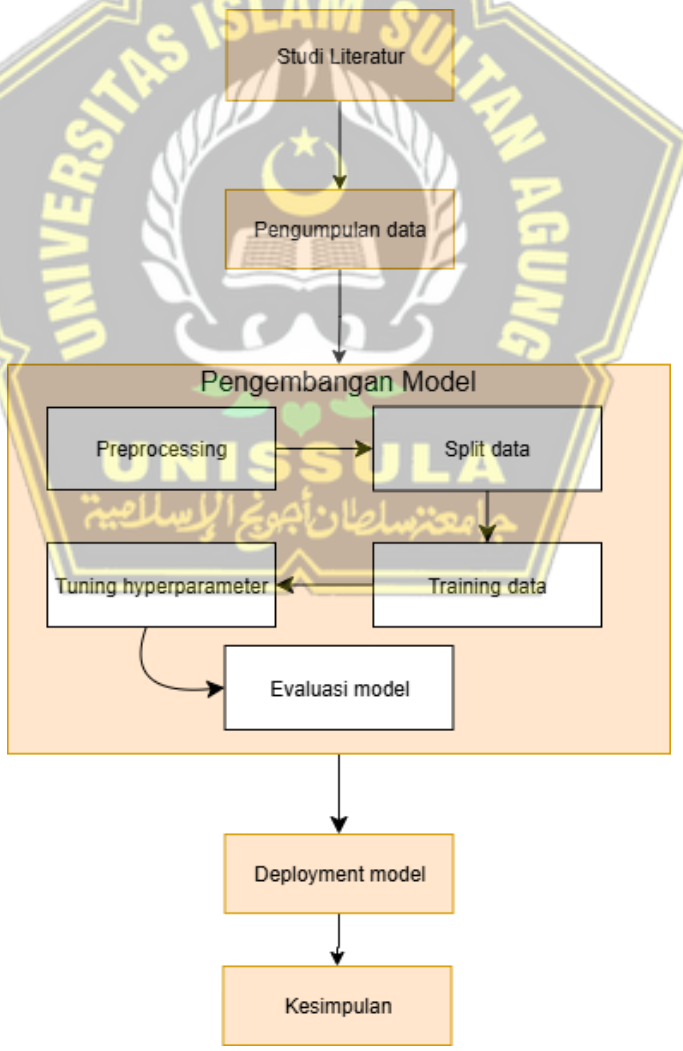
Mengetahui kondisi tanah salah satu upaya untuk meningkatkan produktivitas pertanian. Setiap jenis tanaman memiliki karakteristik tertentu seperti suhu rata-rata, kelembapan, pH tanah. Hal ini sejalan dengan penelitian (Nuriati dkk. 2021), yang menunjukkan bahwa pemanfaatan parameter tanah tersebut dapat membantu petani dalam menentukan tanaman yang cocok ditanam. Oleh karena itu, pemahaman parameter tanah menjadi dasar penting dalam pengembangan sistem prediksi berbasis data.

Pada penelitian (Liu, Yang, dan Li 2005), menunjukkan bahwa parameter seperti *nitrogen*(N), *fosfor*(F), *kalium*(K), dan kelembaban tanah berkontribusi secara signifikan terhadap hasil produksi tanaman. Selain itu, pemanfaatan data kondisi tanah seperti tekstur, kandungan unsur hara, dan suhu tanah telah menjadi dasar dalam sistem rekomendasi berbasis kecerdasan buatan untuk pemilihan jenis tanaman. Hal ini menunjukkan bahwa pemahaman terhadap parameter kondisi tanah bukan hanya penting dari sisi agronomis, tetapi juga berperan dalam pengembangan sistem prediksi dan pengambilan keputusan yang berbasis data.

BAB III
METODOLOGI PENELITIAN

3.1 Tahap Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem prediksi tanaman berbasis *machine learning* dengan pendekatan *boosting*. Metodologi yang digunakan mencakup serangkaian tahapan mulai dari pengumpulan data, pra-pemrosesan, hingga pengembangan dan evaluasi model. Data yang diperoleh dari sumber publik diolah melalui proses pembersihan, *encoding*, serta normalisasi untuk memastikan kualitas data. Adapun alur tahapan penelitian bisa dilihat pada gambar 3.1.



Gambar 3. 1 Tahap penelitain

1. Studi literatur

   Studi literatur dilakukan untuk mengkaji teori, metode dari penelitian terdahulu yang relevan dengan topik yang diambil.

2. Pengumpulan data

   Tahap ini bertujuan mengumpulkan data yang sesuai dengan kebutuhan penelitian penulis.

3. Pengembangan model

   Pengembangan model dilakukan melalui eksperimen dengan berbagai parameter guna memperoleh kinerja terbaik. Tahap ini juga mencakup proses *tuning hyperparameter*, dan optimalisasi arsitektur model agar dapat menghasilkan prediksi yang akurat dan andal sesuai tujuan penelitian.

4. Evaluasi model

   Evaluasi model dengan menguji model menggunakan data uji dan menghitung metrik seperti akurasi, presisi, atau F1-Score. Untuk memastikan bahwa model memiliki performa yang baik dan mampu menggeneralisasi data baru dengan cepat.

5. Deployment model

   Setelah dilakukannya evaluasi model, kemudian pada tahap akhir ini adalah *deployment*, yaitu menerapkan model ke dalam sistem. Ini memungkinkan model digunakan secara langsung.

6. Kesimpulan

   Kesimpulan disusun berdasarkan rangkaian hasil yang diperoleh selama proses penelitian. Fokus utamanya adalah pada model yang telah berhasil dikembangkan, performa yang dicapai, serta temuan-temuan penting yang muncul selama eksperimen.

   Pada tahap ini juga dievaluasi apakah solusi yang dirancang mampu menjawab rumusan masalah yang telah ditetapkan. Secara keseluruhan, seluruh tahapan penelitian dirangkum menjadi hasil akhir yang merepresentasikan pencapaian penelitian secara menyeluruh.

3.1.1 Studi Literatur

Studi literatur merupakan tahapan penting dalam proses penelitian yang bertujuan untuk mengkaji teori-teori, konsep, metode, serta hasil penelitian terdahulu yang relevan dengan topik yang sedang dikaji. Melalui studi literatur, peneliti memperoleh pemahaman yang mendalam terkait perkembangan teknologi, pendekatan metodologi, serta permasalahan yang telah diteliti sebelumnya.

Dalam penelitian ini, studi literatur difokuskan pada beberapa aspek utama, antara lain: konsep dasar *machine learning*, algoritma *Gradient Boosting* dan *LightGBM*, teknik *hyperparameter tuning*, serta metrik evaluasi model klasifikasi. Selain itu, penelitian-penelitian terdahulu yang berkaitan dengan prediksi tanaman berdasarkan data kondisi tanah juga turut dikaji untuk memperkuat landasan teoritis dan metodologis.

Dengan melakukan kajian literatur yang komprehensif, peneliti dapat mengidentifikasi celah penelitian (*research gap*), memperjelas kontribusi penelitian ini terhadap bidang keilmuan, serta memilih pendekatan yang paling tepat dalam membangun dan mengevaluasi model prediksi.

3.1.2 Pengumpulan Data

Dataset yang digunakan adalah data yang memuat informasi mengenai kondisi tanah seperti nilai N, P, K, suhu, kelembaban, pH serta label tanaman yang direkomendasikan. Dataset ini bersumber dari Kaggle yang dibuat dan diunggah oleh Atharva Ingle (Ingle Atharva 2020). Pada tabel 3.1 memuat informasi mengenai fitur kolom dan tipe data yang ada pada data penelitian yang digunakan.

Tabel 3. 1 informasi kolom pada dataset

No	Nama Kolom	Tipe Data	Keterangan
1	N	Int64	Rasio kandungan Nitrogen dalam tanah
2	P	Int64	Kandungan fosfor (Phosphorus) dalam tanah.
3	K	Int64	Kandungan Kalium dalam tanah
4	Temperature	Float64	Suhu dalam tanah dengan satuan derajat Celcius

5	Humidity	Float64	Kelembaban dalam tanah dengan satuan persen (%)
6	pH	Float64	Tingkat keasaman tanah
7	label	String	nama tanaman sesuai parameter tanah

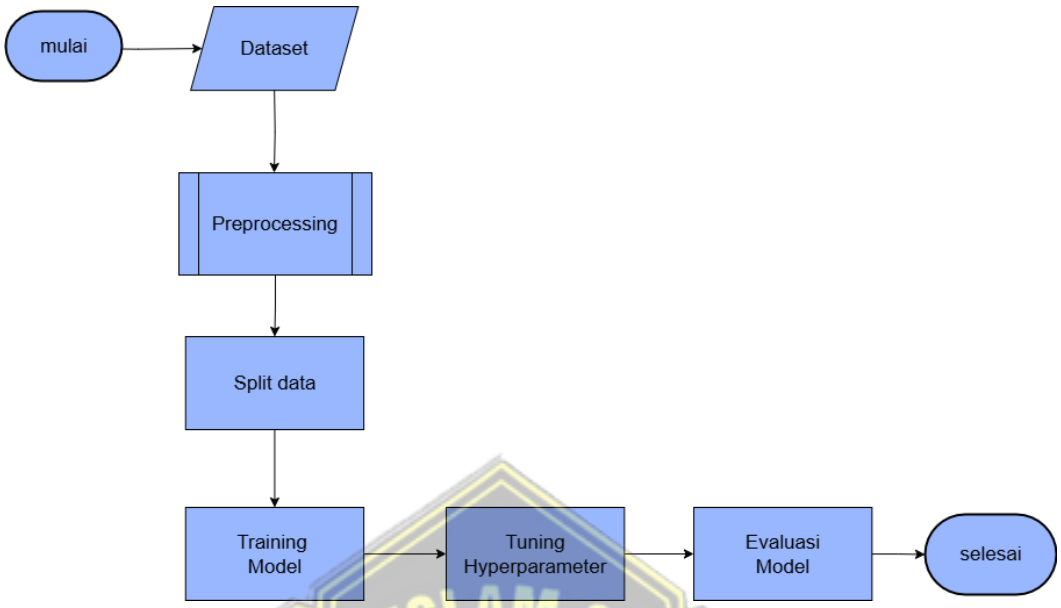
Pada tabel 3.2 berikut ini adalah gambaran contoh 10 data teratas dari data yang telah dijadikan objek penelitian.

Tabel 3. 2 contoh sepuluh data teratas

N	**P**	**K**	**temperature**	**humidity**	**pH**	**rainfall**	**Label**
90	42	43	20.880	82.002	6.50	202.935	Rice
85	58	41	21.770	80.320	7.03	226.655	Rice
60	55	44	23.004	82.320	7.84	263.964	Rice
74	35	40	26.491	80.158	6.98	242.864	Rice
78	42	42	20.130	81.604	7.62	262.717	Rice
69	37	42	23.058	83.370	7.07	251.054	Rice
69	55	38	22.708	82.639	5.70	271.324	Rice
94	53	40	20.277	82.894	5.71	241.974	Rice
89	54	38	24.515	83.535	6.68	230.446	Rice
68	58	38	23.223	83.033	6.33	221.209	Rice

3.1.3 Pengembangan Model

Untuk memberikan gambaran yang lebih jelas mengenai tahapan pengembangan model, maka divisualisasikan dalam bentuk diagram alir. Diagram tersebut menunjukkan langkah-langkah utama mulai dari pengumpulan dataset hingga tahap akhir evaluasi model. Secara berurutan, penelitian diawali dengan pengumpulan dataset, dilanjutkan dengan preprocessing data untuk memastikan kualitas input yang baik. Data kemudian dibagi menjadi data latih dan data uji (*split data*). Pada tahap berikutnya dilakukan pelatihan model menggunakan algoritma *Gradient Boosting Machine* (GBM) dan *Light Gradient Boosting Machine* (LightGBM), yang kemudian disempurnakan melalui proses *hyperparameter tuning* guna mendapatkan performa optimal. Hasil akhir dari pemodelan ini dievaluasi menggunakan metrik akurasi, *confusion matrix*. Alur pengembangan model secara lengkap ditunjukkan pada Gambar 3.2.



Gambar 3. 2 alur pengembangan model

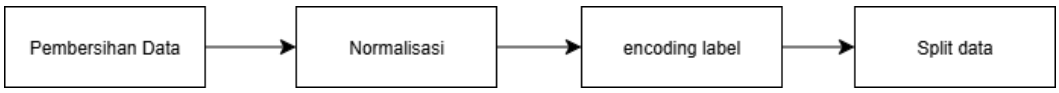
Gambar 3.2 diatas merupakan diagram alir yang menggambarkan tahapan sistematis dalam proses pengembangan model *machine learning* untuk keperluan penelitian. Proses dimulai dengan tahap *Mulai*, kemudian diikuti oleh:

1. *Dataset*

   Langkah ini merupakan proses pengumpulan data yang akan digunakan sebagai bahan utama dalam pelatihan dan pengujian model *machine learning*. Dataset yang digunakan harus relevan dengan permasalahan yang diteliti, memiliki kualitas data yang baik, serta mencakup fitur-fitur yang signifikan.

2. *Preprocessing*

   Pada tahap ini, data yang telah dikumpulkan dibersihkan dan dipersiapkan agar sesuai untuk proses pemodelan. Proses ini mencakup berbagai tahapan seperti:



Gambar 3. 3 Tahap *Preprocessing*

Gambar 3.3 menjelaskan tahapan pra-pemrosesan data (*preprocessing*) sebelum digunakan dalam proses pelatihan model *machine learning* untuk prediksi jenis tanaman berdasarkan kondisi tanah. Proses ini terdiri dari empat langkah utama:

a. Pembersihan Data

   Langkah pertama adalah membersihkan data dari nilai-nilai yang hilang (*missing values*), inkonsistensi, atau kesalahan input. Tujuannya adalah

memastikan bahwa data yang akan digunakan bersih dan valid untuk dianalisis.

b. Normalisasi

Setelah data dibersihkan, dilakukan proses normalisasi menggunakan *RobustScaler*. Proses ini mengubah data sehingga terpusat di sekitar median dan diskalakan berdasarkan rentang interkuartil (IQR), sehingga lebih tahan terhadap pengaruh nilai ekstrem atau outlier. Normalisasi ini penting agar fitur-fitur seperti pH, kelembapan, dan kadar nutrisi tanah berada dalam skala yang seragam.

c. Encoding Label

Label target yang berupa nama tanaman dikonversi menjadi format numerik menggunakan teknik encoding. Dalam hal ini digunakan label *encoding* agar model *machine learning* dapat memproses label target dengan benar dalam bentuk angka.

d. *Split data*

Tahapan terakhir adalah membagi dataset menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Pembagian ini bertujuan agar model dapat dilatih pada satu bagian data dan diuji kinerjanya pada bagian data yang belum pernah dilihat sebelumnya.

Keseluruhan proses *preprocessing* ini dilakukan untuk memastikan bahwa data yang digunakan dalam proses pelatihan memiliki kualitas yang baik, terstruktur, dan siap untuk menghasilkan model prediksi yang optimal.

3. *Split data*

Dataset yang telah melalui *preprocessing* kemudian dibagi menjadi dua bagian utama, yaitu data latih (*training data*) dan data uji (*testing data*). Pembagian ini bertujuan agar model dapat dilatih pada sebagian data dan diuji pada data yang belum pernah dilihat sebelumnya. Proporsi pembagian berkisar antara 80:20.

4. Training Model

Pada tahap ini, model *machine learning* dilatih menggunakan data latih. Model akan mempelajari pola dari data untuk membangun suatu fungsi prediktif.

Dalam penelitian ini, model yang digunakan adalah *Gradient Boosting* dan LightGBM, yang keduanya termasuk dalam kelompok algoritma *ensemble learning* berbasis pohon keputusan.

5. *Hyperparameter tuning*

   Setelah model dilatih secara awal, dilakukan proses *hyperparameter tuning* untuk meningkatkan kinerja model. *Hyperparameter* merupakan parameter yang nilainya ditentukan sebelum proses pelatihan dimulai. Dalam penelitian ini, digunakan metode *Randomized Search* untuk mencari kombinasi *hyperparameter* terbaik pada model *Gradient Boosting* dan LightGBM. Parameter yang di*tuning* antara lain:

   - *n_estimators*: jumlah total pohon yang dibangun
   - *max_depth*: kedalaman maksimum dari setiap pohon
   - *learning_rate*: tingkat penyesuaian dalam proses pembelajaran

   Proses ini penting untuk menghindari *overfitting* dan meningkatkan akurasi prediksi.

6. Evaluasi model

   Dalam evaluasi model, digunakan beberapa metrik untuk mendapatkan gambaran menyeluruh tentang performa model. Akurasi mengukur persentase prediksi yang benar pada data testing, sedangkan *precision, recall*, dan *F1-Score* memberikan gambaran keseimbangan antara kemampuan model dalam mengidentifikasi kelas positif dan negatif. Selain itu, *confusion matrix* merupakan salah satu alat evaluasi yang umum digunakan. Dalam konteks penelitian ini, metrik digunakan untuk memberikan gambaran menyeluruh mengenai seberapa baik model dalam memprediksi jenis tanaman berdasarkan parameter kondisi tanah. Tujuan dari evaluasi ini adalah untuk menemukan keseimbangan antara akurasi dan kinerja model agar mampu mengidentifikasi jenis tanaman yang sesuai secara efektif, sehingga dapat mendukung pengambilan keputusan di bidang pertanian berbasis data.

Model yang telah di*tuning* selanjutnya dievaluasi menggunakan data uji. Evaluasi dilakukan dengan menghitung metrik kinerja seperti:

a. Akurasi

Akurasi adalah metrik yang mengukur proporsi prediksi yang benar terhadap seluruh data. Metrik ini paling umum digunakan karena memberikan gambaran umum seberapa sering model membuat prediksi yang benar.

b. *Precision*

Presisi mengukur seberapa tepat model dalam memprediksi suatu kelas. Presisi tinggi menunjukkan bahwa dari seluruh prediksi terhadap suatu kelas, sebagian besar benar.

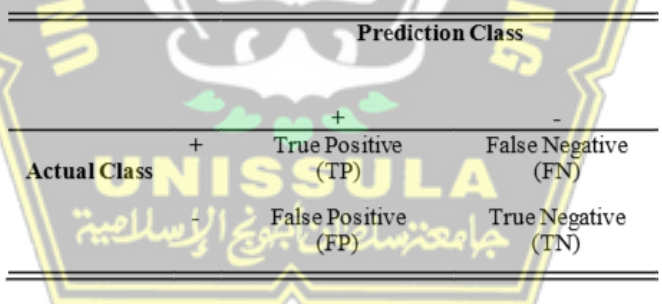
c. *Recall*

Recall mengukur seberapa baik model dalam menemukan semua data yang sebenarnya termasuk dalam suatu kelas.

d. F1-*Score*

F1-score adalah metrik gabungan dari presisi dan recall yang dihitung sebagai harmonik rata-rata keduanya. Metrik ini memberikan keseimbangan ketika kita perlu mempertimbangkan presisi dan recall secara bersamaan.

Hasil evaluasi ini digunakan untuk menilai apakah model telah bekerja sesuai harapan dan dapat digunakan untuk melakukan prediksi pada data baru.



Gambar 3. 4 rumus *confusion matrix*

Seperti yang terlihat di Gambar 3.4 yaitu dua konsep matematika yang digunakan untuk menggambarkan sensitivitas dan spesifisitas suatu pengujian dalam deteksi tanaman adalah akurasi prediksi positif dan negatif. Dalam konteks ini, data yang memenuhi kriteria disebut sebagai “positif”, sedangkan yang tidak memenuhi kriteria disebut sebagai “negatif” (Bhuyan dkk. 2023).

Nilai-nilai evaluasi seperti *precision*, *recall, accuray*, dan *F1-score* ditampilkan melalui rumus-rumus pada tabel 3.3 berikut ini.

Tabel 3. 3 rumus evaluasi

$$Precision = TP/((TP + FP)) \quad (4)$$

$$Recall = TP/((TP + FN)) \quad (5)$$

$$Accuracy = ((TP + TN))/((TP + TN + FP + FN)) \quad (6)$$

$$F1\ score\ ((TP + TN))/((TP + TN + FP + FN)) \quad (7)$$

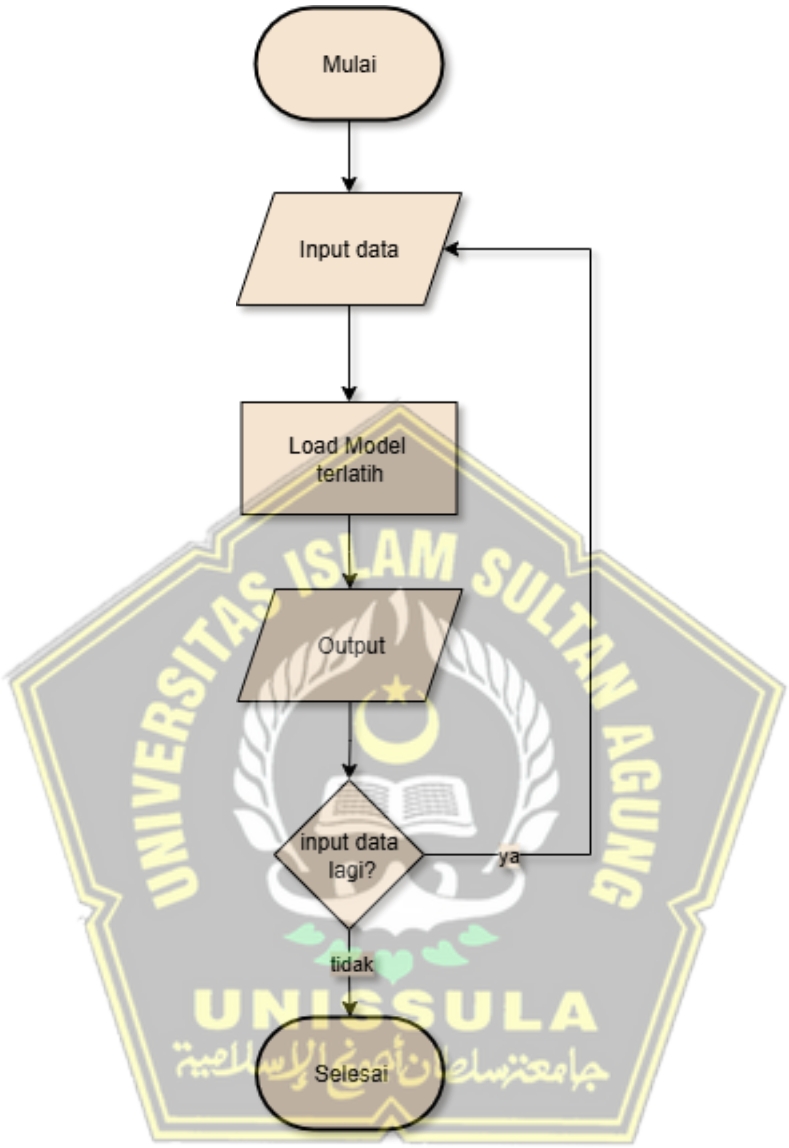
- *True Positive* (TP) mengacu pada jumlah jenis tanaman yang diprediksi dengan benar sesuai kenyataannya.
- *True Negative* (TN) adalah jumlah prediksi yang benar terhadap jenis tanaman yang memang tidak sesuai.
- *False Positive* (FP) adalah jumlah prediksi yang salah ketika model menganggap suatu jenis tanaman sesuai padahal tidak.
- *False Negative* (FN) adalah jumlah kesalahan prediksi ketika model menyatakan jenis tanaman tidak sesuai, padahal sebenarnya cocok.

Selain itu, analisis *overfitting* dilakukan dengan membandingkan akurasi antara data *training* dan *testing*

7. Selesai

   Merupakan tahap akhir dari keseluruhan proses. Pada tahap ini, model yang sudah dilatih dan dievaluasi dapat disimpan dan digunakan untuk keperluan implementasi lebih lanjut. Selain itu, hasil dan temuan dari proses ini dapat dianalisis dan disusun menjadi laporan penelitian.

3.2 Deployment Sistem



Gambar 3. 5 Diagram alir sistem

Diagram alir seperti ditunjukkan pada gambar 3.5 menggambarkan alur kerja sistem prediksi berbasis model *machine learning* yang telah dilatih sebelumnya. Proses dimulai dengan langkah *Mulai*, kemudian pengguna melakukan *Input data* sesuai parameter yang dibutuhkan oleh sistem. Selanjutnya, sistem akan memuat (*load*) model yang telah dilatih untuk memproses data tersebut. Setelah pemodelan selesai, sistem menghasilkan *Output* berupa hasil prediksi atau rekomendasi. Setelah output ditampilkan, pengguna diberikan opsi untuk melakukan *input data lagi*. Jika pengguna memilih “ya”, maka sistem akan kembali ke tahap input data.

Namun jika user tidak ingin menginputkan data lagi, maka proses akan dilanjutkan ke tahap *Selesai*. Alur ini dirancang agar sistem dapat digunakan secara berulang dengan efisien tanpa perlu memuat ulang model di setiap iterasi. Model yang telah dioptimasi akan diintegrasikan ke dalam sistem prediksi jenis tanaman. Sistem ini dilengkapi dengan antarmuka yang memungkinkan pengguna memasukkan nilai fitur secara manual, seperti nilai N, P, K, suhu, kelembaban, pH tanah. Berdasarkan input tersebut, model akan melakukan prediksi, dan hasilnya akan ditampilkan dalam format teks.

3.3 Analisis Kebutuhan

Dalam membuat dan membangun sistem prediksi berbasis *machine learning* ini, dibutuhkan beberapa komponen pendukung yang mencakup kebutuhan perangkat lunak, serta ekosistem python yang sesuai. Analisis kebutuhan ini bertujuan untuk mengetahui komponen pendukung apa saja yang dibutuhkan dan memastikan sistem berjalan sesuai dengan apa yang diinginkan. Berikut adalah daftar komponen pendukung yang digunakan:

A. *Python 3*

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer untuk pengembangan aplikasi berbasis data dan kecerdasan buatan. Versi 3.10 menawarkan stabilitas dan kompatibilitas yang luas dengan pustaka pembelajaran mesin modern. *Python* memiliki sintaks yang sederhana namun powerful, sehingga memudahkan dalam menulis, membaca, dan mengelola kode. Dalam penelitian ini, *Python* digunakan sebagai bahasa utama untuk seluruh proses mulai dari eksplorasi data, pelatihan model, evaluasi performa, hingga pembangunan antarmuka pengguna.

B. *Scikit-learn*

Merupakan pustaka *Python* yang menyediakan berbagai algoritma *machine learning* untuk klasifikasi, regresi, clustering, dan pemrosesan data. Dalam proyek ini, *scikit-learn* digunakan untuk mengimplementasikan Gradient *Boosting* Classifier, melakukan proses *tuning hyperparameter* dengan *RandomizedSearch* , serta mengevaluasi model menggunakan metrik seperti akurasi, precision, recall, F1-score, dan *confusion matrix*. Kelebihannya adalah

antarmuka API yang konsisten, dokumentasi lengkap, dan integrasi dengan pustaka lain seperti NumPy dan Pandas.

C. *Pandas* dan *NumPy*

Pandas dan *NumPy* merupakan pustaka dasar untuk manipulasi data dalam Python. NumPy digunakan untuk mengelola array dan operasi matematika numerik, sedangkan Pandas digunakan untuk memproses dataset berbasis tabel seperti DataFrame. Dalam proyek ini, Pandas digunakan untuk membaca dataset, melakukan eksplorasi data, dan membuat pivot table. NumPy banyak digunakan saat mengonversi data input ke format array numerik sebelum diberikan ke model.

D. *Jupyter Notebook*

Jupyter Notebook adalah lingkungan pengembangan interaktif berbasis web yang digunakan secara luas untuk *data science* dan *machine learning*. Jupyter memungkinkan pengguna untuk menulis kode Python, menjalankan perintah, melihat output, serta menyisipkan dokumentasi (*markdown*) dalam satu antarmuka yang sama. Dalam penelitian ini, Jupyter digunakan sebagai alat utama untuk melakukan eksplorasi data, pelatihan model, *tuning hyperparameter*, serta menampilkan visualisasi interaktif. Kelebihannya terletak pada kemudahan penggunaan dan visual feedback yang memudahkan debugging dan dokumentasi proses secara bersamaan.

E. *Streamlit*

Streamlit adalah framework *python* yang digunakan untuk membangun antarmuka pengguna berbasis web secara cepat dan interaktif. Dalam penelitian ini, Streamlit digunakan untuk membuat sistem rekomendasi tanaman berbasis input parameter tanah seperti nitrogen, fosfor, kalium, suhu, kelembaban, dan pH. Dengan Streamlit, pengguna dapat langsung memasukkan nilai input dan melihat hasil prediksi secara real-time tanpa perlu menulis kode.

F. Visual Studio Code

Visual Studio Code adalah text editor ringan namun powerful yang mendukung berbagai bahasa pemrograman termasuk Python. Dalam penelitian ini, VS

Code digunakan sebagai pendukung dalam menulis dan mengelola kode Python yang lebih kompleks, seperti implementasi sistem antarmuka menggunakan *Streamlit*. VS Code memiliki fitur seperti integrasi Git, terminal interaktif, *IntelliSense* (*auto-complete*), dan ekstensi Python yang sangat membantu pengembangan sistem secara modular dan efisien. VS Code menjadi pilihan yang ideal untuk proyek yang membutuhkan manajemen file terstruktur dan tampilan editor yang fleksibel.

BAB IV
HASIL DAN ANALISIS PENELITIAN

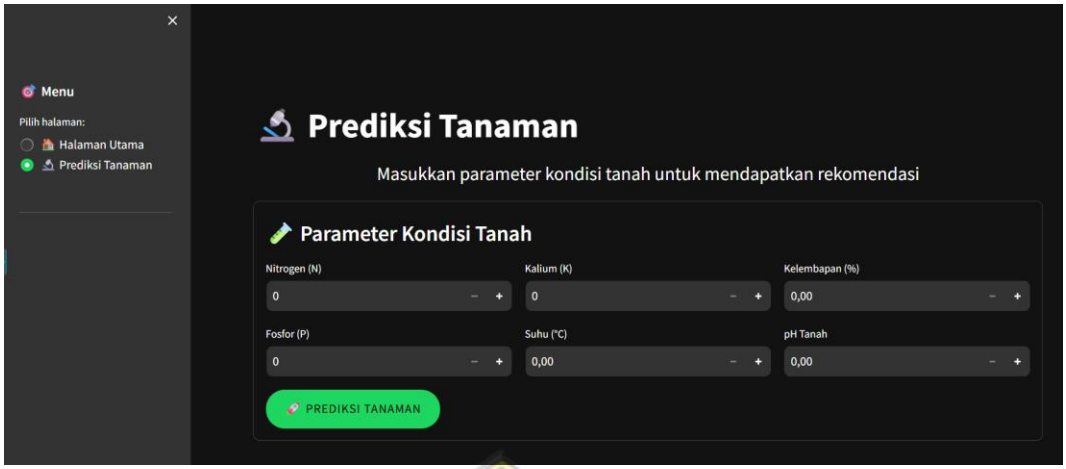
4.1 Hasil

4.1.1 Dashboard Streamlit

Sistem ini dikembangkan menggunakan model *Light Gradient Boosting Machine* (LightGBM), yang dipilih karena memberikan hasil akurasi prediksi yang lebih tinggi dibandingkan dengan model *Gradient Boosting Machine* (GBM) standar dalam tahap pengujian. Pemilihan model ini didasarkan pada evaluasi performa menggunakan metrik akurasi dan efisiensi pemrosesan data, sehingga mampu memberikan hasil prediksi yang optimal terhadap kondisi tanah yang bervariasi.

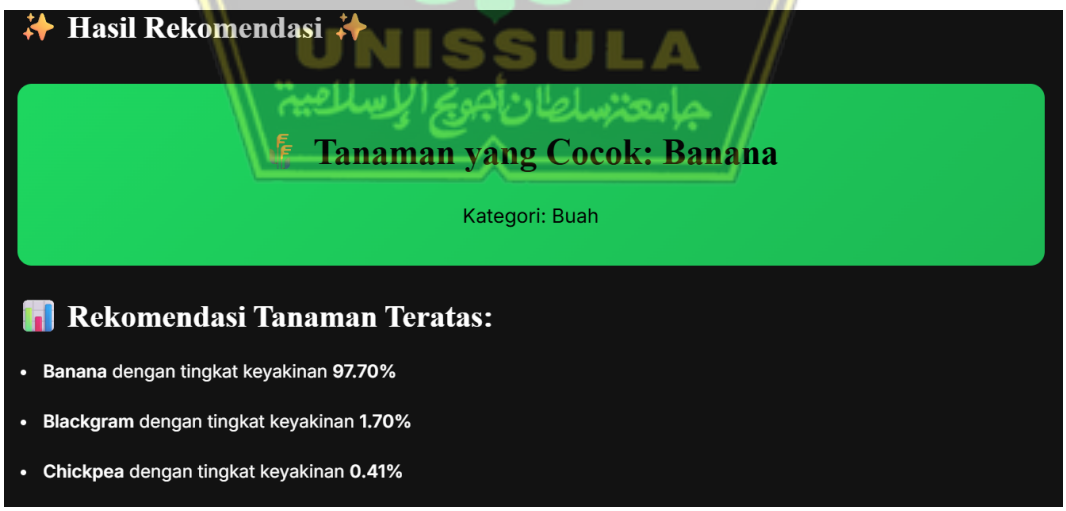
Gambar 4. 1 Tampilan Halaman Utama Sistem

Pada gambar 4.1 merupakan tampilan halaman utama sistem, halaman utama menampilkan informasi tentang sistem dan cara penggunaanya serta ada *sidebar* dibagian kiri halaman untuk memudahkan pengguna.



Gambar 4. 2 Tampilan Halaman Prediksi

Pada gambar 4.2 menampilkan antarmuka pengguna dari halaman prediksi pada sistem prediksi jenis tanaman berbasis kondisi tanah. Dalam tampilan ini, pengguna diminta untuk menginput sejumlah parameter penting yang berkaitan dengan kondisi tanah, yaitu kadar Nitrogen (N), Fosfor (P), Kalium (K), suhu tanah (°C), kelembapan(%), dan tingkat keasaman atau pH tanah. Setiap parameter dilengkapi dengan fitur input numerik yang memudahkan pengguna dalam memasukkan data secara manual. Setelah semua data terisi, pengguna dapat menekan tombol "Prediksi Tanaman" untuk memperoleh rekomendasi jenis tanaman yang sesuai berdasarkan kondisi tanah yang diberikan.



Gambar 4. 3 Hasil output sistem

Pada gambar 4.3 menampilkan hasil keluaran dari sistem setelah pengguna melakukan penginputan parameter kondisi tanah. Pada tampilan tersebut, sistem

merekomendasikan jenis tanaman yang paling sesuai, yaitu *banana*, lengkap dengan informasi mengenai kategorinya. Selain itu, sistem juga menampilkan tiga tanaman teratas lainnya yang diprediksi cocok berdasarkan data yang telah dimasukkan. Di bagian bawah, disediakan pula informasi tambahan berupa persentase kesesuaian antara parameter kondisi tanah yang telah diinputkan dengan karakteristik tanah ideal untuk masing-masing tanaman tersebut. Tampilan ini dirancang untuk mempermudah pengguna dalam memahami tingkat kecocokan dan mempertimbangkan alternatif tanaman lainnya berdasarkan hasil prediksi sistem.

4.2 Pengumpulan dan Eksplorasi Data

4.2.1 Deskripsi Dataset

Pengumpulan dataset yang yang digunakan bersumber dari platform *Kaggle*. Dataset ini memuat informasi mengenai parameter-parameter kondisi tanah seperti Nitogren, Phosphor, Kalium, suhu, kelembapan, pH tanah, serta label tanaman yang ada. Pada Tabel 4.1 menampilkan jumlah data per label.

Tabel 4. 1 jumlah data per label

No	Label	Jumlah
1	Rice	100
2	Maize	100
3	Chickpea	100
4	Kidneybeans	100
5	Pigeonpeas	100
6	Mothbeans	100
7	Mungbean	100
8	Blackgram	100
9	Lentil	100
10	Pomegranate	100
11	Banana	100
12	Mango	100
13	Grapes	100
14	Watermelon	100

No	Label	Jumlah
15	Muskmelon	100
16	Apple	100
17	Orange	100
18	Papaya	100
19	Coconut	100
20	Cotton	100
21	Jute	100
22	Coffee	100

Selanjutnya, fungsi shape digunakan untuk mengetahui jumlah baris dan kolom pada dataset, yang diketahui memiliki sebanyak 2200 baris dan 8 kolom data seperti yang terlihat pada Tabel 4.2.

Tabel 4. 2 contoh 10 data teratas

No	N	P	K	temperature	humidity	pH	rainfall	Label
1	90	42	43	20.880	82.002	6.50	202.935	Rice
2	85	58	41	21.770	80.320	7.03	226.655	Rice
3	60	55	44	23.004	82.320	7.84	263.964	Rice
4	74	35	40	26.491	80.158	6.98	242.864	Rice
5	78	42	42	20.130	81.604	7.62	262.717	Rice
6	69	37	42	23.058	83.370	7.07	251.054	Rice
7	69	55	38	22.708	82.639	5.70	271.324	Rice
8	94	53	40	20.277	82.894	5.71	241.974	Rice
9	89	54	38	24.515	83.535	6.68	230.446	Rice
10	68	58	38	23.223	83.033	6.33	221.209	Rice

Tabel 4.2 merupakan gambaran data yang akan digunakan dalam penelitian ini. Data ini bersumber dari platform *Kaggle* dengan format *CSV*, yang memuat informasi mengenai parameter kondisi tanah. Langkah awal yang akan dilakukan adalah memuat dataset ke dalam lingkungan *Python* dan melakukan eksplorasi awal untuk memahami struktur data. Proses ini mencakup beberapa proses yakni identifikasi tipe data, pengecekan nilai yang kosong, serta distribusi nilai dari tiap fitur.

4.2.2 Penghapusan Fitur

Yang dimaksud pada penghapusan fitur ini ialah menghapus fitur yang tidak relevan pada dataset yang digunakan, fitur yang akan dihapus yaitu *rainfall* karena dianggap kurang relevan terhadap penelitian ini.

	N	P	K	temperature	humidity	ph	label
0	90	42	43	20.879744	82.002744	6.502985	rice
1	85	58	41	21.770462	80.319644	7.038096	rice
2	60	55	44	23.004459	82.320763	7.840207	rice
3	74	35	40	26.491096	80.158363	6.980401	rice
4	78	42	42	20.130175	81.604873	7.628473	rice

```
(2200, 7)
```

Gambar 4.4 Hasil tampilan penghapusan fitur *rainfall*

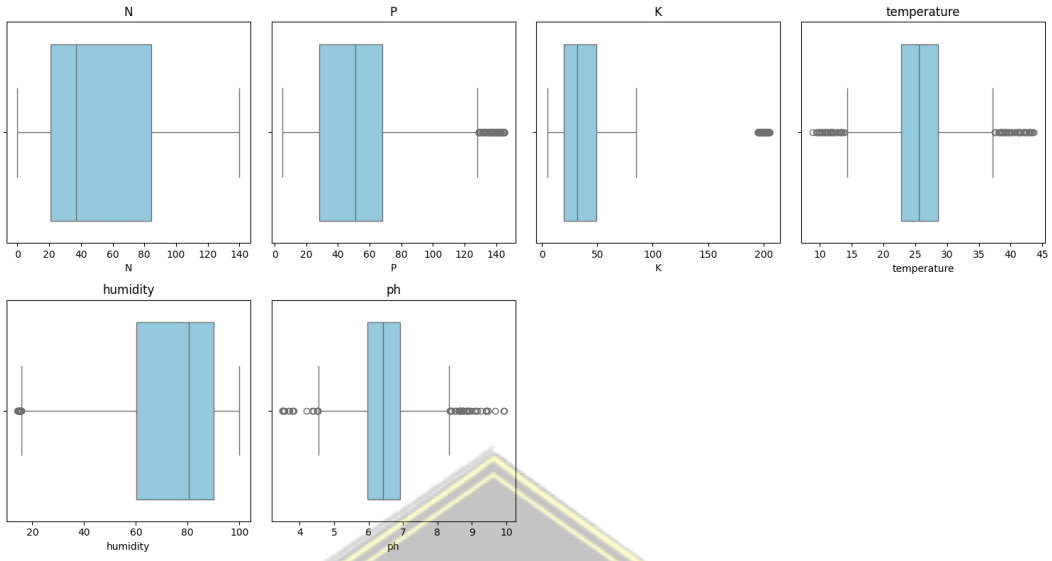
Pada Gambar 4.4 adalah hasil dari penghapusan fitur *rainfall*, fitur ini dihapus karena tidak digunakan dalam penelitian ini, dengan tujuan menyederhanakan model dan menyesuaikan input sistem prediksi.

Setelah dataset dimuat, dilakukan eksplorasi awal untuk memahami distribusi data dan karakteristik setiap fitur. Distribusi label tanaman diperiksa untuk memastikan bahwa dataset memiliki representasi kelas yang seimbang.

```
1  df = df.drop('rainfall', axis=1)
2  display(df.head())
3  print(df.shape)
```

4.2.3 Analisis Distribusi dan Outlier

Pada analisis distribusi dan outlier menyajikan visualisasi *boxplot* untuk setiap variabel numerik pada dataset, yaitu kadar nitrogen (N), fosfor (P), kalium (K), suhu (temperature), kelembapan (humidity), dan tingkat keasaman tanah (pH). *Boxplot* digunakan untuk menggambarkan distribusi data, rentang nilai, serta mendeteksi keberadaan outlier pada masing-masing fitur. Visualisasi ini membantu dalam memahami pola sebaran data dan potensi nilai ekstrem yang dapat mempengaruhi proses pemodelan.



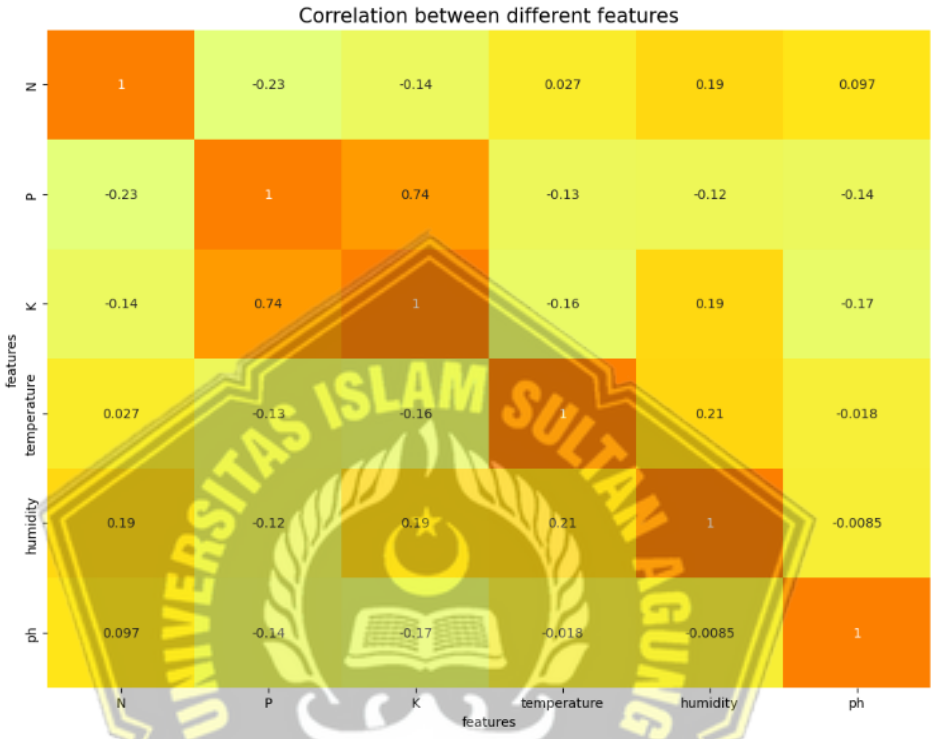
Gambar 4.5 visualisasi bloxplot

Untuk memahami pola sebaran data pada masing-masing fitur numerik, dilakukan analisis distribusi menggunakan visualisasi *boxplot*. *Boxplot* menampilkan nilai minimum, kuartil pertama (Q1), median, kuartil ketiga (Q3), serta mendeteksi keberadaan *outlier*, yaitu nilai-nilai yang berada di luar rentang interkuartil (IQR). Gambar 4.5 *boxplot* di atas menggambarkan distribusi setiap fitur numerik dalam dataset yang digunakan, yaitu Nitrogen (N), Phosphorus (P), Kalium (K), suhu (*temperature*), kelembapan (*humidity*), dan pH tanah (ph). Dari visualisasi tersebut, terlihat bahwa beberapa fitur memiliki nilai-nilai ekstrem (outlier), terutama pada fitur P dan K, yang menunjukkan sebaran data ke arah atas secara signifikan. Fitur temperature, humidity, dan ph juga mengandung sejumlah outlier, meskipun dalam jumlah yang relatif kecil. Sementara itu, fitur N memiliki distribusi yang cukup luas, namun tidak menunjukkan outlier yang mencolok.

Keberadaan nilai-nilai outlier pada sebagian fitur tersebut menunjukkan bahwa distribusi data tidak sepenuhnya simetris atau normal. Oleh karena itu, sebagai bagian dari tahap prapemrosesan, digunakan metode normalisasi *RobustScaler* yang lebih tahan terhadap pengaruh outlier dibandingkan metode normalisasi lain seperti *StandardScaler*. *RobustScaler* melakukan transformasi data berdasarkan nilai median dan rentang antar kuartil (IQR), sehingga mampu menjaga stabilitas skala tanpa terdistorsi oleh nilai-nilai ekstrem yang ada pada data.

4.2.4 Korelasi Antar Fitur

Selanjutnya dilakukan juga visualisasi korelasi antar fitur menggunakan *heatmap* dari nilai korelasi *Pearson*. *Heatmap* ini memberikan gambaran sejauh mana hubungan antar variabel numerik dalam dataset.



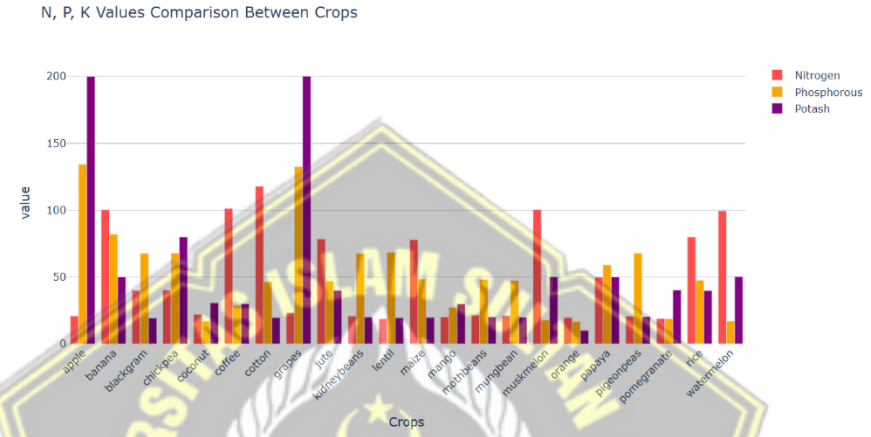
Gambar 4. 6 visualisasi korelasi antar fitur

Hasil visualisasi pada Gambar 4.6, terlihat bahwa terdapat korelasi yang sangat kuat antar fitur yang menunjukkan angka 1, dan itu menunjukkan bahwa setiap fitur memiliki kontribusi informasi yang relatif independen terhadap prediksi label tanaman. Dan terlihat bahwa korelasi tertinggi terjadi antara unsur phosphor (P) dan kalium (K) dengan nilai sebesar 0.74, yang menandakan adanya hubungan positif yang cukup kuat antara keduanya.

Di sisi lain, fitur-fitur lainnya menunjukkan korelasi yang relatif lemah satu sama lain, seperti antara *nitrogen* (N) dan kelembaban (*humidity*) sebesar 0.19, atau antara pH dengan variabel lain yang nilainya mendekati nol. Hal ini menunjukkan bahwa setiap fitur menyumbangkan informasi unik terhadap proses klasifikasi tanaman, sehingga seluruh fitur tetap relevan untuk dipertahankan dalam pemodelan.

4.2.5 Visualisasi Group Bar Chart

Grafik berikut menampilkan perbandingan kadar tiga unsur hara utama dalam tanah, yaitu nitrogen (N), fosfor (P), dan kalium (K), untuk berbagai jenis tanaman pada dataset. Perbandingan ini penting untuk memahami kebutuhan nutrisi spesifik setiap tanaman sehingga dapat dilakukan pengelolaan lahan dan pemupukan yang tepat sasaran.



Gambar 4. 7 visual *group bar chart*

Visualisasi pada Gambar 4.7 yang menampilkan perbandingan tiga elemen penting tanah (*Nitrogen, Phosphor, Kalium*) pada berbagai jenis tanaman dalam bentuk batang berkelompok, agar kita bisa membandingkan berapa banyak N, P, K yang dibutuhkan oleh masing-masing tanaman dalam satu grafik.

4.2.6 Visualisasi Pivot Tabel

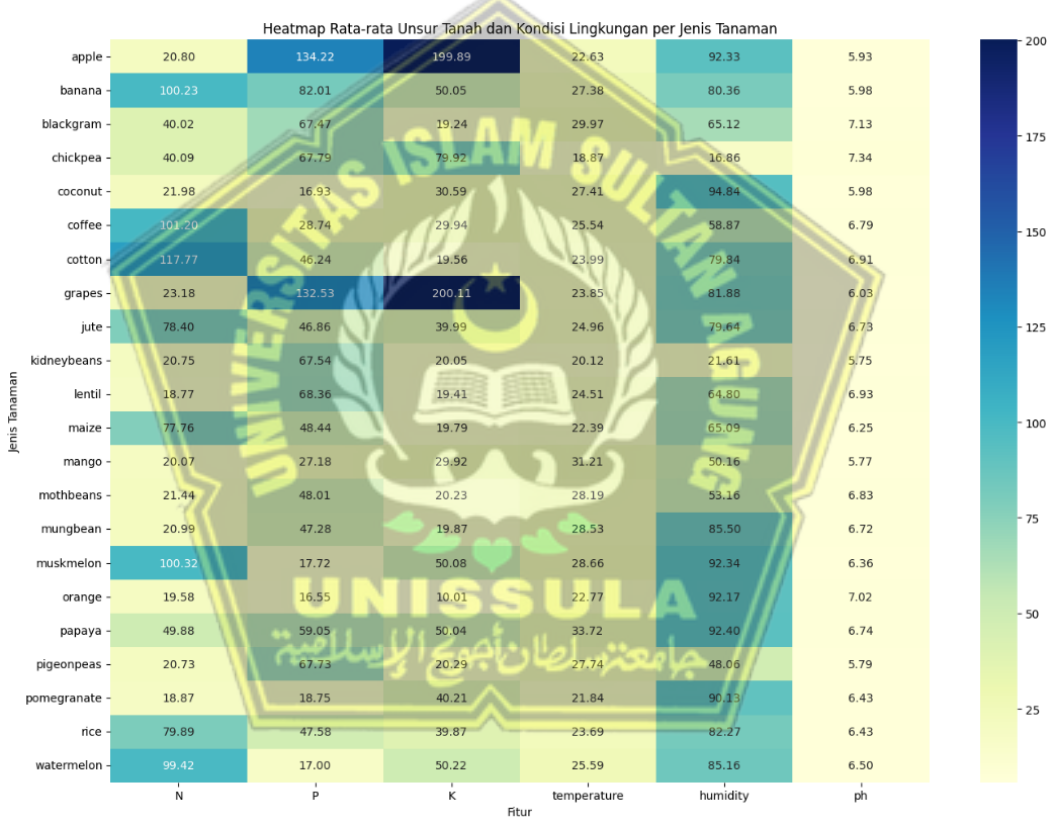
Untuk memahami pola distribusi unsur tanah berdasarkan jenis tanaman, dilakukan analisis menggunakan pivot table. Pivot table digunakan untuk menghitung nilai rata-rata dari masing-masing fitur numerik terhadap setiap jenis tanaman (label). Analisis ini bertujuan untuk mengidentifikasi karakteristik umum dari setiap tanaman berdasarkan kebutuhan unsur hara dan kondisi tanahnya.

Tabel 4. 3 sample rata-rata unsur hara tanah

label	K	N	P	humidity	ph	temperature
apple	199.89	20.8	134.22	92.333	5.929	22.6309424
banana	50.05	100.23	82.01	80.358	5.983	27.3767983

label	K	N	P	humidity	ph	temperature
blackgram	19.24	40.02	67.47	65.118	7.133	29.9733396
chickpea	79.92	40.09	67.79	16.860	7.336	18.8728467

Pada tabel 4.2 merupakan sampel hasil dari pembuatan pivot tabel yang memberikan informasi dengan bentuk angka yang digunakan untuk mengetahui rata-rata kebutuhan jenis tanaman terhadap unsur tanah. Dan hasil ini juga menjadi dasar untuk visualisasi dan interpretasi kebutuhan spesifik tanaman, serta untuk memahami sebaran nilai fitur yang digunakan oleh model.



Gambar 4. 8 heatmap rata-rata fitur numerik

Untuk memahami karakteristik masing-masing tanaman dalam dataset, dilakukan agregasi nilai rata-rata fitur numerik berdasarkan jenis tanaman menggunakan pivot tabel. Visualisasi dalam bentuk heatmap pada Gambar 4.8 menunjukkan perbedaan kebutuhan unsur hara tanah antar tanaman. Sebagai contoh, tanaman *apple* dan *grapes* memiliki rata-rata nilai *phosphorus* (P) dan *potassium* (K) yang jauh lebih tinggi dibandingkan tanaman lain, menunjukkan

bahwa kedua tanaman ini cenderung memerlukan tanah dengan kandungan hara tinggi. Sementara itu, tanaman seperti *chickpea* dan *lentil* tumbuh pada kondisi tanah dan kelembaban yang relatif rendah. Visualisasi ini membantu dalam memahami distribusi data serta membentuk intuisi awal sebelum melakukan pemodelan *machine learning*.

4.3 Pra-premrosesan Data

Pra-pemrosesan data merupakan langkah krusial yang harus dilakukan untuk memastikan bahwa data mentah siap dan optimal untuk digunakan dalam pelatihan model *machine learning*. Proses ini melibatkan serangkaian tahapan transformatif yang bertujuan untuk menyesuaikan dengan kebutuhan algoritma pembelajaran mesin, serta untuk meningkatkan performa dan akurasi model secara signifikan. Beberapa tahapan kunci dalam proses ini meliputi:

4.3.1 Pembersihan Data

Sebelum data digunakan dalam proses pelatihan model *machine learning*, dilakukan terlebih dahulu tahap pembersihan data (*data cleaning*). Langkah ini bertujuan untuk memastikan bahwa data bebas dari kesalahan, ketidakkonsistenan, atau nilai yang hilang (*missing values*) yang dapat memengaruhi kinerja model. Salah satu langkah awal dalam pembersihan data adalah mengecek keberadaan nilai kosong pada masing-masing kolom dalam dataset.

Pemeriksaan nilai hilang dilakukan menggunakan fungsi *isnull().sum()* dari pustaka *pandas* di Python. Fungsi ini akan menjumlahkan seluruh nilai kosong pada setiap kolom dalam dataset.

	0
N	0
P	0
K	0
temperature	0
humidity	0
ph	0
label	0

dtype: int64

Gambar 4. 9 pembersihan *missing values*

Gambar 4.9 menunjukkan hasil dari perintah *df.isnull().sum()* yang digunakan untuk mendeteksi nilai hilang pada setiap kolom dalam dataset. Dataset ini terdiri atas fitur-fitur numerik yang merepresentasikan parameter kondisi tanah, seperti kandungan nitrogen (N), fosfor (P), kalium (K), suhu (temperature), kelembapan (humidity), pH, serta label target (label) yang menunjukkan jenis tanaman.

Hasilnya menunjukkan bahwa semua kolom memiliki nilai nol, artinya tidak terdapat *missing values* pada seluruh data. Oleh karena itu, tidak diperlukan proses imputasi atau penghapusan baris, dan data dapat langsung digunakan untuk tahap *preprocessing* berikutnya, seperti normalisasi dan encoding.

4.3.2 *Label encoding*

Data label tanaman pada kolom label berbentuk teks seperti "*rice*", "*maize*", dan sebagainya agar dapat diproses oleh algoritma *machine learning*, label tersebut dikonversi menjadi nilai numerik menggunakan *LabelEncoder* dari *library sklearn*. Proses ini dilakukan dengan menyimpan hasil encoding dan label asli untuk keperluan interpretasi hasil.

Pada gambar 4.10 Kemudian memanggil kelas *LabelEncoder*, lalu memanggil fungsi *fit_transform()* untuk mengubah nilai string pada kolom label menjadi format numerik, dan hasilnya disimpan ke dalam variabel y. Variabel ini nantinya digunakan sebagai target (label) dalam proses pelatihan model.

```
# Pisahkan fitur dan label
X = df.drop('label', axis=1)
y_raw = df['label']

# Label Encoding
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y_raw)
```

Gambar 4. 10 kode *Label encoding*

Pada tahap pra-pemrosesan data, dilakukan proses encoding terhadap label untuk mengubah data kategorikal (nama-nama tanaman) menjadi bentuk numerik yang dapat diproses oleh algoritma *machine learning*. Encoding ini menggunakan metode *LabelEncoder* dari pustaka *Scikit-Learn*.

	label_asli	label_encoded
0	rice	20
1	rice	20
2	rice	20
3	rice	20
4	rice	20
...	...	...
2195	coffee	5
2196	coffee	5
2197	coffee	5
2198	coffee	5
2199	coffee	5

[2200 rows x 2 columns]

Gambar 4. 11 hasil label encoder

Pada gambar 4.11 menunjukkan hasil dari proses encoding, di mana kolom label_asli berisi nama tanaman seperti *rice* dan *coffee*, sementara kolom label_encoded berisi representasi numerik dari label tersebut, misalnya *rice* menjadi 20 dan *coffee* menjadi 5. Setiap label unik dalam dataset telah diberikan kode angka yang bersifat diskret dan tidak berurutan secara semantik, tetapi sangat diperlukan dalam tahap pemodelan.

4.3.3 Normalisasi

Fitur numerik seperti nitrogen (N), fosfor (P), kalium (K), suhu, kelembaban, dan pH dinormalisasi menggunakan metode *RobustScaler* dari pustaka Scikit-Learn. Normalisasi ini bertujuan untuk memastikan bahwa setiap fitur memiliki skala yang seragam, sehingga tidak ada satu fitur pun yang mendominasi proses pembelajaran akibat perbedaan skala antar fitur.

Meskipun algoritma tree-based seperti *Gradient Boosting Machine* (GBM) dan LightGBM (LGBM) relatif tidak sensitif terhadap skala fitur, proses normalisasi tetap dilakukan untuk menjaga konsistensi dalam pipeline pra-pemrosesan.

RobustScaler bekerja dengan median dan interkuartil range (IQR), sehingga lebih tahan terhadap nilai-nilai ekstrem atau outlier. Ini membuatnya sangat sesuai digunakan dalam dataset yang mengandung variabel dengan sebaran tidak simetris atau memiliki nilai ekstrim. Seperti dijelaskan oleh Rousseeuw dan Hubert, statistik robust bertujuan untuk menyesuaikan model terhadap mayoritas data dan mengurangi pengaruh outlier secara signifikan (Rousseeuw dan Hubert 2011). Hal ini didukung pula oleh Singh dkk, yang menunjukkan bahwa strategi estimasi *robust* dapat menjaga akurasi model meskipun data mengandung nilai ekstrem (Singh, dkk 2023).

```
scaler = RobustScaler()
X_scaled_array = scaler.fit_transform(X_numeric)
```

Gambar 4. 12 kode Normalisasi fitur

Sebelumnya, kolom label telah dipisahkan dari data fitur dan disimpan dalam variabel X. Pada gambar 4.12 menunjukkan kode normalisasi fitur menggunakan RobustScaler.

Dengan diterapkannya *RobustScaler*, data menjadi lebih siap untuk digunakan dalam tahap pelatihan model machine learning, sekaligus mengurangi risiko ketidakseimbangan skala dan dampak negatif dari outlier, sehingga dapat meningkatkan stabilitas dan akurasi model secara keseluruhan.

4.3.4 *Split* data

Sebelum melatih model *machine learning*, dataset kemudian dibagi menjadi dua bagian, yaitu data training dan data testing. Pada penelitian ini, data dibagi dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian menggunakan fungsi *train_test_split* dari pustaka *Scikit-learn*. Proses ini menghasilkan output seperti yang terlihat pada tabel 4.4. Pembagian data dilakukan dengan teknik stratifikasi berdasarkan label agar distribusi kelas tetap proporsional di antara data

training dan testing, teknik ini penting untuk menghindari *bias* distribusi kelas yang dapat memengaruhi akurasi model.

Tabel 4. 4 *Split data*

1	Jumlah data training	1760 data
2	Data Testing	440 data

Proses pembagian data ini sangat penting untuk menghindari *overfitting* dan untuk mengukur kemampuan model secara objektif. Dengan membagi data secara acak namun terstruktur, model dapat dilatih dengan baik menggunakan data pelatihan dan kemudian diuji pada data pengujian untuk mengetahui sejauh mana performa prediktifnya terhadap data yang tidak dikenal.

4.4 Pengembangan Model

Pengembangan model merupakan inti dari proses penelitian ini. Pada tahap ini, model *machine learning* dibangun dengan tujuan untuk mempelajari pola-pola tersembunyi dalam data yang telah diproses melalui tahapan eksplorasi dan pra-pemrosesan. Data yang digunakan dalam penelitian ini bersifat tabular dan mengandung berbagai fitur yang merepresentasikan kondisi tanah seperti pH, kadar air, dan tingkat kesuburan. Target dari model adalah jenis tanaman yang sesuai dengan kondisi tanah tersebut.

Dua algoritma *ensemble boosting* yang digunakan adalah *Gradient Boosting Machine* (GBM) dan *Light Gradient Boosting Machine* (LightGBM). Pemilihan algoritma tersebut bukan tanpa alasan. GBM dikenal mampu menangani permasalahan klasifikasi dengan kompleksitas tinggi secara efisien melalui pendekatan *boosting*, yaitu membangun model secara bertahap di mana setiap model baru mencoba memperbaiki kesalahan dari model sebelumnya. Sedangkan LightGBM adalah pengembangan dari GBM yang lebih modern dan dioptimalkan untuk performa lebih cepat dan efisien, terutama dalam hal kecepatan pelatihan dan penggunaan memori, sehingga sangat cocok digunakan untuk dataset besar dengan banyak fitur numerik.

Data dibagi menjadi dua bagian: 80% digunakan sebagai data latih (sebanyak 1.760 data) dan 20% sebagai data uji (sebanyak 440 data). Pembagian ini dilakukan menggunakan metode *stratified sampling* untuk menjaga distribusi kelas target agar

tetap seimbang antara data latih dan data uji. Tujuan dari proses ini adalah untuk membangun model yang tidak hanya dapat mempelajari pola dari data latih, tetapi juga mampu melakukan generalisasi dengan baik terhadap data baru yang belum pernah dilihat sebelumnya.

4.4.1 Hasil Pelatihan Model Awal

Proses pelatihan dimulai dengan membangun model dasar dari masing-masing algoritma, yaitu *Gradient Boosting Machine* (GBM) dan LightGBM (LGBM). Model dilatih menggunakan parameter default sebagai baseline untuk mengetahui sejauh mana masing-masing algoritma mampu mempelajari pola data. Hasil dari pelatihan awal ini akan menjadi dasar dalam melakukan *hyperparameter tuning* yang dijelaskan pada subbab selanjutnya.

Dengan pendekatan *supervised learning*, model mempelajari hubungan antara fitur-fitur masukan seperti unsur hara dan parameter tanah dengan target output berupa label tanaman. Setelah proses pelatihan selesai, model dievaluasi menggunakan data uji untuk mengukur akurasi prediksi klasifikasi.

1. Performa Model Sebelum *Tuning*

Tabel 4. 5 performa model sebelum *tuning*

Model	akurasi	precision	recall	F1-Score
GBM	96.36%	0.96	0.96	0.96
LGBM	96.36%	0.96	0.96	0.96

Sebelum dilakukan *hyperparameter tuning*, model dilatih menggunakan parameter default yang tersedia pada *library scikit-learn*. Tabel 4.5 menunjukkan hasil evaluasi performa awal dari kedua model. Model GBM menghasilkan akurasi sebesar 96,36%, dengan nilai precision, recall, dan F1-score rata-rata sebesar 0,96. Hasil ini menunjukkan bahwa model GBM sudah cukup andal dalam melakukan klasifikasi multikelas berdasarkan kondisi tanah.

Secara lebih rinci, nilai *precision, recall,* dan f1-score rata-rata (*macro average)* untuk seluruh kelas juga mencapai nilai sebesar 0.96, yang mengindikasikan bahwa model memiliki kemampuan yang cukup seimbang dalam mengenali semua kelas yang ada. Beberapa kelas seperti kelas 2, 8, 10, dan 20 menunjukkan nilai *recall* dan *precision* yang sedikit lebih rendah dibandingkan

kelas lainnya, meskipun nilainya masih tergolong tinggi. Hal ini mengindikasikan bahwa terdapat beberapa prediksi yang masih belum optimal pada kelas-kelas tertentu.

Namun, terdapat beberapa kelas seperti kelas 2, 8, 10, dan 20 yang memiliki nilai *precision* dan *recall* sedikit lebih rendah dibandingkan kelas lainnya, meskipun secara keseluruhan tetap berada pada rentang nilai tinggi. Hal ini mengindikasikan bahwa model masih belum sepenuhnya optimal dalam membedakan karakteristik antar kelas tersebut.

Sementara itu, model LGBM menunjukkan performa yang sedikit lebih tinggi dibandingkan GBM, dengan akurasi sebesar 96,82% dan nilai rata-rata precision, recall, serta F1-score sebesar 0,96. Keunggulan ini menunjukkan bahwa LGBM lebih efektif dalam mengenali pola-pola kompleks yang terdapat dalam data, meskipun selisihnya tidak terlalu signifikan.

Secara umum, kedua model telah menunjukkan performa awal yang sangat baik bahkan sebelum dilakukan *tuning*. Namun, untuk meningkatkan kemampuan generalisasi dan memaksimalkan potensi dari masing-masing model, diperlukan optimasi *hyperparameter* guna mencapai konfigurasi terbaik yang disesuaikan dengan karakteristik dataset yang digunakan.

4.4.2 *Hyperparameter Tuning*

Untuk meningkatkan performa model prediksi tanaman berdasarkan data tanah, proses *hyperparameter tuning* dilakukan terhadap algoritma *Gradient Boosting Machine* (GBM) dan LightGBM (LGBM). *Hyperparameter* merupakan parameter yang mengatur cara kerja internal algoritma dan tidak dapat dipelajari langsung dari data. Oleh karena itu, pencarian nilai *hyperparameter* yang optimal menjadi langkah krusial dalam membangun model yang tidak hanya akurat, tetapi juga stabil dan dapat melakukan generalisasi dengan baik terhadap data baru.

Pada penelitian ini, digunakan teknik *RandomizedSearchCV*, yaitu metode pencarian *hyperparameter* secara acak dari ruang parameter yang telah ditentukan. *RandomizedSearchCV* lebih efisien dari sisi komputasi, terutama ketika ruang pencarian luas. Meskipun tidak menjamin menemukan kombinasi absolut terbaik,

pendekatan ini sangat berguna dalam menemukan konfigurasi parameter yang kompetitif dalam waktu lebih singkat.

Adapun *hyperparameter* yang di *tuning* dalam model meliputi:

- *n_estimators*: Menentukan jumlah pohon (trees) dalam ensemble. Jumlah pohon yang lebih besar memungkinkan model mempelajari lebih banyak pola, namun berisiko menyebabkan overfitting jika tidak dikontrol dengan tepat.
- *max_depth*: Mengatur kedalaman maksimum setiap pohon. Semakin dalam pohon, semakin kompleks pola yang bisa ditangkap, namun berpotensi mengurangi kemampuan generalisasi model.
- *learning_rate*: Menentukan besar langkah pembaruan bobot selama proses boosting. Nilai yang terlalu tinggi dapat menyebabkan model tidak stabil, sedangkan nilai terlalu rendah memperlambat proses pembelajaran.
- *subsample*: Menentukan proporsi data pelatihan yang digunakan untuk membentuk setiap pohon. Nilai yang lebih kecil dapat meningkatkan variasi antar pohon dan mengurangi risiko overfitting.

Proses *tuning* dilakukan pada data latih menggunakan teknik *cross-validation* (validasi silang) untuk memastikan bahwa performa model tidak hanya baik pada satu subset data, tetapi juga konsisten pada data lain. Setelah proses *RandomizedSearchCV* selesai dijalankan, sistem akan memilih kombinasi *hyperparameter* terbaik berdasarkan nilai akurasi tertinggi yang diperoleh dari proses validasi silang.

Model dengan kombinasi parameter terbaik kemudian dilatih ulang menggunakan seluruh data latih untuk membentuk model akhir (*final model*). Model inilah yang selanjutnya digunakan untuk mengevaluasi performa pada data uji guna menilai kemampuan model dalam mengklasifikasikan jenis tanaman secara akurat berdasarkan kondisi tanah yang diberikan.

a. *Light Gradient Boosting Machine* (LGBM)

Untuk memperoleh performa klasifikasi yang optimal, dilakukan *hyperparameter tuning* pada model LightGBM menggunakan pendekatan *RandomizedSearchCV*. Metode *RandomizedSearchCV* mengeksplorasi sejumlah

kombinasi secara acak dari ruang parameter yang telah ditentukan. Pendekatan ini lebih efisien dalam hal waktu dan sumber daya, terutama ketika ruang pencarian parameter cukup luas.

```
param_dist = {
    'n_estimators': sp_randint(100, 500),
    'num_leaves': sp_randint(20, 100),
    'learning_rate': sp_uniform(0.01, 0.2),
    'max_depth': [-1, 5, 10, 15],
    'subsample': sp_uniform(0.6, 0.4),
    'colsample_bytree': sp_uniform(0.6, 0.4)
}
```

Gambar 4. 13 paramater LGBM

Pada Gambar 4.13 ditunjukkan konfigurasi ruang pencarian (*parameter space*) yang digunakan dalam proses *tuning* model LightGBM. Enam *hyperparameter* utama diatur untuk menemukan kombinasi terbaik menggunakan pendekatan *RandomizedSearchCV*. Pertama, *n_estimators*, yang menentukan jumlah total pohon dalam *ensemble*, diuji pada rentang nilai acak antara 100 hingga 500. Kedua, *num_leaves*, yaitu jumlah maksimum daun pada setiap pohon, divariasikan pada rentang 20 hingga 100. Ketiga, *learning_rate*, yang mengontrol besarnya pembaruan bobot pada setiap iterasi, diacak pada rentang 0.01 hingga 0.21. Keempat, *max_depth*, yang membatasi kedalaman maksimum pohon, dicoba dengan nilai -1, 5, 10, dan 15. Nilai -1 berarti tidak ada batasan kedalaman secara eksplisit, sehingga pertumbuhan pohon dikendalikan oleh parameter lain seperti *num_leaves*.

Selanjutnya, *subsample*, yang mengatur proporsi data latih yang digunakan pada setiap iterasi *boosting*, diambil secara acak pada rentang 0.6 hingga 1.0. Terakhir, *colsample_bytree*, yaitu proporsi fitur (kolom) yang dipilih secara acak untuk setiap pohon, juga diacak pada rentang 0.6 hingga 1.0. Kombinasi nilai-nilai acak dari seluruh parameter ini digunakan dalam proses pencarian untuk memperoleh konfigurasi *hyperparameter* yang optimal, sehingga diharapkan dapat meningkatkan performa prediktif model terhadap dataset penelitian.

Masing-masing parameter memiliki beberapa nilai alternatif yang akan dieksplorasi secara acak oleh *RandomizedSearch*.

```
lgbm = LGBMClassifier(random_state=42)
random_search = RandomizedSearchCV(lgbm, param_distributions=param_dist,
                                   n_iter=20, scoring='accuracy',
                                   cv=5, random_state=42, n_jobs=-1)
```

Gambar 4. 14 proses *tuning* LGBM

Pada gambar 4.14 tersebut menunjukkan Implementasi *tuning* dilakukan dengan menelusuri 20 kombinasi acak parameter (n_iter=20) dan validasi silang 5 fold (cv=5). Metrik evaluasi yang digunakan adalah akurasi (scoring='accuracy'), dan proses dilakukan secara paralel menggunakan seluruh inti prosesor (n_jobs=-1) untuk efisiensi waktu. Nilai *random_state*=42 digunakan agar hasil *tuning* bersifat reprodusibel. Setelah proses *tuning* selesai, model terbaik dari hasil pencarian disimpan dalam variabel *best_lgbm*, yang kemudian digunakan untuk proses pelatihan ulang serta evaluasi akhir pada data pengujian.

Setelah *tuning* selesai, diperoleh kombinasi hyperparameter terbaik sebagaimana ditampilkan pada Tabel 4.6, yaitu:

Tabel 4. 6 *best parameter LGBM*

colsample_bytree	*subsample*	*num_leaves*	*n_estimators*	*max_depth*	*learning_rate*
0.79	0.6	55	180	5	0.01

Kombinasi pada tabel 4.6 menunjukkan bahwa model memperoleh performa optimal dengan jumlah pohon yang besar, kedalaman pohon sedang, dan laju pembelajaran yang kecil sehingga pembaruan bobot berjalan secara bertahap. Nilai *subsample* yang rendah juga memberikan efek regularisasi dengan meningkatkan variasi antar pohon, sehingga membantu mengurangi risiko *overfitting*.

Secara keseluruhan, *hyperparameter tuning* yang dilakukan pada model LightGBM terbukti efektif dalam meningkatkan kemampuan klasifikasi sistem rekomendasi tanaman berdasarkan kondisi tanah. Kombinasi parameter optimal ini digunakan dalam pelatihan ulang model untuk memperoleh performa terbaik pada tahap evaluasi akhir.

b. *Gradient Boosting Machine* (GBM)

```
param_dist_gbm = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.05, 0.1],
    'max_depth': [3, 5, 7],
    'subsample': [0.6, 0.8, 1.0],
    'min_samples_split': [2, 5, 10]
}
```

Gambar 4. 15 parameter GBM

Adapun ruang pencarian hyperparameter yang digunakan ditampilkan pada Gambar 4.15, yang mencakup lima parameter utama: *n_estimators, learning_rate, max_depth, subsample,* dan *min_samples_split*. Parameter *n_estimators*, yang mengatur jumlah total pohon, diuji dengan nilai 100, 200, dan 300. Parameter *learning_rate*, yang menentukan besar kontribusi setiap pohon terhadap prediksi akhir, diuji dengan nilai 0.01, 0.05, dan 0.1. Nilai yang lebih rendah memungkinkan proses pembelajaran yang lebih hati-hati, namun memerlukan lebih banyak pohon untuk mencapai konvergensi.

Selanjutnya, parameter *max_depth*, yang mengatur kedalaman maksimum pohon, divariasikan pada nilai 3, 5, dan 7 guna menyesuaikan kompleksitas model terhadap pola dalam data. Parameter subsample diuji pada nilai 0.6, 0.8, dan 1.0 untuk menentukan proporsi data pelatihan yang digunakan dalam setiap iterasi *boosting*. Nilai yang lebih kecil dapat meningkatkan keragaman pohon dan mengurangi risiko *overfitting*. Terakhir, parameter *min_samples_split* diuji pada nilai 2, 5, dan 10 untuk mengatur jumlah minimum sampel yang diperlukan untuk memecah sebuah node; nilai yang lebih tinggi dapat menghasilkan model yang lebih sederhana dan tahan terhadap *overfitting*.

```
random_search_gbm = RandomizedSearchCV(
    gbm,
    param_distributions=param_dist_gbm,
    n_iter=20,
    scoring='accuracy',
    cv=5,
    random_state=42,
    n_jobs=-1
)
```

Gambar 4. 16 proses *tuning* GBM

Implementasi proses *tuning* menggunakan *RandomizedSearchCV* ditunjukkan pada Gambar 4.16. Dalam gambar tersebut, model dasar

GradientBoostingClassifier dikonstruksi terlebih dahulu, lalu pencarian 20 kombinasi parameter secara acak (n_iter=20) dilakukan dengan validasi silang 5-fold (cv=5). Evaluasi model dilakukan menggunakan metrik akurasi (*scoring='accuracy'*). Parameter *random_state*=42 digunakan untuk menjaga reprodusibilitas hasil, dan *n_jobs*=-1 diaktifkan untuk memanfaatkan seluruh inti prosesor yang tersedia guna mempercepat proses pencarian.

Untuk model *Gradient Boosting Machine* (GBM), proses *tuning* menghasilkan kombinasi *hyperparameter* terbaik yaitu :

Tabel 4. 7 best parameter GBM

subsample	n_*estimators*	*min_samples_split*	*max_depth*	*learning_rate*
0.68	489	3	6	0.0359

Setelah *tuning* selesai, diperoleh kombinasi hyperparameter terbaik seperti yang ditampilkan pada tabel 4.7, yaitu: *subsample*=0.68, n_*estimators*=489, *max_depth*=7, dan *learning_rate*=0.0359. Kombinasi ini menunjukkan bahwa GBM bekerja lebih optimal ketika hanya menggunakan sebagian data pelatihan pada setiap iterasi dan membangun pohon dengan kedalaman sedang untuk menangkap pola yang kompleks. Nilai *learning rate* yang sedang juga memberikan keseimbangan antara ketepatan dan kestabilan dalam pembelajaran model.

Dengan konfigurasi ini, model GBM berhasil mencapai performa yang baik dalam melakukan klasifikasi jenis tanaman berdasarkan parameter tanah. Hasil *tuning* ini kemudian digunakan untuk pelatihan ulang model akhir dan dievaluasi pada data uji guna menilai kemampuan generalisasi dari model yang telah dioptimalkan.

4.5 Evaluasi Model

Evaluasi model dilakukan untuk menilai performa algoritma dalam mengklasifikasikan jenis tanaman berdasarkan parameter kondisi tanah. Pada penelitian ini, dua algoritma yang dibandingkan yaitu *Gradient Boosting Machine* (GBM) dan *Light Gradient Boosting Machine* (LGBM). Evaluasi dilakukan dengan mengukur beberapa metrik utama yaitu akurasi, *precision*, *recall*, dan F1-score, yang masing-masing memberikan gambaran berbeda terhadap kinerja model.

Tabel 4. 8 Hasil evaluasi performa kedua model

Model	Akurasi	*Precision*	*Recall*	F1-*Score*	Jumlah data uji
GBM	96.14%	0.96	0.96	0.96	440
LGBM	96.82%	0.97	0.97	0.97	440

Berdasarkan tabel 4.8, dapat disimpulkan bahwa kedua model menunjukkan performa yang sangat baik dalam melakukan klasifikasi. Model *Gradient Boosting Machine* (GBM) memperoleh akurasi*, precision, recall, dan F1-score* sebesar 96.14%, sedangkan model LightGBM menunjukkan performa yang lebih unggul dengan akurasi sebesar 96.82% serta nilai *precision, recall, dan F1-score* sebesar 0.97, 0.97, dan 0.97 secara berurutan. Hal ini menunjukkan bahwa model LGBM memiliki kemampuan klasifikasi yang lebih akurat dan konsisten dibandingkan GBM pada jumlah data uji yang sama, yaitu sebanyak 440 data.

Namun untuk memperoleh pemahaman yang lebih mendalam terhadap performa model, khususnya dalam konteks klasifikasi multikelas, dilakukan analisis lanjutan melalui *confusion matrix*. Analisis ini membantu mengidentifikasi secara spesifik kelas mana yang mudah dikenali dan kelas mana yang cenderung mengalami kesalahan prediksi.

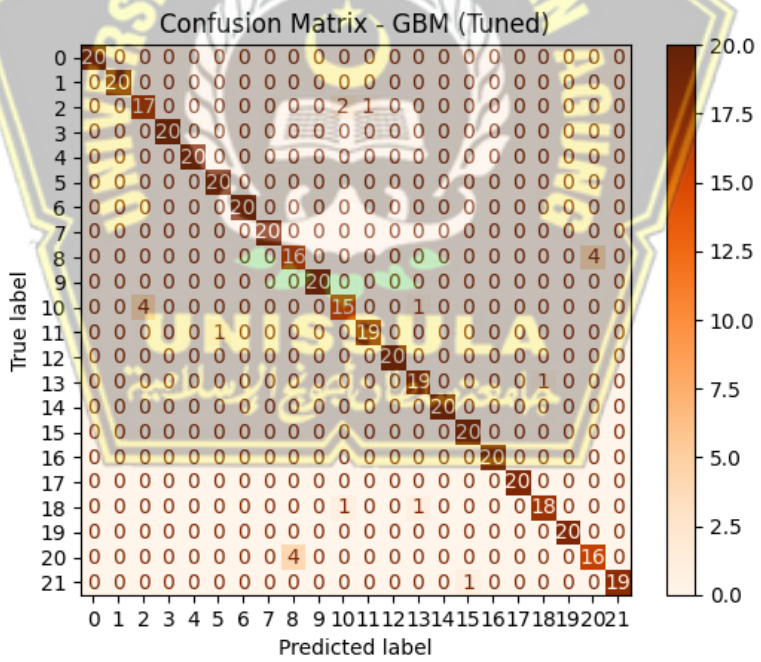
Setelah seluruh model dilatih menggunakan data latih, evaluasi performa dilakukan dengan menggunakan *confusion matrix*. Evaluasi ini penting untuk memahami secara mendetail bagaimana model memprediksi setiap kelas pada data uji, terutama dalam konteks klasifikasi multikelas seperti penelitian ini. *Confusion matrix* memberikan informasi menyeluruh terkait jumlah prediksi yang benar (*True Positive*, TP), prediksi salah (*False Positive*, FP dan *False Negative*, FN), serta *True Negative* (TN) pada masing-masing kelas. Dengan demikian, confusion matrix sangat membantu dalam mengidentifikasi pola kesalahan dan kekuatan model terhadap tiap label tanaman.

Dalam klasifikasi multikelas, matrix ini berbentuk persegi, di mana baris mewakili label sebenarnya dan kolom mewakili label prediksi. Nilai-nilai yang terletak di diagonal utama menunjukkan jumlah prediksi yang benar (TP) untuk tiap kelas. Semakin tinggi nilai diagonal, semakin baik performa model dalam mengenali label dengan tepat.

Visualisasi *confusion matrix* pada penelitian ini disajikan dalam bentuk heatmap berwarna, yang membantu memperjelas perbedaan akurasi antar kelas secara visual. Semakin gelap warna pada diagonal, semakin tinggi akurasi model pada label tersebut.

4.5.1 *Confusion Matrix Gradient Boosting Machine*

Berdasarkan hasil *confusion matrix* dari model *Gradient Boosting* yang terlihat pada gambar 4.17, dapat diamati bahwa sebagian besar prediksi berada pada diagonal utama matriks, yang menandakan bahwa model mampu mengklasifikasikan tanaman dengan sangat baik. Misalnya, tanaman seperti *'apple'*, *'coffee'*, *'grapes'*, dan *'rice'* semuanya berhasil diprediksi dengan akurasi tinggi. Hanya terdapat sedikit kesalahan prediksi yang terlihat dari nilai-nilai di luar diagonal, yang menunjukkan bahwa model memiliki presisi dan recall yang tinggi pada hampir seluruh kelas.



Gambar 4. 17 *Confusion Matrix* GBM

Gambar 4.17 memperlihatkan confusion matrix untuk model *Gradient Boosting Machine* (GBM). Dari visualisasi tersebut, terlihat bahwa mayoritas prediksi model berada di diagonal utama, yang mengindikasikan bahwa model mampu melakukan klasifikasi dengan sangat baik pada hampir semua kelas tanaman.

Beberapa contoh keberhasilan klasifikasi yang terlihat sempurna adalah pada label seperti:

- Kelas 0 (misal: Apple**)**, diprediksi dengan benar sebanyak 20 dari 20 sampel.
- Kelas 5 (misal: Coffee) dan Kelas 6 (misal: Grapes) juga menunjukkan hasil prediksi sempurna tanpa kesalahan.

Meski demikian, masih ditemukan sejumlah prediksi yang tidak tepat, sebagaimana ditunjukkan oleh nilai-nilai di luar diagonal. Sebagai contoh, kelas 10 hanya berhasil diklasifikasikan dengan benar sebanyak 15 dari 20 data, dengan 5 sisanya salah diklasifikasikan ke kelas lain. Demikian pula, kelas 18 mengalami 2 kesalahan klasifikasi yang menyebar ke beberapa label berbeda. Kesalahan-kesalahan ini kemungkinan besar disebabkan oleh kemiripan nilai-nilai fitur antar kelas, seperti kondisi tanah yang tumpang tindih atau rentang parameter yang mirip, yang menyulitkan model dalam membedakan secara jelas antar jenis tanaman.

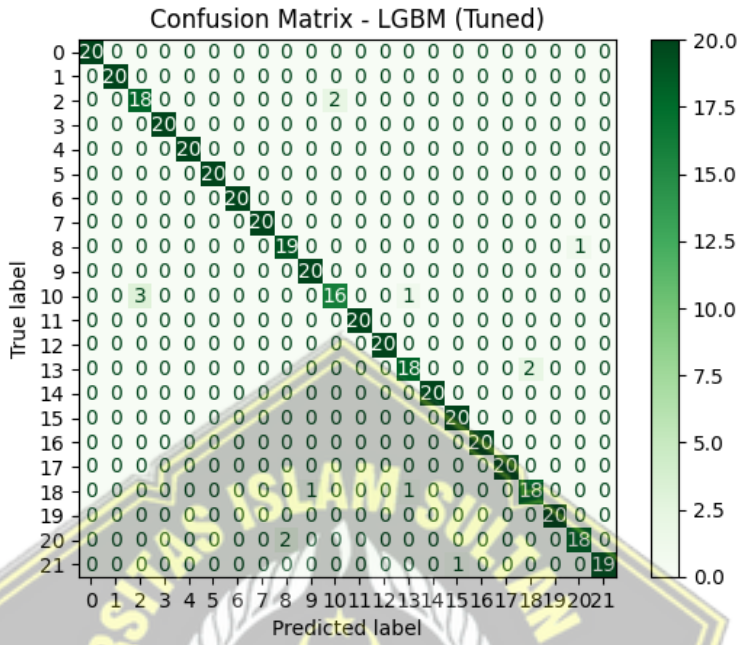
Secara keseluruhan, performa model GBM tergolong sangat baik dan stabil. Hal ini diperkuat oleh dominasi warna biru tua pada diagonal utama *heatmap confusion matrix*, serta rendahnya intensitas warna pada area luar diagonal, yang menunjukkan bahwa kesalahan klasifikasi bersifat minimal. Berdasarkan pola ini, dapat disimpulkan bahwa model memiliki nilai *precision* dan *recall* yang tinggi pada sebagian besar kelas, serta kemampuan generalisasi yang baik terhadap data uji.

4.5.2 *Confussion Matrix Light Gradient Boosting Machine*

Sementara itu, pada gambar 4.18 visual *confusion matrix* model *LightGBM* menunjukkan hasil yang hampir serupa dengan model GBM, dengan dominasi nilai diagonal yang tinggi yang menandakan bahwa sebagian besar prediksi dilakukan dengan tepat. Meskipun terdapat sedikit prediksi yang salah, seperti kesalahan klasifikasi minor pada label *pomegranate* dan *watermelon*, secara umum LGBM tetap mempertahankan akurasi yang tinggi pada sebagian besar label tanaman.

Performa *LightGBM* yang stabil ini menunjukkan bahwa model mampu mengenali pola data dengan baik dan menggeneralisasi secara efektif terhadap data uji. Dengan akurasi prediksi tinggi pada sebagian besar kelas dan kesalahan minor yang tersebar tipis di luar diagonal, LGBM terbukti menjadi salah satu alternatif

model yang kompetitif dan efisien untuk sistem rekomendasi tanaman berbasis kondisi tanah.



Gambar 4. 18 *Confusion Matrix* LGBM

Visualisasi confusion matrix pada Gambar 4.18 menggambarkan performa klasifikasi model LightGBM terhadap 22 kelas jenis tanaman. Matriks ini didominasi oleh nilai tinggi pada diagonal utama, menandakan bahwa mayoritas prediksi sesuai dengan label sebenarnya. Tingkat akurasi dan skor F1 yang masing-masing mencapai 97% menunjukkan bahwa model mampu mengenali dan membedakan pola data uji secara efektif. Meskipun terdapat beberapa kesalahan klasifikasi minor pada label blackgram (2), lentil (10), mothbeans (13), pigeonpeas (18), dan rice (20), jumlahnya relatif kecil dan tersebar tipis di luar diagonal. Distribusi kesalahan yang minim ini tidak memberikan dampak signifikan terhadap performa keseluruhan, sehingga LightGBM tetap terbukti konsisten dan akurat sebagai alternatif efisien untuk sistem prediksi jenis tanaman berbasis kondisi tanah.

4.5.3 Perbandingan Waktu Komputasi

Selain mengevaluasi performa berdasarkan metrik akurasi dan *confusion matrix*, penelitian ini juga membandingkan efisiensi waktu komputasi pada proses

pelatihan model. Waktu komputasi diukur sejak proses training dimulai hingga selesai membentuk model.

Tabel 4.9 berikut menunjukkan hasil perbandingan waktu pelatihan antara algoritma *Gradient Boosting Machine* (GBM) dan *Light Gradient Boosting Machine* (LightGBM):

Tabel 4. 9 waktu komputasi

Model	Waktu Pelatihan (detik)
GBM	3378.41
LGBM	281.66

Berdasarkan tabel tersebut, dapat dilihat bahwa waktu pelatihan GBM jauh lebih lama dibandingkan LightGBM. Selisih waktu yang signifikan ini menunjukkan keunggulan LightGBM dalam hal efisiensi komputasi. Hal ini sesuai dengan teori bahwa LightGBM menggunakan pendekatan *leaf-wise tree growth* dengan teknik *histogram-based learning*, sehingga mampu mempercepat proses pelatihan tanpa mengurangi kualitas hasil prediksi.

Sementara itu, GBM menggunakan strategi *level-wise tree growth*, yang lebih stabil namun membutuhkan sumber daya komputasi yang lebih besar. Dengan demikian, LightGBM lebih unggul untuk penelitian ini dari segi kecepatan pelatihan model, terutama jika dataset semakin besar.

BAB V
KESIMPULAN DAN SARAN

5.2 Kesimpulan

Penelitian ini berhasil mengembangkan sistem rekomendasi tanaman berbasis *machine learning* dengan memanfaatkan parameter kondisi tanah seperti nitrogen, phosphor, kalium, suhu, kelembaban, dan pH. Sistem ini dibangun menggunakan dua metode *boosting*, yaitu *Gradient Boosting Machine* (GBM) dan *Light Gradient Boosting Machine* (LGBM), yang keduanya terbukti mampu memberikan hasil klasifikasi yang sangat baik. Berdasarkan evaluasi performa, model GBM menunjukkan akurasi sebesar 96.14%, sedangkan model LGBM mencatatkan akurasi 96.82%, yang mengindikasikan bahwa metode *boosting* sangat efektif dalam memodelkan hubungan antara kondisi tanah dengan jenis tanaman yang sesuai.

Proses *hyperparameter tuning* menggunakan *RandomizedSearch CV* berkontribusi secara signifikan dalam meningkatkan akurasi model. Kombinasi parameter terbaik yang diperoleh menjadikan model lebih optimal dan mampu menghindari *overfitting*. Selain itu, hasil visualisasi dan analisis error menunjukkan bahwa sebagian besar prediksi berada pada kelas yang benar, meskipun masih terdapat beberapa kesalahan klasifikasi kecil pada tanaman dengan karakteristik tanah yang serupa.

5.3 Saran

1. Untuk penelitian selanjutnya, disarankan untuk menambahkan parameter lingkungan tambahan seperti curah hujan, jenis tanah, atau lokasi geografis guna memperkaya informasi yang digunakan dalam prediksi.
2. Perlu dilakukan pengumpulan data tambahan yang lebih luas dan mencakup variasi lokasi tanah yang berbeda agar model memiliki kemampuan generalisasi yang lebih baik dalam kondisi nyata.

3. Sistem yang telah dibangun dapat diintegrasikan lebih lanjut dengan sensor IoT atau platform mobile/web agar dapat diakses langsung di lapangan, sehingga mendukung pertanian presisi secara digital dan *real-time*.

DAFTAR PUSTAKA

Alawee, Wissam H dkk. 2024. "Forecasting sustainable water production in convex tubular solar stills using gradient boosting analysis." 318(April).

Allahyari, Mohammad S, dan Alireza Poursaeed. 2021. "Sustainable Agriculture: Implication for SDG2 (Zero Hunger)." 2(May 2019).

Atlantic, Virginnia, Evy Sulistianingsih, dan Hendra Perdana. 2024. "Gradient Boosting Machine Pada Klasifikasi Kelulusan Mahasiswa." *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster)* 13(2): 165–74.

Bengio, Yoshua, dan James Bergstra. 2022. "Random Search for Hyper-Parameter Optimization James." *ACM International Conference Proceeding Series* 13: 90–94.

Bhuyan, Bikram Pratim, Ravi Tomar, dan T P Singh. 2023. "Crop Type Prediction : A Statistical and Machine Learning Approach." : 1–17.

Cendani, Linggar Maretva, dan Adi Wibowo. 2022. "Perbandingan Metode Ensemble Learning pada Klasifikasi Penyakit Diabetes." *Jurnal Masyarakat Informatika* 13(1): 33–44.

Dahlia Rizka, Fitriana Lady Agustin, Seimahuira Syarah. 2025. "ANALISIS ALGORITMA GRADIENT BOOSTING DALAM PENGARUH." 8: 36–44.

Dahlia, Rizka, dan Cucu Ika Agustyaningrum. 2022. "Perbandingan Gradient Boosting dan Light Gradient Boosting Dalam Melakukan Klasifikasi Rumah Sewa." *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)* 5(6): 1016–20.

Dhivya, P, dan A Bazilabanu. 2023. "Deep hyper optimization approach for disease classification using artificial intelligence." *Data Knowl. Eng.* 145: 102147. https://consensus.app/papers/deep-hyper-optimization-approach-for-disease-bazilabanu-dhivya/442e3f8e71275566994f70cd297346b5/.

Fafalios, S, Pavlos Charonyktakis, dan I Tsamardinos. 2020. "Gradient Boosting Trees." https://consensus.app/papers/gradient-boosting-trees-tsamardinos-fafalios/9a1ea138cf8e56ccb7ab5b89ad862c3f/.

Farah, Abdikarim Abdullahi, Mohamud Ahmed Mohamed, Osman Sayid Hassan Musse, dan Bile Abdisalan Nor. 2025. "The multifaceted impact of climate

change on agricultural productivity: a systematic literature review of SCOPUS-indexed studies (2015–2024).” *Discover Sustainability* 6(1). https://doi.org/10.1007/s43621-025-01229-2.

Geeksforgeeks. 2025. “What is Box plot and the condition of outliers?” https://www.geeksforgeeks.org/data-visualization/what-is-box-plot-and-the-condition-of-outliers/.

González-Castro, Lorena dkk. 2024. “Impact of Hyperparameter Optimization to Enhance Machine Learning Performance: A Case Study on Breast Cancer Recurrence Prediction.” *Applied Sciences (Switzerland)* 14(13).

H Yabes Dwi Nugroho, Zakiyabarsi Furqan, Paramita Andi Jamiati. 2025. “IMPLEMENTASI SMOTE-ENN DAN BORDERLINE SMOTE TERHADAP PERFORMA LIGHTGBM PADA IMBALANCED CLASS PENDAHULUAN Perkembangan teknologi atau transformasi digital disektor perdagangan seperti e- commerce telah meningkatkan popularitas belanja daring secara global.” 10(1): 51–59.

Hajihosseinlou, Mahsa, A Maghsoudi, dan R Ghezelbash. 2023. “A Novel Scheme for Mapping of MVT-Type Pb–Zn Prospectivity: LightGBM, a Highly Efficient Gradient Boosting Decision Tree Machine Learning Algorithm.” *Natural Resources Research* 32: 2417–38. https://consensus.app/papers/a-novel-scheme-for-mapping-of-mvttype-pb–zn-prospectivity-ghezelbash-maghsoudi/2204c2af0b165d859b13b4b934ba3164/.

Handayani, Susi dkk. 2024. “Peningkatan Performa Model Gradient Boosting dalam Klasifikasi Stroke Melalui Optimasi Grid Search.” *Jurnal Fasilkom* 14(3): 722–28.

Hosen, Md Saikat, dan Ruhul Amin. 2021. “Significant of Gradient Boosting Algorithm in Data Management System.” *Engineering International* 9(2): 85–100.

Ingle Atharva. 2020. “dataset crop recommendation.” https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset/data.

Jeppesen, Jacob Høxbroe, Rune Hylsberg Jacobsen, Rasmus Nyholm Jørgensen,

dan Thomas Skjødeberg Toftegaard. 2022. "Towards Data-Driven Precision Agriculture using Open Data and Open Source Software." *International Conference on Agricultural Engineering 2016*: 1–6. https://arxiv.org/abs/2204.05582%0Ahttps://arxiv.org/pdf/2204.05582.

Kaur, Manmeet. 2023. "A Comprehensive Overview of Artificial Intelligence-Based Classification Techniques." *International Journal of Science and Research Archive*. https://consensus.app/papers/a-comprehensive-overview-of-artificial-kaur/b74733d7a33c56a7bb1601c90cdd1c2e/.

Kriuchkova, Anastasiia, Varvara Toloknova, dan Svitlana Drin. 2024. "Predictive model for a product without history using LightGBM. Pricing model for a new product." *Mohyla Mathematical Journal*. https://consensus.app/papers/predictive-model-for-a-product-without-history-using-kriuchkova-toloknova/e3d20aad05565c20bfc081df93d09006/.

Liu, Gang, Xuehong Yang, dan Minzan Li. 2005. "An artificial neural network model for crop yield responding to soil parameters." *Lecture Notes in Computer Science* 3498(III): 1017–21.

Malek, Nur Hanisah Abdul dkk. 2023. "Comparison of ensemble hybrid sampling with bagging and boosting machine learning approach for imbalanced data." *Indonesian Journal of Electrical Engineering and Computer Science* 29(1): 598–608.

Meshram, Vishal dkk. 2021. "Machine learning in agriculture domain: A state-of-art survey." *Artificial Intelligence in the Life Sciences* 1(October).

Muhammad, Aldi Cahya dkk. 2023. "Dasar-dasar Pembelajaran Mesin." : 131.

Nanda Putri Cintari, Dkk. 2024. "Analisis Perbandingan Kinerja Metode Ensemble Bagging dan Boosting pada Klasifikasi Bantuan Subsidi Listrik di Kabupaten/Kota Bogor." *Indonesian Journal of Computer Science* 13(6): 284–301. http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135.

Nugraha, Wahyu, dan Agung Sasongko. 2022. "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search." *SISTEMASI: Jurnal Sistem*

Informasi 11(2): 2540–9719. https://doi.org/10.32520/stmsi.v11i2.1750.

Nuriati, Irma, Budi Serasi Ginting, dan Yani Maulita. 2021. "Sistem Pendukung Keputusan Pemilihan Jenis Tanaman Pangan Berdasarkan Kondisi Tanah dengan Metode Moora." *Seminar Nasional Informatika*: 285–94.

Rakuasa, Heinrich, Dzaka Ashriel Faris, dan Muh Hidayatullah. 2024. "Transforming education in the age of artificial intelligence: Challenges and opportunities in Indonesia, a literature review." *Journal Education Innovation E-ISSN* 2(1): 180–86. https://jurnal.ypkpasid.org/index.php/jei.

Rangkuti, Fauzan Asyraf, Siti Sundari, Teknik Informatika, dan Universitas Harapan Medan. 2025. "IMPLEMENTASI GRADIENT BOOSTING MACHINES UNTUK PREDIKSI HARGA IMPLEMENTATION OF GRADIENT BOOSTING MACHINES FOR HOUSE PRICE PREDICTION IN SOUTH JAKARTA." 4(2): 164–72.

Rousseeuw, P, dan M Hubert. 2011. "Robust statistics for outlier detection." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1. https://consensus.app/papers/robust-statistics-for-outlier-detection-hubert-rousseeuw/e093ffc6c5535f59aea1698b9b0ef6e8/.

Septiana Rizky, Putri, Ristu Haiban Hirzi, dan Umam Hidayaturrohman. 2022. "Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang." *J Statistika: Jurnal Ilmiah Teori dan Aplikasi Statistika* 15(2): 228–36.

Setyarini, Dela Ananda, Agnes Ayu Maharani Dyah Gayatri, Christian Sri Kusuma Aditya, dan Didih Rizki Chandranegara. 2024. "Stroke Prediction with Enhanced Gradient Boosting Classifier and Strategic Hyperparameter." *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer* 23(2): 477–90.

Singh, G, D Bhattacharyya, dan A Bandyopadhyay. 2023. "Robust estimation strategy for handling outliers." *Communications in Statistics - Theory and Methods* 53: 5311–30. https://consensus.app/papers/robust-estimation-strategy-for-handling-outliers-singh-bandyopadhyay/620d027de38c504c8245eec32b6743ca/.

Singh Mohan, Devendra dkk. 2023. “International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING IoT Framework for Precision Agriculture: Machine Learning Crop Prediction.” *Original Research Paper International Journal of Intelligent Systems and Applications in Engineering IJISAE* 2023(5s): 300–313. www.ijisae.org.

Smith, Pete dkk. “Status of the World ’ s Soils.” : 73–104.

Vargo, Alexander, Fan Zhang, M Yurochkin, dan Yuekai Sun. 2021. “Individually Fair Gradient Boosting.” *ArXiv* abs/2103.1. https://consensus.app/papers/individually-fair-gradient-boosting-sun-zhang/fc95f069f23f588ca9e7d1d54766a6a3/.

Wardhana, Indrawata, Musi Ariawijaya, Vandri Ahmad Isnaini, dan Rahmi Putri Wirman. 2022. “Gradient Boosting Machine, Random Forest dan Light GBM untuk Klasifikasi Kacang Kering.” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 6(1): 92–99.

Zeng, Jinshan, Min Zhang, dan Shao Bo Lin. 2022. “Fully corrective gradient boosting with squared hinge: Fast learning rates and early stopping.” *Neural Networks* 147: 136–51. https://www.sciencedirect.com/science/article/abs/pii/S0893608021004950 (Juli 22, 2025).