Laporan Akhir Tugas MK. Data Mining (KOM1338), Semester Genap 2024/2025
Program Studi Sarjana Ilmu Komputer
SSMI IPB

Klasifikasi Multilabel Jenis Tanaman dan Pupuk untuk Pertanian Presisi Menggunakan Algoritma Random Forest

Rizkika Deviyanti (G6401221007) , Viby Ladyscha Yalasena Winarno (G6401221025), Zara Zannetta (G6401221033), Jesika Oktaviani (G6401221038)

Kelompok: 2, Kelas Paralel: 3

Abstrak

Dengan hadirnya inovasi teknologi di masa kini, sektor pertanian mengalami kemajuan dengan menerapkan konsep pertanian presisi. Salah satu penerapan pertanian presisi adalah dalam pemilihan jenis pupuk yang sesuai dengan jenis tanaman dan kondisi lingkungan tertentu. Para petani dalam hal menentukan jenis pupuk untuk tanaman membutuhkan waktu yang cukup lama dan kurang cepat karena harus meneliti secara manual tentang komposisi dan kriteria yang diinginkan. Sehingga, diperlukan suatu sistem yang mampu mengolah data lingkungan secara otomatis dan memberikan rekomendasi yang tepat mengenai jenis tanaman dan pupuk yang sesuai dengan kondisi lahan. Penelitian ini menerapkan klasifikasi multilabel menggunakan algoritma *Random Forest* untuk memberikan rekomendasi yang sesuai berdasarkan data kondisi lingkungan lahan. Permasalahan ketidakseimbangan kelas pada dataset diatasi dengan teknik *oversampling* guna meningkatkan performa model. Selain itu, data outlier ditangani menggunakan metode *capping* untuk menjaga kualitas data. Hasil dari algoritma menunjukkan akurasi yang tinggi, yaitu sebesar 99% untuk label *crop* dan 93% untuk label *fertilizer*. Secara keseluruhan, nilai *precision, recall*, dan *F1-score* untuk kedua label berada di atas 0.9 secara rata-rata. Hal ini menunjukkan bahwa algoritma *Random Forest* efektif diterapkan dalam sistem rekomendasi jenis tanaman dan pupuk berdasarkan kondisi lingkungan lahan.

Kata Kunci**:** Klasifikasi**,** Multilabel, Pupuk, *Random Forest*, Tanaman.

PENDAHULUAN

Latar Belakang

Sumber daya alam mencakup berbagai sektor penting seperti kelautan, pertambangan, dan pertanian. Di antara sektor-sektor tersebut, pertanian memiliki peran penting dalam perekonomian negara serta dalam menunjang kebutuhan hidup masyarakat. Dengan hadirnya inovasi teknologi di masa kini, sektor pertanian mengalami kemajuan dengan menerapkan konsep pertanian presisi. Konsep dasar pertanian presisi adalah penggunaan input seakurat mungkin sesuai kebutuhan tanaman, sehingga diperoleh keuntungan berupa penghematan dalam pembiayaan input, tenaga kerja, dan hasil panen yang lebih baik. Penerapan pertanian presisi memungkinkan diterapkan mulai dari pendekatan sederhana hingga tingkat yang lebih kompleks, tentunya akan diikuti oleh konsekuensi perbedaan keakurasian dan besaran investasi instrumen teknologi yang digunakannya (Pitono 2020). Salah satu penerapan pertanian presisi adalah dalam pemilihan jenis pupuk yang sesuai dengan jenis tanaman dan kondisi lingkungan tertentu.

Pemilihan pupuk yang tepat dapat meningkatkan efisiensi pemupukan serta berdampak positif terhadap produktivitas hasil pertanian.

Namun, dalam praktiknya, pemilihan jenis pupuk yang tepat masih menjadi tantangan bagi sebagian besar petani. Para petani dalam hal menentukan jenis pupuk untuk tanaman membutuhkan waktu yang cukup lama dan kurang cepat. Karena harus meneliti secara manual tentang komposisi dan kriteria yang diinginkan (Imanuloh *et al*. 2022). Selain itu, kondisi lahan yang memiliki keragaman tanah dan iklim juga dapat mempengaruhi pemilihan jenis pupuk yang tepat. Salah satu aspek yang dibutuhkan seorang petani untuk mempunyai hasil panen salah satu tumbuhan yang baik yaitu data parameter kondisi pH tanah, suhu dan kelembaban lingkungan sekitar (Rukmana *et al*. 2019). Oleh karena itu, diperlukan suatu sistem yang mampu mengolah data lingkungan secara otomatis dan memberikan rekomendasi yang tepat mengenai jenis tanaman dan pupuk yang sesuai dengan kondisi lahan. Sistem rekomendasi berbasis *data mining* dapat menganalisis data seperti pH tanah, suhu, curah hujan, dan faktor lingkungan lainnya untuk menentukan pilihan pupuk dan tanaman yang paling optimal. Sistem ini tidak hanya meningkatkan efisiensi dalam pemupukan, tetapi juga berkontribusi pada peningkatan hasil panen dengan memastikan tanaman memperoleh nutrisi yang sesuai kebutuhannya. Selain itu, penerapan sistem berbasis data mining dapat mengurangi kesalahan dalam pengambilan keputusan, mempercepat proses pemilihan pupuk dan tanaman, serta mendukung pengelolaan lahan pertanian secara lebih efektif dan berkelanjutan.

Tujuan

Tujuan dari penelitian ini adalah membangun model klasifikasi yang sesuai untuk sistem rekomendasi jenis tanaman dan pupuk berdasarkan kondisi lingkungan lahan.

Ruang Lingkup

Dataset yang digunakan berasal dari situ Kaggle dengan judul "*Crop and Fertilizer Dataset for Western Maharashtra*" yang akan diolah menggunakan model klasifikasi dengan algoritma *Random Forest*.

Manfaat

Manfaat dari penelitian ini adalah untuk memperoleh model terbaik yang dapat digunakan sebagai sistem rekomendasi jenis tanaman dan pupuk berdasarkan kondisi lingkungan lahan, sehingga dapat mendukung penerapan pertanian presisi.

TINJAUAN PUSTAKA

Penggunaan Machine Learning dalam Pertanian Presisi

Pertanian presisi merupakan strategi pertanian yang dilakukan dengan mengumpulkan, memproses, dan menganalisis data temporal, spasial, dan individual sebagai dasar pengambilan keputusan yang tepat untuk meningkatkan efisiensi penggunaan sumber daya, hasil panen, kualitas, keuntungan, dan keberlanjutan produksi pertanian (Beneduzzi *et al.* 2022). Seiring pertumbuhan penduduk dan terbatasnya lahan pertanian, sektor pertanian menghadapi tantangan dalam memenuhi kebutuhan pangan. Untuk mengatasi masalah ini, pertanian presisi mulai mengadopsi teknologi cerdas seperti *Internet of Things* (IoT) dan *Artificial Intelligence* (AI). Sistem ini memungkinkan pemantauan data seperti kelembaban tanah, suhu, dan nutrisi melalui sensor, serta penerapan algoritma *machine learning* untuk menentukan keputusan yang optimal dan meningkatkan hasil produksi (Sharma *et al.* 2020).

Algoritma *Random Forest*

Random Forest merupakan algoritma klasifikasi yang terdiri atas sejumlah d*ecision tree* di mana setiap *tree* dibentuk secara independen menggunakan vektor sampel acak yang memiliki distribusi identik dan seragam untuk seluruh *tree* dalam model. Dalam proses klasifikasi, *Random Forest* membentuk beberapa pohon keputusan, sedangkan untuk prediksi, hasilnya diperoleh melalui rata-rata prediksi seluruh pohon (Lingga 2017). Metode ini merupakan pengembangan dari algoritma *Classification and Regression Tree* (CART) dengan menerapkan pendekatan *bagging* (*bootstrap aggregation*) dan pemilihan fitur secara acak. Pendekatan *bagging* berperan dalam meningkatkan performa klasifikasi dengan membentuk beberapa model dari data yang berbeda-beda hasil *bootstrap* (Mahmuda 2024). Langkah-langkah algoritma *Random Forest* menurut Mahmuda (2024) adalah sebagai berikut:

1. Mengambil n sampel data dari *dataset* utama menggunakan metode *bootstrap resampling* (dengan pengembalian).
2. Membangun pohon keputusan dari setiap sampel *bootstrap* dan memilih variabel pemisah terbaik dari subset fitur yang dipilih secara acak. Jumlah fitur acak ini umumnya ditentukan dengan rumus seperti $\sqrt{m}$, $2\sqrt{m}$, atau $\frac{1}{2}\sqrt{m}$, di mana m adalah jumlah total fitur prediktor.
3. Melakukan klasifikasi terhadap data berdasarkan pohon yang telah dibentuk.
4. Ulangi langkah 1–3 hingga jumlah pohon yang diinginkan terbentuk.
5. Menggabungkan hasil prediksi dari semua pohon untuk menghasilkan klasifikasi akhir.
6. Jumlah fitur prediktor (m) yang digunakan untuk membagi simpul pada pohon sangat mempengaruhi hasil, semakin besar nilainya, semakin tinggi pula korelasi antar pohon yang terbentuk.

Menurut Mahmuda (2024), dalam proses membangun pohon pada metode CART, digunakan perhitungan *information gain* untuk menentukan atribut terbaik dalam membagi data. *Information gain* mengukur sejauh mana suatu variabel berkontribusi dalam memisahkan kelas pada simpul tertentu dalam pohon keputusan. Misalnya simpul N digunakan untuk membagi *dataset* D berdasarkan variabel yang memiliki nilai *information gain* tertinggi. Rumus yang digunakan adalah sebagai berikut:

$$Informasi\ Gain\ (IG) = entropi(D) - entropi_A(D) \quad (1)$$

Nilai $entropi(D)$ dapat diperoleh dengan menggunakan rumus pada persamaan (2) dan nilai $entropi_A(D)$ dapat diperoleh melalui rumus pada persamaan (3).

$$Entropi(D) = -\Sigma_{i=1}^{c} p_i^2 log(p_i) \quad (2)$$

dimana:
c : jumlah kelas target (variabel prediktor)
p_i : proporsi kelas ke-i partisi D

$$Entropi(D) = \Sigma_{j=1}^{v} \frac{n_j}{n} \times entropi(D_j) \quad (3)$$

dimana:
v : jumlah partisi
n : jumlah observasi
n_j : jumlah observasi j
D : partisi ke-j

Proses partisi setiap iterasi pada algoritma pohon klasifikasi pada dasarnya mencari metode partisi yang memberikan informasi *gain* tertinggi.

Random Forest termasuk algoritma machine learning yang tidak sensitif terhadap multikolinearitas (Susetyoko *et al*. 2022). Multikolinearitas terjadi ketika adanya inter-korelasi yang tinggi antara variabel-variabel independen (Boonprong *et al*. 2018). Ketika multikolinearitas terjadi, standard error dari koefisien regresi akan meningkat, yang dapat menyebabkan beberapa variabel yang seharusnya signifikan menjadi terlihat tidak signifikan secara statistik (Daoud 2017). Algoritma Random Forest cocok digunakan pada data yang memiliki banyak variabel independen yang berkorelasi tinggi karena algoritma ini tidak mengasumsikan distribusi data maupun hubungan linear antar variabel independen pada data (Boonprong *et al*. 2018).

Metrik Evaluasi

Metrik evaluasi yang digunakan untuk mengevaluasi model setelah dilakukan pelatihan terdiri dari:

a) *Confusion Matrix*

Tabel 1.1 *Confusion Matrix*

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Menurut Swaminathan dan Tantri (2024), *Confusion Matrix* adalah matriks persegi berukuran N × N, dengan N merupakan jumlah kelas keluaran (*output*). Tiap baris pada matriks merepresentasikan jumlah nilai yang diprediksi dan tiap kolom merepresentasikan jumlah nilai sebenarnya. *Confusion matrix* menunjukkan performa dari model, misalnya banyak prediksi yang benar atau banyak kesalahan yang dilakukan. Banyak nilai yang ditunjukkan terbagi menjadi empat tipe:

(1) *True Positive* (TP): Model memprediksi hasil positif secara benar. Hasil prediksi dan hasil sebenarnya adalah positif.
(2) *False Positive* (FP): Model salah memprediksi hasil positif. Hasil prediksi bernilai positif, sedangkan hasil sebenarnya adalah negatif. Biasa disebut dengan Type I error.
(3) *True Negative* (TN): Model memprediksi hasil negatif secara benar. Hasil prediksi dan hasil sebenarnya adalah negatif.
(4) *False Negative* (FN): Model salah memprediksi hasil negatif. Hasil prediksi bernilai negatif, sedangkan hasil sebenarnya adalah positif. Biasa disebut dengan Type II error.

b) *Accuracy*

Menurut Swaminathan dan Tantri (2024), *Accuracy* atau akurasi adalah seberapa sering prediksi yang dihasilkan suatu model bernilai benar. Akurasi berasal

dari perbandingan antara jumlah hasil prediksi yang benar dengan total prediksi. Akurasi dapat dihitung dengan rumus:

$$Accuracy = \frac{(TP + TN)}{(TP+FP+TN+FN)} \quad (4)$$

c) *F1-Score*

Menurut Swaminathan dan Tantri (2024), *F1-Score* adalah metrik evaluasi yang dihasilkan dari gabungan *precision* dan *recall. Precision* adalah bagian dari hasil positif yang diprediksi dengan benar. Metrik ini menunjukkan seberapa *confident* suatu model dalam memprediksi kelas positif sebagai positif. *Precision* dapat dihitung dengan rumus:

$$Precision = \frac{TP}{(TP+FP)} \quad (5)$$

Sedangkan, *recall* adalah metrik yang mengukur seberapa akurat suatu model memprediksi hasil positif. *Recall* dapat dihitung dengan rumus:

$$Recall = \frac{TP}{(TP+FN)} \quad (6)$$

F1-Score didapatkan dari rata-rata harmonis dari *precision* dan *recall*. Rata-rata harmonis ini tidak hanya memperhatikan nilai prediksi yang benar, tetapi juga memperhatikan nilai prediksi yang salah. Metrik *F1-Score* dapat dihitung dengan rumus:

$$F1\ Score = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \quad (7)$$

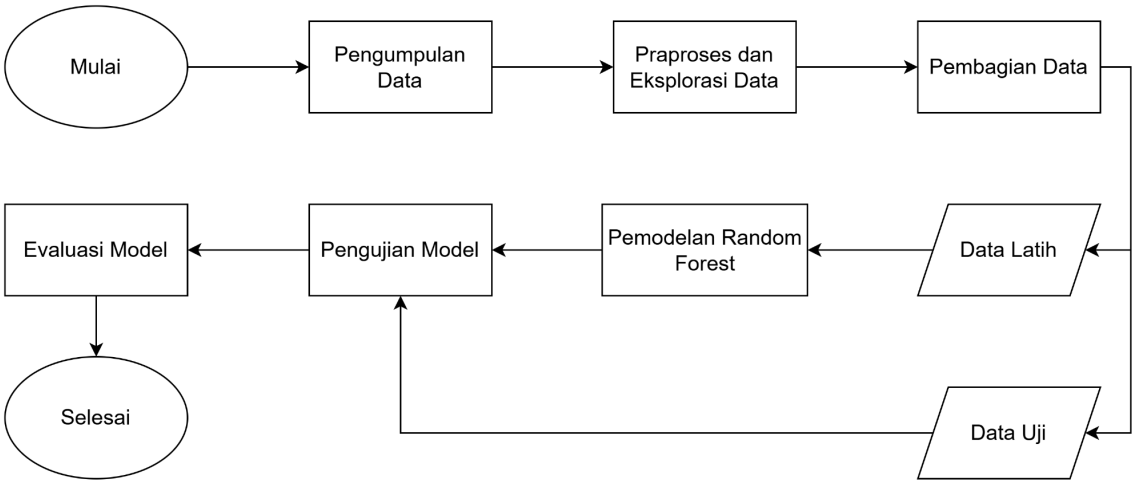
METODE

Data

Penelitian ini menggunakan data sekunder yang diunduh dari situs *dataset online* Kaggle dengan judul "*Crop and Fertilizer Dataset for Western Maharashtra*". *Dataset* ini terdiri dari 4513 baris dengan 11 atribut. Atribut-atribut tersebut meliputi berbagai variabel di antaranya nama distrik (*District Name*), warna tanah (*Soil Color*), kadar unsur hara (Nitrogen, Phosphorus, Potassium), tingkat keasaman tanah (pH), curah hujan (*Rainfall*), suhu lingkungan (*Temperature*), jenis tanaman yang cocok ditanam (*Crop*), serta jenis pupuk yang sesuai (*Fertilizer*). Selain itu, terdapat atribut *Link* yang memuat tautan video YouTube mengenai cara merawat tanaman. Namun, atribut *District Name* dan *Link* tidak digunakan dalam penelitian ini karena tidak relevan serta dianggap tidak berkontribusi langsung terhadap sistem rekomendasi yang dikembangkan. *Dataset* disimpan dalam file dengan format .CSV yang kemudian akan digunakan dalam proses pengolahan data.

Tahapan Kegiatan

Penelitian dilakukan melalui beberapa tahapan, antara lain pengumpulan data, praproses dan eksplorasi data, pembagian(*splitting*) data, pemodelan menggunakan *Random Forest*, pengujian model, dan evaluasi model. Alur tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1 Tahapan Kegiatan

1. Pra-proses dan Eksplorasi Data

Pada tahap ini kita melakukan serangkaian proses dan eksplorasi data untuk memastikan kualitas data sebelum digunakan dalam pemodelan. Langkah pertama yang dilakukan adalah menghapus fitur-fitur yang tidak relevan untuk mengurangi redudansi dan meningkatkan efisiensi analisis pada data. Selanjutnya, dilakukan identifikasi dan penanganan terhadap data kosong baik *null* atau *NaN* dengan strategi yang sesuai. Data duplikat juga dihapus untuk menghindari bias pada hasil analisis. Deteksi outlier dilakukan menggunakan metode *Interquartile Range* (IQR), kemudian dilakukan imputasi dengan *capping* atau penghapusan terhadap nilai-nilai ekstrem tersebut. Setelah itu, dilakukan normalisasi terhadap fitur numerik dengan menggunakan *MinMaxScaler* untuk menyelaraskan skala nilai seluruh fitur. Selain itu, dilakukan juga penanganan data yang tidak seimbang pada label dengan menggunakan teknik *random oversampling* dengan memanfaatkan fungsi *RandomOverSampler* untuk menambah sampel data-data kelas minoritas untuk menghindari kemungkinan model tidak belajar dengan baik pada kelas minoritas. Pendekatan ini bertujuan untuk meningkatkan performa model dalam mendeteksi data minoritas, terutama pada kasus dengan distribusi kelas yang tidak seimbang.

2. Pembagian Data

Setelah melalui tahap pra proses, data kemudian dibagi menjadi dua bagian yaitu data latih dan data uji. Data latih digunakan untuk membangun dan melatih model agar dapat mengenali pola dari data, sedangkan data uji digunakan untuk melakukan evaluasi kinerja model yang telah dibuat. Pada penelitian ini, pembagian data dilakukan dengan proporsi sebesar 80% untuk data latih dan sebesar 20% untuk data uji. Parameter *random state* di-*set* dengan nilai 42 untuk memastikan konsistensi terhadap hasil pembagian data.

3. Pemodelan

Pemodelan dalam penelitian ini dilakukan dengan menggunakan algoritma *Random Forest Classifier*, yaitu metode ensemble yang membangun banyak pohon keputusan untuk meningkatkan akurasi dan kestabilan prediksi. Karena permasalahan yang dihadapi melibatkan lebih dari satu variabel target, digunakan pendekatan *MultiOutputClassifier* yang memungkinkan model menangani keluaran ganda secara simultan namun terstruktur. Dalam proses pelatihannya, beberapa parameter penting seperti *n_estimators* (jumlah

pohon), *max_depth* (kedalaman maksimum pohon), dan *min_samples_split* (jumlah minimum sampel untuk membagi simpul) dioptimalkan untuk memperoleh performa model yang maksimal. Selain itu, dilakukan juga proses *hyperparameter tuning* untuk menemukan kombinasi parameter terbaik, yang tidak hanya meningkatkan akurasi model tetapi juga meminimalkan risiko *overfitting* pada model yang digunakan, sehingga model dapat bekerja dengan baik dan akurat pada data baru.

4. Pengujian Model

Setelah proses pembangunan model selesai, tahap selanjutnya adalah melakukan pengujian terhadap model yang telah dibangun. Pengujian model dilakukan dengan menggunakan data uji yang telah dipisahkan sebelumnya. Pengujian ini bertujuan untuk mengevaluasi kemampuan model dalam melakukan prediksi terhadap data baru secara objektif. Model yang telah dilatih diimplementasikan dalam fungsi prediksi berbasis Python. Fungsi ini menerima input berupa data baru yang sudah diproses dan memberikan prediksi yang relevan untuk setiap target. Pengujian akhir ini memberikan gambaran tentang performa model dalam menghadapi data yang belum pernah dilihat sebelumnya, sehingga mencerminkan potensi penerapannya dalam konteks dunia nyata.

5. Evaluasi Model

Evaluasi model dilakukan dengan menggunakan *confusion matrix* untuk mengukur distribusi kesalahan klasifikasi pada setiap kelas. Selain itu, digunakan juga *classification report* untuk menghitung metrik evaluasi seperti *precision*, *recall*, *f1-score*, dan akurasi. Analisis terhadap kemungkinan terjadinya *overfitting* dilakukan dengan membandingkan kinerja model pada data latih dan data uji. Apabila ditemukan indikasi *overfitting*, maka dilakukan penyesuaian parameter model untuk meningkatkan generalisasi dan kinerja model secara keseluruhan.

Lingkungan Pengembangan

Dalam pengembangan penelitian, *Jupyter Notebook*, *Google Colab*, dan *VS Code* digunakan sebagai *Integrated Development Environment* (IDE) utama. Ketiga IDE tersebut digunakan untuk proses eksplorasi data, pra-proses data, pemodelan, serta evaluasi model. Berbagai pustaka Python juga digunakan untuk mendukung proses analisis dan pengembangan model *machine learning*, antara lain *NumPy* dan *Pandas* untuk manipulasi data, *Matplotlib.pyplot* dan *Seaborn* untuk visualisasi, serta *MinMaxScaler* untuk normalisasi fitur. Untuk pemodelan, digunakan algoritma *Random Forest Classifier* yang diimplementasikan melalui *MultiOutputClassifier* karena project ini memprediksi dua label. Pembagian data dilakukan menggunakan *train_test_split*, sementara evaluasi performa model dilakukan dengan menggunakan *accuracy_score*, *confusion_matrix*, dan *classification_report*. Penanganan ketidakseimbangan data dilakukan melalui metode *Random Oversampling* menggunakan library *imblearn.over_sampling*.

HASIL DAN PEMBAHASAN

Hasil Pra-Proses Data

1. Penanganan Nilai Null dan Duplikat

   Pertama, fitur-fitur yang tidak relevan dihapus untuk mengurangi redundansi dalam analisis. Pada project ini, variabel yang dihapus adalah *Districk_Name* dan *Link*. Data kosong (null atau NaN) diidentifikasi dan ditangani dengan strategi yang sesuai, sementara data duplikat dihapus untuk menghindari bias pada analisis. Namun, pada

dataset yang digunakan tidak terdapat data yang kosong atau duplikat, sehingga tidak perlu menghapus baris apapun pada dataset.

2. Analisis Statistik Deskriptif Data

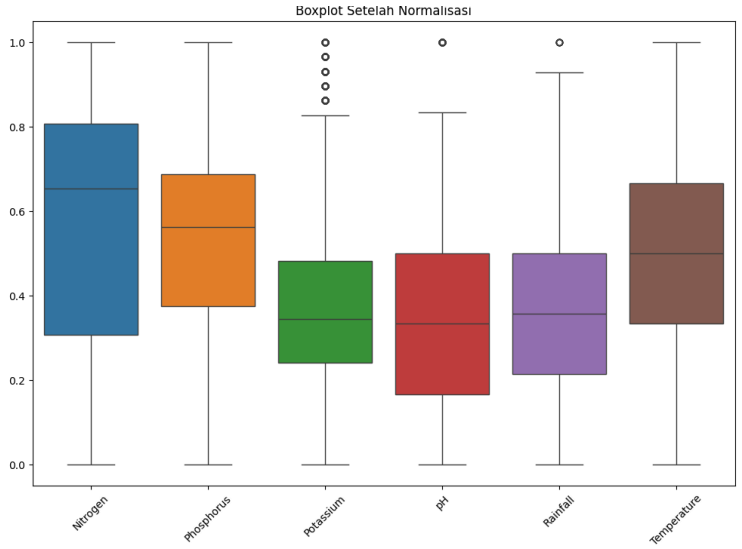
   Berdasarkan deskripsi statistik, data yang dianalisis memiliki ukuran yang cukup besar dengan 4.513 sampel untuk setiap variabel. Kandungan unsur hara seperti Nitrogen, Phosphorus, dan Potassium menunjukkan variasi yang cukup tinggi, terutama pada Nitrogen dan Potassium, yang mengindikasikan adanya perbedaan kebutuhan nutrisi antar lokasi atau tanaman. Nilai pH tanah rata-rata berada di kisaran 6,7 yang bisa dikatakan tergolong netral dan ideal untuk pertumbuhan sebagian besar tanaman, meskipun terdapat data dengan pH cukup asam hingga cukup basa. Curah hujan memiliki rata-rata sebesar 819 mm namun dengan sebaran yang luas (300–1700 mm), hal ini mencerminkan variasi iklim yang signifikan di wilayah pengambilan data. Sementara itu, suhu berkisar antara 10°C hingga 40°C dengan rata-rata 25,9°C, menunjukkan lingkungan yang umumnya tropis namun tetap memiliki rentang suhu yang luas. Secara keseluruhan, data ini mencerminkan kondisi agrikultur yang beragam dan cocok untuk eksplorasi lebih lanjut dalam pemodelan prediksi berbasis lingkungan.

3. Pemisahan Fitur Kategorik dan Fitur Numerik

   Pada tahap eksplorasi data, dilakukan identifikasi terhadap nilai unik pada fitur kategorikal, yaitu *Fertilizer*, *Crop*, dan *Soil_color*. Hal ini penting untuk memahami variasi dan distribusi kategori yang ada dalam data. Selanjutnya, fitur dalam dataset diklasifikasikan menjadi dua jenis, yaitu fitur numerik yang terdiri dari Nitrogen, Phosphorus, Potassium, pH, *Rainfall*, dan *Temperature*, serta fitur kategorikal yang meliputi *Soil_color* yang merupakan fitur input, serta *Crop* dan *Fertilizer* yang merupakan variabel target. Pemisahan ini bertujuan untuk mempermudah proses praproses dan pemodelan data sesuai dengan jenis masing-masing variabel.

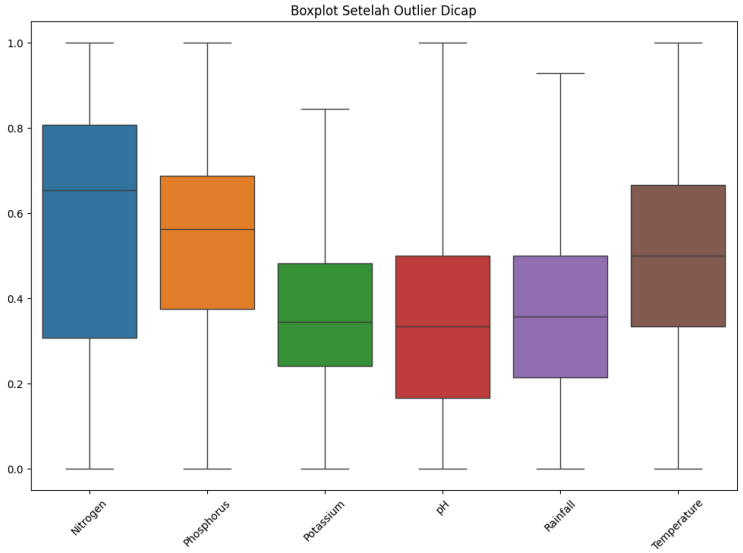
4. Penanganan Outlier

   Sebelum melakukan penanganan outlier, data yang bersifat numerik divisualisasikan menggunakan boxplot terlebih dahulu untuk melihat apakah data memiliki outlier. Ternyata setelah dilakukan visualisasi dengan menggunakan boxplot, rentang data dengan salah satu variabel terbilang cukup jauh sehingga perlu dilakukan normalisasi terlebih dahulu pada dataset untuk menyamakan skala tiap variabel. Setelah dilakukan normalisasi, outlier pada setiap variabel numerik semakin terlihat sehingga diperlukan penanganan.



Gambar 2 Outlier setelah dilakukan normalisasi

Pada penelitian ini, kami menggunakan teknik *capping* untuk menangani outlier yang ada. Outlier yang signifikan diidentifikasi menggunakan metode *Interquartile Range* (IQR) dan dilakukan *capping* atau penghapusan terhadap nilai-nilai ekstrem tersebut. Pendekatan yang digunakan adalah berbasis *Interquartile Range (IQR)*, yaitu selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1) yang merepresentasikan rentang nilai tengah dari data tanpa memperhatikan outlier. Dari hasil perhitungan IQR, batas bawah (*lower bound*) dan batas atas (*upper bound*) ditentukan menggunakan rumus *Lower Bound*=Q1−1.5×IQR dan *Upper Bound*=Q3+1.5×IQR. Data yang berada di bawah *lower bound* atau di atas *upper bound* dianggap sebagai outlier.



Gambar 3 Outlier setelah dibersihkan dengan teknik *capping*

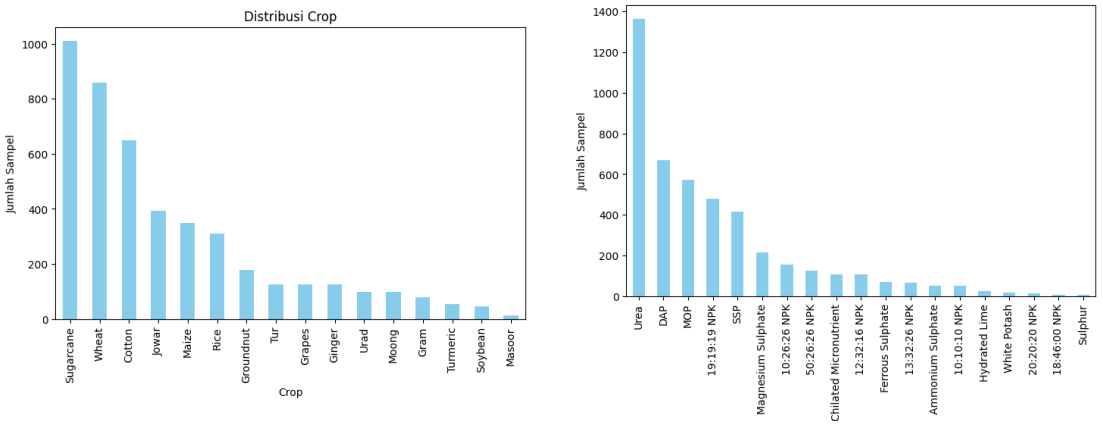
5. Normalisasi Data Numerik

Untuk memastikan bahwa semua fitur numerik berada dalam skala yang seragam dan tidak mendominasi satu sama lain dalam proses pemodelan, dilakukan normalisasi menggunakan *MinMaxScaler*. Proses ini mengubah nilai setiap fitur numerik ke dalam rentang nilai di antara 0 hingga 1. Fitur yang dinormalisasi meliputi Nitrogen, Phosphorus, Potassium, pH, *Rainfall,* dan *Temperature*. Hasil normalisasi kemudian

disimpan dalam bentuk *DataFrame* baru untuk digunakan pada tahap analisis dan pemodelan selanjutnya.

6. Analisis Univariate untuk Fitur Kategorik

   Untuk menangani data yang tidak seimbang, kita perlu melihat sebaran kelas target (label) yang mana kedua variabel target tersebut merupakan variabel yang sifatnya kategorik. Setelah dilakukan visualisasi distribusi, terlihat bahwa jumlah sampel pada masing-masing kelas setiap variabel target tidak seimbang, sehingga diperlukan penanganan lebih lanjut untuk menghindari bias dalam memprediksi data yang distribusi kelasnya minoritas pada model.



Gambar 4 Distribusi fitur *Crop* dan *Fertilizer*

7. *Label Encoding*

   Agar data dapat diproses oleh model *machine learning*, kita perlu untuk mengubah fitur kategorik menjadi format numerik terlebih dahulu. Untuk mengubah data kategorik menjadi bentuk numerik yang dapat digunakan oleh algoritma *machine learning*, dilakukan *encoding* pada fitur '*Soil_color*' dan kedua variabel target menggunakan *LabelEncoder* dari pustaka *sklearn.preprocessing*. Proses ini mengonversi setiap kategori unik pada kolom kategorik menjadi representasi numerik yang sesuai. Hasilnya, fitur tersebut kini dapat digunakan bersama fitur numerik lainnya dalam proses pelatihan model.

Splitting Data

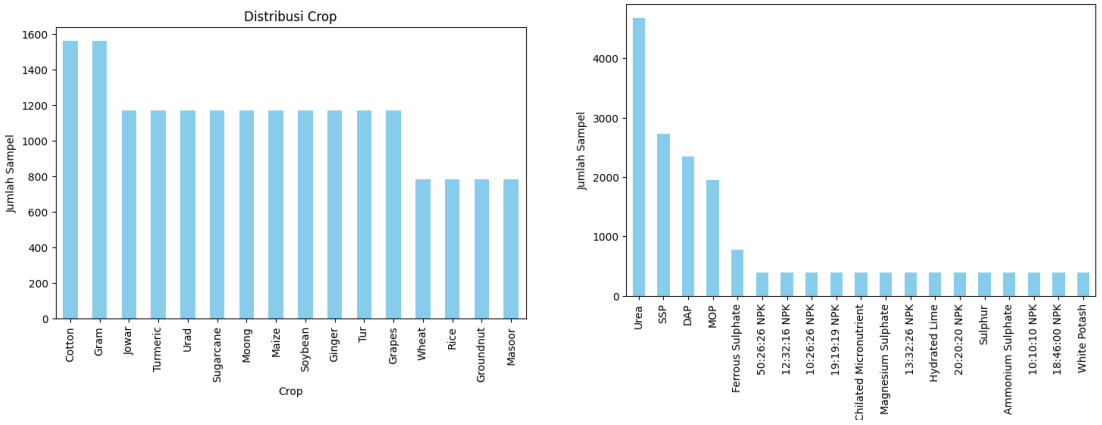
Data dibagi menjadi dua bagian menggunakan fungsi *train_test_split*, yaitu 80% data untuk pelatihan (*X_train, y_train*) dan 20% untuk pengujian (*X_test, y_test*). Tujuan dari pembagian ini adalah untuk mengevaluasi performa model secara objektif terhadap data yang belum pernah dilihat selama proses pelatihan.

Oversampling Data

Setelah melakukan pembagian data, data latih (*X_train* dan *y_train*) digabung menjadi satu *DataFrame* untuk keperluan *oversampling*. Hal tersebut dikarenakan fungsi *RandomOverSampler* tidak dapat bekerja pada data multidimensional, oleh karena itu kita perlu menggabungkan datanya terlebih dahulu. *Oversampling* diperlukan karena dataset bersifat tidak seimbang (*imbalance*), sehingga beberapa kombinasi label *Crop* dan *Fertilizer* mungkin memiliki jumlah sampel yang jauh lebih sedikit dibanding lainnya.

Untuk mengatasi hal ini, dibuat label gabungan sementara dengan menggabungkan kedua label (*Crop* dan *Fertilizer*) menjadi satu string, misalnya "*Wheat_Organic*".

Oversampling dilakukan menggunakan teknik *RandomOverSampler* dari *library* imblearn, yang akan menyeimbangkan jumlah sampel untuk setiap kombinasi label gabungan tersebut. Setelah proses *oversampling*, hasilnya dipisahkan kembali menjadi fitur (*X_train_resampled*) dan target *multi-output* (*y_train_resampled*) yang terdiri dari dua kolom yaitu *Crop* dan *Fertilizer*.



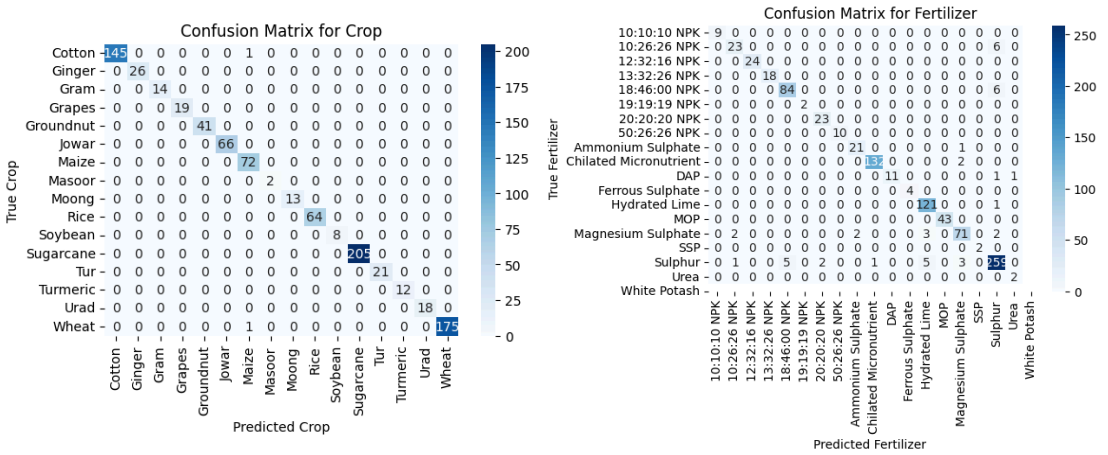
Gambar 5 Distribusi fitur *Crop* dan *Fertilizer* hasil *oversampling*

Pelatihan Model

Selanjutnya, dilakukan pelatihan model menggunakan *Random Forest Classifier* yang dibungkus dalam *MultiOutputClassifier*. Model ini digunakan karena target prediksi terdiri dari dua output berbeda secara bersamaan. Parameter seperti *max_depth*, *max_features*, dan *min_samples_split* dikonfigurasi untuk menghindari *overfitting* dan meningkatkan generalisasi model. Setelah model dilatih, dilakukan prediksi terhadap data pengujian (*X_test*). Hasil prediksi disimpan dalam *DataFrame y_pred_df* dan kemudian didekode menggunakan LabelEncoder untuk mengembalikan hasil prediksi ke bentuk format label aslinya, bukan angka. Demikian pula, data *y_test* yang merupakan label asli juga dilakukan *decode* untuk keperluan evaluasi.

Evaluasi Model

Setelah proses pelatihan model selesai, dilakukan evaluasi performa prediksi terhadap dua label target (*Crop* dan *Fertilizer*) dengan menggunakan *confusion matrix* dan *classification report*. *Confusion matrix* memberikan visualisasi jumlah prediksi yang benar dan salah untuk masing-masing kelas. Matriks ini ditampilkan dalam bentuk *heatmap*, yang memudahkan dalam mengamati seberapa akurat model dalam membedakan antar kelas. Dua *confusion matrix* dibuat secara terpisah untuk label *Crop* dan *Fertilizer* dengan sumbu vertikal merepresentasikan aktual dan sumbu horizontal merepresentasikan prediksi.



Gambar 6 *Confusion matrix* label *Crop* dan *Fertilizer*

Selanjutnya, *classification report* disajikan untuk masing-masing label guna memberikan evaluasi lebih mendalam. Laporan ini mencakup metrik-metrik penting seperti *precision, recall, f1-score*, dan jumlah *support* untuk setiap kelas. Pada label *Crop*, model menunjukkan performa sangat tinggi dengan akurasi total sebesar 99%. Hampir semua kelas memiliki *f1-score* di atas 0.98 yang menunjukkan bahwa model mampu mengenali berbagai jenis tanaman dengan sangat baik. Hal ini membuktikan bahwa proses pelatihan dan *oversampling* berhasil meningkatkan kinerja klasifikasi terhadap label ini.

Classification Report for Crop:

	precision	recall	f1-score	support
Cotton	1.00	0.99	1.00	146
Ginger	0.96	1.00	0.98	26
Gram	1.00	1.00	1.00	14
Grapes	1.00	1.00	1.00	19
Groundnut	1.00	1.00	1.00	41
Jowar	1.00	0.97	0.98	66
Maize	0.97	1.00	0.99	72
Masoor	1.00	1.00	1.00	2
Moong	0.87	1.00	0.93	13
Rice	1.00	1.00	1.00	64
Soybean	1.00	1.00	1.00	8
Sugarcane	1.00	1.00	1.00	205
Tur	1.00	1.00	1.00	21
Turmeric	1.00	1.00	1.00	12
Urad	1.00	1.00	1.00	18
Wheat	1.00	0.99	0.99	176
accuracy			0.99	903
macro avg	0.99	1.00	0.99	903
weighted avg	0.99	0.99	0.99	903

Gambar 7 *Classification report* label *Crop*

Pada label *Fertilizer*, model mencatat akurasi sebesar 93%. Beberapa kelas seperti "10:26:26 NPK", "Chilated Micronutrient", dan "SSP" memiliki *f1-score* yang lebih rendah dibanding kelas lainnya, kemungkinan karena jumlah sampel yang lebih sedikit atau kesamaan pola fitur antar jenis pupuk. Meskipun demikian, model secara keseluruhan tetap menunjukkan performa yang kuat, dengan *precision* dan *recall* yang tinggi di sebagian besar kelas. Evaluasi ini menegaskan bahwa pendekatan *multioutput classifier* yang dikombinasikan dengan *oversampling* cukup efektif dalam menangani prediksi dua label kategorikal.

Classification Report for Fertilizer:

	precision	recall	f1-score	support
10:10:10 NPK	1.00	1.00	1.00	9
10:26:26 NPK	0.88	0.76	0.81	29
12:32:16 NPK	0.96	1.00	0.98	24
13:32:26 NPK	1.00	1.00	1.00	18
19:19:19 NPK	0.91	0.88	0.89	90
20:20:20 NPK	1.00	1.00	1.00	2
50:26:26 NPK	0.92	0.96	0.94	23
Ammonium Sulphate	1.00	1.00	1.00	10
Chilated Micronutrient	0.89	0.77	0.83	22
DAP	0.98	0.96	0.97	134
Ferrous Sulphate	1.00	0.92	0.96	13
Hydrated Lime	1.00	1.00	1.00	4
MOP	0.92	0.95	0.94	122
Magnesium Sulphate	1.00	1.00	1.00	43
SSP	0.84	0.88	0.86	80
Sulphur	1.00	1.00	1.00	2
Urea	0.93	0.94	0.93	276
White Potash	1.00	1.00	1.00	2
accuracy			0.93	903
macro avg	0.96	0.95	0.95	903
weighted avg	0.93	0.93	0.93	903

Gambar 8 *Classification report* label *Fertilizer*

Fungsi Prediksi Jenis Tanaman

Fungsi *predict_new_data* dibuat untuk memudahkan prediksi jenis tanaman (*Crop*) dan pupuk (*Fertilizer*) yang sesuai berdasarkan data input baru berupa kandungan Nitrogen, Phosphorus, Potassium, nilai pH, curah hujan, suhu, dan warna tanah. Pertama, data input diubah menjadi sebuah *DataFrame* dengan format yang sama seperti data latih. Selanjutnya, kolom kategorikal *Soil_color* yang merupakan satu-satunya variabel non-numerik diolah menggunakan *one-hot encoding* agar dapat diinterpretasikan oleh model. Kemudian kolom-kolom dalam *DataFrame* ini disesuaikan dan diurutkan agar sama dengan data fitur saat model dilatih. Setelah itu, nilai-nilai numerik pada data tersebut distandarisasi menggunakan objek *scaler* yang sudah dipelajari pada proses *training* agar skala fitur konsisten. Dengan data yang sudah siap, model kemudian melakukan prediksi dan menghasilkan output berupa label numerik untuk tanaman dan pupuk. Label tersebut kemudian dikonversi kembali ke bentuk aslinya menggunakan *label encoder*, sehingga menghasilkan rekomendasi tanaman dan pupuk yang mudah dipahami. Contohnya, saat diberikan input seperti kandungan Nitrogen 200, Phosphorus 15, Potassium 30, pH 6.5, curah hujan 100, suhu 25, dan warna tanah hitam, model memprediksi tanaman yang cocok adalah *Sugarcane* dan pupuk yang direkomendasikan adalah Urea. Hal ini menunjukkan bahwa model mampu memberikan rekomendasi yang relevan dan dapat digunakan untuk membantu pengambilan keputusan dalam pertanian berdasarkan kondisi tanah dan lingkungan.

KESIMPULAN DAN SARAN

Algoritma *Random Forest* dapat diterapkan pada sistem rekomendasi jenis tanaman dan pupuk berdasarkan kondisi lingkungan tanah dengan pendekatan *multilabel classification*. *Random Forest* merupakan algoritma klasifikasi yang terdiri atas sejumlah d*ecision tree* di mana setiap *tree* dibentuk secara independen menggunakan vektor sampel acak yang memiliki distribusi identik dan seragam untuk seluruh *tree* dalam model. Untuk label *Crop*, model menunjukkan akurasi sangat tinggi sebesar 99%, dengan hampir semua kelas memiliki *f1-score* di atas 0.98. Hal ini menunjukkan bahwa model mampu mengenali berbagai jenis tanaman secara akurat. Sementara itu, untuk label *Fertilizer* memiliki akurasi model mencapai 93%. Meskipun demikian, beberapa kelas seperti "10:26:26

NPK", "*Chilated Micronutrient*", dan "SSP" memiliki *f1-score* yang lebih rendah dibanding kelas lainnya. Hal ini kemungkinan disebabkan oleh jumlah data yang tidak seimbang atau kemiripan pola fitur antar jenis pupuk. Secara keseluruhan, nilai *precision*, *recall*, dan *f1-score* untuk kedua label berada di atas 0.9 secara rata-rata. Ini membuktikan bahwa model mampu mengklasifikasikan kedua label dengan akurasi tinggi dan dapat dimanfaatkan untuk mendukung pengambilan keputusan di bidang pertanian berdasarkan kondisi tanah dan lingkungan.

Saran yang dapat diberikan adalah melakukan *oversampling* tambahan pada label *Fertilizer*, khususnya untuk kelas-kelas yang memiliki *f1-score* rendah, agar distribusi data menjadi lebih seimbang. Selain itu, disarankan untuk mengeksplorasi penggunaan algoritma lain untuk membandingkan performa dan melihat apakah ada peningkatan akurasi. Penambahan variasi data pelatihan juga penting agar model dapat belajar dari pola yang lebih kompleks dan mengurangi *overfitting*.

DAFTAR PUSTAKA

Beneduzzi H, Souza E, Oliveira W, Sobjak R, Bazzi C, Rodrigues M. 2022. FERTILIZER RECOMMENDATION METHODS FOR PRECISION AGRICULTURE – A SYSTEMATIC LITERATURE STUDY. *Engenharia Agrícola*. 42(1). doi:10.1590/1809-4430-eng.agric.v42n1e20210185/2022.

Boonprong S, Cao C, Chen W, Bao S. 2018. Random Forest Variable Importance Spectral Indices Scheme for Burnt Forest Recovery Monitoring—Multilevel RF-VIMP. *Remote Sensing*. 10(6): 807. doi: 10.3390/rs10060807.

Daoud J I. 2017. Multicollinearity and Regression Analysis. *Journal of Physics: Conference Series*. 949. doi: 10.1088/1742-6596/949/1/012009.

Imanuloh NR, Kuswulandari R, Listiani T, Hartanti D. 2022. Sistem Pendukung Keputusan Pemilihan Pupuk Terbaik untuk Tanaman Padi Di Desa Panggisari dengan Metode Fuzzy. *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB) 2022*.

Lingga P R F, Fatichah C, & Purwitasari D. 2017. Deteksi Gempa Berdasarkan Data Twitter Menggunakan Decision Tree, Random Forest, dan SVM. *Jurnal Teknik ITS*. 6(1): 153-158.

Mahmuda S. 2024. Implementasi Metode Random Forest pada Kategori Konten Kanal Youtube. *Jurnal Jendela Matematika*. 2(1): 21-31.

Pitono J. 2020. Pertanian Presisi Dalam Budidaya Lada. *National Research and Innovation Agency: The Precision Farming on Pepper Cultivation*. 18(2):91. doi:10.21082/psp.v18n2.2019.91-103.

Rukmana A, Susilawati H, Galang. 2019. Pencatat pH Tanah Otomatis. *Journal Universitas Garu*t. 10(1):25.

Sharma A, Jain A, Gupta P, Chowdary V. 2020. Machine Learning Applications for Precision Agriculture: A Comprehensive Review. *IEEE Access*. 1(1). doi:10.1109/ACCESS.2020.3048415.

Susetyoko R, Yuwono W, Purwantini E, Ramadijanti N. 2022. Perbandingan Metode Random Forest, Regresi Logistik, Naïve Bayes, dan Multilayer Perceptron Pada Klasifikasi Uang Kuliah Tunggal (UKT). *Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan.* 7(1): 8-16. doi: 10.30811/jim.v7i1.2916.

Swaminathan S, Tantri B R. 2024. Confusion Matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*. 27(4): 4023-4031. doi:10.53555/AJBR.v27i4S.4345.