

Perbandingan Efektivitas Metode *K-Nearest Neighbor* dan *Naive Bayes* dalam Data Pengamatan Kesehatan Tanaman

Nur Aizah*¹, Ahmad Homaidi², Lukman Fakhid Lidimilah³

^{1,2,3} Universitas Ibrahimy Situbondo

Email: ¹aizaheshal39@gmail.com, ²ahmadhomaidi@ibrahimiy.ac.id, ³lukmanfakhidlidimilah@ibrahimiy.ac.id

*Penulis Korespondensi

Abstrak

Tujuan dari penelitian ini yakni untuk membandingkan efektivitas dua metode klasifikasi, *K-Nearest Neighbor* (K-NN) dan *Naive Bayes*, dalam memantau kesehatan tanaman berdasarkan data lingkungan seperti suhu, kelembapan, intensitas cahaya, dan kandungan unsur hara lainnya. Kesehatan tanaman merupakan kondisi fisik dan fisiologis yang mencerminkan kemampuan tanaman untuk tumbuh dan berkembang secara optimal, yang dipengaruhi oleh faktor biotik dan abiotik serta interaksi dengan mikroorganisme di sekitar *rizosfer*. Ketidakseimbangan unsur hara, stres lingkungan, dan keterbatasan sistem pemantauan tradisional yang bersifat subjektif sering menyebabkan kerugian ekonomi dan mengancam ketahanan pangan akibat tidak adanya pengetahuan mengenai gejala stress tanaman sehingga terjadilah kesalahan penanganan. Penelitian ini menggunakan metode CRISP-DM untuk memfasilitasi proses analisis data secara terstruktur, mulai dari identifikasi kebutuhan hingga implementasi hasil. Data yang digunakan pada penelitian ini merupakan data sekunder yang diperoleh dari studi pustaka dan *repository online platform* Kaggle. Data yang dikumpulkan dianalisis menggunakan teknik deskriptif kuantitatif untuk menilai kinerja masing-masing algoritma. Hasil penelitian menunjukkan bahwa *Naive Bayes* mencapai akurasi lebih tinggi sebesar 76,25%, sementara K-NN menunjukkan akurasi sebesar 52,92%. Hasil ini menunjukkan bahwa metode *Naive Bayes* dengan pendekatan berbasis probabilistik lebih efektif dalam memantau kesehatan tanaman dan dapat digunakan sebagai solusi dalam pengelolaan pertanian berbasis teknologi. Penelitian ini diharapkan dapat mendukung pengambilan keputusan yang lebih tepat dan meningkatkan produktivitas di bidang pertanian.

Kata kunci: Klasifikasi, *K-Nearest Neighbor*, *Naive Bayes*, Kesehatan Tanaman, CRISP-DM.

Abstract

The purpose of this study is to compare the effectiveness of two classification methods, *K-Nearest Neighbor* (K-NN) and *Naive Bayes*, in monitoring plant health based on environmental data such as temperature, humidity, light intensity, and other nutrient content. Plant health is a physical and physiological condition that reflects the ability of plants to grow and develop optimally, which is influenced by biotic and abiotic factors and interactions with microorganisms around the rhizosphere. Nutrient imbalances, environmental stress, and the limitations of traditional subjective monitoring systems often cause economic losses and threaten food security due to the absence of knowledge about the symptoms of plant stress, resulting in mishandling. This research uses the CRISP-DM method to facilitate the data analysis process in a structured manner, from identification of needs to implementation of results. The data used in this research is secondary data obtained from literature study and online repository of Kaggle platform. The data collected was analysed using quantitative descriptive techniques to assess the performance of each algorithm. The results showed that *Naive Bayes* achieved a higher accuracy of 76.25%, while K-NN showed an accuracy of 52.92%. These results indicate that the *Naive Bayes* method with a probabilistic-based approach is more effective in monitoring plant health and can be used as a solution in technology-based agricultural management. This research is expected to support more informed decision-making and increase productivity in agriculture.

Keywords: Classification, *K-Nearest Neighbor*, *Naive Bayes*, Plant Health, CRISP-DM.

I. PENDAHULUAN

Secara biologis, kesehatan tanaman dipengaruhi oleh interaksi genetik dengan mikroorganisme di *rizosfer* serta faktor lingkungan seperti kelembapan dan suhu tanah [1]. Unsur hara esensial seperti nitrogen, fosfor, dan kalium penting untuk pertumbuhan organ tanaman, sementara kandungan klorofil mencerminkan efektivitas fotosintesis dalam menghasilkan energi. Pertanian merupakan sektor yang sangat vital dalam mendukung ketahanan pangan dan pembangunan ekonomi suatu negara. Menurut *Food and Agriculture Organization* (FAO), sekitar 40% produksi tanaman pangan global hilang setiap tahun akibat hama dan penyakit [2]. Dalam era modern saat ini, terdapat kebutuhan mendesak untuk meningkatkan efisiensi dan akurasi dalam memantau kondisi kesehatan tanaman. Berdasarkan penelitian sebelumnya oleh Putu Prianka Vedanty, bahwa sistem pemantauan tradisional yang mengandalkan pengamatan visual secara

manual seringkali memiliki keterbatasan dan cenderung bersifat subjektif, terutama dalam mendeteksi gejala awal penyakit atau stres tanaman secara cepat dan akurat[3]. Kondisi ini mendorong pengembangan teknologi berbasis data dan algoritma cerdas yang mampu memberikan informasi *real-time* dan lebih dapat diandalkan, guna mendukung pengambilan keputusan yang tepat di lapangan.

Penelitian ini mengacu pada konsep pengklasifikasian data yang mencakup metode *K-Nearest Neighbor* (K-NN) dan *Naïve Bayes*. Kedua metode dipilih karena umum digunakan sebagai *baseline* dengan karakteristik khusus keduanya.

Tabel 1. Algoritma Klasifikasi yang Umum Digunakan

Algoritma	Penggunaan	Alasan
<i>Naïve Bayes</i> (NB)	Sangat umum	Cepat, berbasis probabilistik, sederhana, cocok untuk data besar & teks[4]
<i>K-Nearest Neighbor</i> (KNN)	Sangat umum	Intuitif, non-parametrik, efektif untuk data numerik, tidak memerlukan pelatihan awal[5]
<i>C4.5 (Decision Tree)</i>	Kadang	Mudah diinterpretasikan, tapi memerlukan proses pruning dan pemisahan atribut kontinu[6]
<i>Support Vector Machine</i>	Tidak umum	Kompleks, memerlukan <i>tuning</i> parameter seperti kernel dan C, lebih cocok untuk evaluasi lanjutan[7]
<i>Random Forest</i> (RF)	Tidak umum	Akurat dan kuat terhadap <i>overfitting</i> , tetapi merupakan model <i>ensemble</i> yang lebih kompleks[8]
<i>Neural Network</i> (NN)	Tidak umum	Butuh data besar, tuning banyak parameter, dan proses training yang berat[9]
<i>Logistic Regression</i> (LR)	Kadang digunakan	Sederhana dan interpretable, tetapi hanya efektif untuk masalah klasifikasi linier sederhana[10]

K-NN adalah algoritma yang bekerja berdasarkan kedekatan data menggunakan jarak antar data sebagai parameter utama dalam pengelompokkan ke dalam kategori tertentu [11]. Sementara itu, *Naïve Bayes* adalah metode probabilistik yang mengasumsikan independensi fitur dan menghitung kemungkinan kelas berdasarkan Teorema Bayes[12]. Kedua metode ini memiliki keunggulan dan kelemahan masing-masing, tergantung dari karakteristik data yang digunakan. Pemilihan metode yang tepat sangat penting dalam aplikasi nyata, terutama dalam klasifikasi kondisi kesehatan tanaman yang kompleks dan memiliki berbagai faktor pengaruh.

Klasifikasi kesehatan tanaman merupakan salah satu pendekatan utama dalam teknologi pertanian berbasis data [13], yang memanfaatkan fitur numerik dari data lingkungan seperti suhu, kelembapan, intensitas cahaya, dan kandungan unsur hara. Penelitian terdahulu menunjukkan bahwa kombinasi data sensor dan algoritma pembelajaran mesin mampu meningkatkan akurasi pendeteksian stres dan penyakit tanaman [14]. Penelitian ini terbatas pada penggunaan data sekunder melalui *platform* daring seperti Kaggle dan studi Pustaka yang menjadi sumber utama untuk memperkaya basis data dan memperkuat landasan analisis. Oleh karena itu, penelitian ini berfokus pada perbandingan kinerja kedua algoritma tersebut dalam mengklasifikasi kondisi kesehatan tanaman secara efektif dan efisien.

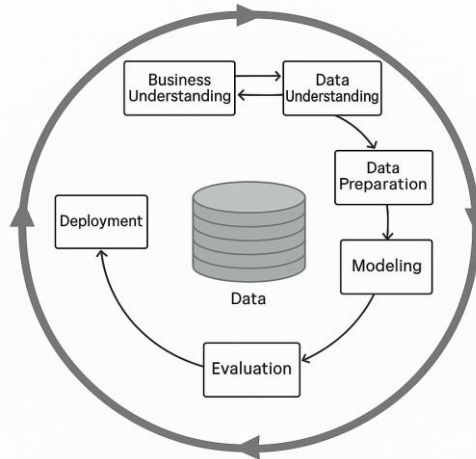
Dalam konteks perbaikan sistem pemantauan pertanian, hasil dari penelitian ini diharapkan dapat memberikan gambaran yang jelas mengenai keunggulan dan kelemahan masing-masing metode tertentu. Selain itu, penelitian ini juga bertujuan untuk menyajikan solusi praktis yang dapat diimplementasikan petani dan praktisi pertanian guna meningkatkan produktivitas serta keberlanjutan usaha tani. Dengan mengkombinasikan landasan teori dari algoritma klasifikasi dan pemanfaatan data lingkungan, diharapkan analisis ini mampu memberikan kontribusi nyata dalam pengembangan teknologi pertanian berbasis data dan mendukung implementasi sistem pemantauan otomatis yang akurat dan responsif.

II. METODE PENELITIAN

2.1. CRISP-DM

Program ini dibangun menggunakan *Python*, yakni bahasa pemrograman tingkat tinggi dalam ilmu data, pemrograman pembelajaran mesin (*machine learning*), analisis data, dan automasi[15]. Beberapa pustaka yang digunakan diantaranya *Numpy*, *Pandas*, dan *Scikit-Learn* dengan metode perancangan berupa pendekatan *Cross Industry Standard Process for Data Mining* (CRISP-DM) sebagai kerangka metodologis dalam proses pengembangan sistem berbasis data. CRISP-DM merupakan metodologi yang telah banyak digunakan dalam proyek data mining karena memberikan alur kerja yang sistematis, menyeluruh, dan fleksibel [16]. Pendekatan ini dipilih karena mampu memfasilitasi proses analisis data secara terstruktur, mulai dari identifikasi kebutuhan hingga implementasi hasil. Dengan adanya tahapan yang jelas, pengembangan sistem dapat dilakukan secara lebih terarah dan sesuai dengan tujuan yang ingin dicapai.

CRISP-DM terdiri atas enam tahapan utama, yaitu pemahaman bisnis (*business understanding*), pemahaman data (*data understanding*), persiapan data (*data preparation*), pembuatan model (*modeling*), evaluasi model (*evaluation*), dan penyebaran hasil (*deployment*) [17]. Tahapan awal bertujuan untuk memahami konteks permasalahan dan kebutuhan pengguna. Selanjutnya, dilakukan eksplorasi dan pembersihan data agar siap digunakan dalam proses pemodelan. Setelah model dibangun dan dievaluasi, tahap akhir berupa *deployment* akan menerapkan hasil analisis ke dalam sistem operasional, baik dalam bentuk visualisasi dan laporan sehingga dapat memberikan manfaat nyata dalam mendukung pengambilan keputusan berbasis data.



Gambar 1. Metode Pengembangan Sistem CRISP-DM

Penelitian ini akan membandingkan dua model dan akan dipilih model terbaik untuk memprediksi label data baru. Misalkan akan ditentukan label untuk data dengan kelembapan tanah 29,896, suhu udara 25,749, suhu tanah 21,357, kelembapan udara 46,014, intensitas cahaya 795,427, pH tanah 7,081, kadar nitrogen 27,858, fosfor 11,412, dan kalium 45,996, kandungan klorofil 39,454, serta sinyal elektrokimia 1,170 terhadap data *training* berikut.

Soil_Mo	Ambient	Soil_Te	Humidity	Light_In	Soil_pH	Nitrogen	Phosph	Potassi	Chlorop	Electrod	Plant_Health	Stat
25,45979	26,81113	19,22304	60,37614	200,6155	7,168728	12,28895	31,97778	13,37533	21,06721	1,325766	High Stress	
17,44073	18,00415	22,92222	53,55764	201,5074	7,219353	45,15537	42,84077	26,11206	20,64199	1,885605	High Stress	
27,59583	27,41347	19,03552	51,94968	201,7362	6,134535	24,55637	22,44584	15,66657	22,76363	0,396653	Moderate Stress	
25,78286	18,04927	23,28845	68,05006	202,507	7,027235	25,65234	48,49655	48,99837	45,65059	1,078309	Moderate Stress	
15,90995	25,91057	24,04474	49,21475	203,6865	6,087579	48,2365	40,59411	10,40538	31,51414	1,766865	High Stress	
39,24293	29,39286	19,18861	61,67028	203,4999	6,001142	19,6177	24,06266	24,71117	36,92426	0,221544	Moderate Stress	
16,20029	27,03876	15,97143	68,87799	204,5532	7,18711	17,95629	34,01862	18,65085	31,17413	0,161749	High Stress	
26,75173	24,41406	20,62668	55,77615	204,469	5,922037	23,08455	37,80488	20,31639	31,17959	0,910027	Moderate Stress	
17,54174	27,64655	15,05866	57,56474	206,1472	6,045506	21,492	14,29134	31,78319	23,373	1,754981	High Stress	
37,11222	20,91563	18,90661	42,20773	207,812	6,903838	32,04791	48,7437	38,75347	28,44228	0,985969	Healthy	
33,2634	20,63105	16,9678	45,71981	208,3444	6,927091	25,00695	15,70055	30,72398	36,64744	0,234616	Healthy	

Gambar 2. Data Training pada Ms. Excel

2.2. K-Nearest Neighbor

Metode *K-Nearest Neighbor* (K-NN) merupakan metode pembelajaran mesin yang bersifat non-parametrik dan berbasis pembelajaran malas (*lazy learning*) [9]. Sifat non-parametrik pada algoritma ini berarti bahwa K-NN tidak mengasumsikan pola tertentu pada distribusi data yang digunakan. Dengan demikian, algoritma ini tidak memerlukan jumlah parameter tetap atau estimasi parameter tertentu dalam modelnya, baik untuk dataset dengan ukuran kecil maupun besar [18]. Algoritma *K-Nearest Neighbors* (KNN) bekerja dengan prinsip kemiripan, yaitu memprediksi label atau nilai data baru berdasarkan K data terdekat dalam data latih. Untuk melakukannya, algoritma menghitung jarak antara data baru dan seluruh data latih menggunakan metrik seperti *Euclidean* dengan rumus berikut.

$$(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2}$$

$$d1: \sqrt{(25,459 - 29,896)^2 + (26,811 - 25,749)^2 + (19,223 - 21,357)^2 + (60,376 - 46,014)^2 + (200,615 - 795,427)^2 + (7,168 - 7,081)^2 + (12,288 - 27,858)^2 + (31,977 - 11,412)^2 + (31,977 - 13,375)^2 + (31,977 - 21,067)^2 + (1,325 - 1,170)^2}$$

$$= \sqrt{19,680 + 1,127 + 4,555 + 206,259 + 353801,330 + 0,008 + 338,088 + 0,024} = 596,742$$

Setelah seluruh jarak dihitung, algoritma memilih K titik terdekat dan menentukan prediksi berdasarkan informasi dari titik-titik tersebut. Pada klasifikasi, label yang paling sering muncul di antara tetangga dipilih sebagai hasil prediksi. Sedangkan dalam regresi, nilai prediksi diambil dari rata-rata nilai tetangga terdekat.

Soil_Mo	Ambient_T	Soil_Temp	Humidity	Light_Int	Soil_pH	Nitrogen_L	Phosphorus	Potassium	Chlorophyll	Electrode	Plant_Health	ED
29,896	25,74948	21,3573	46,01442	795,4275	7,08124	27,85795	11,41228	45,99614	39,45438	1,170426	Moderate Stre	0,00
25,16119	21,05296	23,89919	43,2144	802,8853	6,714822	32,60994	22,6718	43,3252	30,77659	1,574195	Moderate Stre	18,61
27,57085	25,27331	15,84832	45,64127	809,501	6,156364	14,89557	18,40243	49,29361	26,08387	0,737459	High Stress	25,33
37,50098	18,63248	23,25282	44,87563	780,8644	6,200589	35,57616	12,52882	43,54167	23,24058	1,525663	Healthy	25,61
21,53198	22,38562	24,02625	42,01806	782,6989	5,571931	28,82248	19,69957	42,34869	21,70796	0,974661	Moderate Stre	25,82
19,23712	27,99517	19,35361	61,33708	795,288	7,255786	35,55745	20,16809	33,49533	34,93556	0,770933	High Stress	25,89

Gambar 3. Hasil Perhitungan ED dengan nilai K=5

Berdasarkan hasil sortir dengan nilai K=5 tersebut, kedekatan data latih (X) terhadap data baru (Y) dominan dan mengindikasikan status *moderate stress*.

2.3. Naïve Bayes

Algoritma *Naïve Bayes* adalah metode pembelajaran mesin berbasis probabilistik yang menggunakan Teorema *Bayes* dengan asumsi bahwa setiap fitur dalam dataset bersifat independen (*naïve*) [12]. Sebagai algoritma parametris, model ini mengestimasi parameter, seperti rata-rata dan *varians*, yang tetap meskipun ukuran dataset bertambah. *Naïve Bayes* unggul dalam efisiensi pemrosesan dataset besar dan tetap memberikan hasil akurat meskipun asumsi independensi tidak sepenuhnya terpenuhi.

- Penentuan Probabilitas Prior

Probabilitas prior menunjukkan seberapa sering suatu kelas muncul dalam dataset secara keseluruhan.

$$P(C/health) = \frac{\text{Jumlah data kelas health (299)}}{\text{Total data (1200 data)}} = 0,25$$

Perhitungannya dilakukan dengan membandingkan jumlah data dalam suatu kelas dengan jumlah total data dalam dataset, contohnya pada salah satu kelas *health*. Sehingga didapatkan $P(C/health) : 0.25$, $P(C/moderate) : 0.33$, dan $P(C/high) : 0.42$ yang dihitung dengan rumus serupa.

- Perhitungan *Likelihood* untuk Setiap Fitur

Untuk setiap fitur dalam dataset, akan dihitung probabilitas *likelihood* berdasarkan distribusi fitur tersebut dalam setiap kelas. Misalnya, untuk data kontinu, digunakan distribusi *Gaussian* dengan rumus:

$$P(x|C) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

μ adalah rata-rata dan σ^2 adalah varian fitur untuk kelas tertentu. Rumus rata-rata pada excel menggunakan AVERAGE, sedangkan untuk mencari varian menggunakan VAR. berikut hasil perhitungan rata-rata dan varian fitur menggunakan excel.

MEAN											
Soil_Moist	Ambient_T	Soil_Temp	Humidity	Light_Intensity	Soil_pH	Nitrogen_L	Phosphorus	Potassium	Chlorophyll	Electrode	LABEL
25,0093	23,9984	19,9563	54,8343	613,7125	6,5251	30,0356	30,2800	30,1324	34,7337	0,9884	HEALTH
25,0433	23,9966	19,9593	54,8473	612,9408	6,5257	30,0590	30,2678	30,1343	34,7273	0,9862	MODERATE
25,1050	23,9966	19,9625	54,8713	613,1298	6,5244	30,1003	30,2585	30,1370	34,7493	0,9889	HIGH
VARIAN											
Soil_Moist	Ambient_T	Soil_Temp	Humidity	Light_Intensity	Soil_pH	Nitrogen_L	Phosphorus	Potassium	Chlorophyll	Electrode	LABEL
9,2221	12,5254	9,0602	77,3381	53011,7970	0,3721	78,7388	128,8312	136,9386	76,9036	0,3188	HEALTH
20,1063	11,2777	9,1107	75,9406	51072,9625	0,3401	115,1711	134,6407	130,7395	77,8960	0,3202	MODERATE
50,6775	11,9358	7,9287	77,9893	52571,1942	0,3167	156,0861	129,9199	138,9152	76,2584	0,3473	HIGH

Gambar 3. Hasil Perhitungan Rata-rata dan Varian Fitur

Kemudian akan dihitung *Likelihood* untuk salah satu fitur yakni *Soil_Moisture* dalam kelas *Health* menggunakan rumus *Gaussian* diatas. Rumus ini juga digunakan pada seluruh fitur kelas lain.

$$P(\text{soil moist}|\text{health}) = \frac{1}{\sqrt{2 * 3.14 * 9.2221}} * \exp\left(-\frac{(29,896 - 25.0093)^2}{2 * 9.2221}\right) = 0,13136 * 0,27397 = 0,0360$$

- Perhitungan Probabilitas Posterior

$$P(C|X) = \frac{p(X|C) * P(C)}{P(X)}$$

Seluruh *likelihood* dan *prior* digabungkan menggunakan teorema *Bayes*. Karena $P(X)$ adalah konstan untuk semua kelas, fokus perhitungan berada pada pembilang $P(C)*P(X|C)$. Nilai setiap kelas akan dihitung dan dipilih dengan nilai probabilitas tertinggi sebagai prediksi. Sehingga diperoleh nilai probabilitas untuk kelas *health* : 2,21053E-16, *moderate* = 0,00000E+00, dan *high* = 2,50498E-14.

III. HASIL DAN PEMBAHASAN

3.1. Pemahaman Bisnis (*Business Understanding*)

Tabel 2. Pemahaman Bisnis

Aspek	Deskripsi
Tujuan Bisnis	Membandingkan algoritma K-NN dan <i>Naïve Bayes</i> dalam klasifikasi kesehatan tanaman berdasarkan 1.200 baris data sensor untuk menentukan model terbaik.
Kebutuhan Bisnis	• Mengembangkan sistem klasifikasi berbasis machine learning yang akurat dan efisien. • Menyediakan bukti kuantitatif (akurasi, precision, recall, F1-score). • Menghasilkan model dengan prediksi kepercayaan tinggi untuk mendukung intervensi cepat.
Permasalahan	• Pengamatan visual tidak konsisten dan rawan kesalahan.
KPI (<i>Key Performance Indicator</i>)	• Akurasi klasifikasi $\geq 90\%$ → mengurangi kesalahan diagnosis. • Latency prediksi ≤ 1 detik → siap digunakan di sensor edge. • Peningkatan produktivitas lahan $\geq X\%$ setelah 1 musim.

3.2. Pemahaman Data (*Data Understanding*)

Data yang digunakan dalam penelitian merupakan data sekunder yang diperoleh dari repositori *online* aplikasi Kaggle, yaitu *Plant Health Data* dengan tautan akses <https://www.kaggle.com/datasets/ziya07/plant-health-data> yang berisi 1200 data biosensor terkait pemantauan kesehatan tanaman, yang mencakup pengukuran 11 parameter lingkungan dan fisiologis yang penting untuk menilai kesehatan tanaman, seperti kelembaban tanah, suhu, kelembaban udara, intensitas cahaya, kadar nutrisi, dan sinyal stres tanaman. Setiap baris dalam dataset ini mewakili pembacaan tertentu untuk sebuah tanaman pada waktu yang diberikan, dengan berbagai fitur yang menangkap metrik biosensor penting dengan beberapa rentang data sebagaimana berikut.

Tabel 3. Rentang Kriteria Data

Label	Health	Moderate	Stress
<i>Soil_Moisture</i>	30-40	20-40	10-40
<i>Ambient_Temperature</i>	18-30	18-30	18-30
<i>Soil_Temperature</i>	15-25	15-25	15-25
<i>Humidity</i>	40-70	40-70	40-70
<i>Light_Intensity</i>	207-996	201,7-1000	200-1000
<i>Soil_pH</i>	5.5-7.5	5.5-7.5	5.5-7.5
<i>Nitrogen_Level</i>	20-50	15-50	10-50
<i>Phosphorus_Level</i>	10-50	10-50	10-50
<i>Potassium_Level</i>	10-50	10-50	10-50
<i>Chlorophyll_Content</i>	20-50	20-50	20-50
<i>Electrochemical_Signal</i>	0,002-1,992	0,006-1,994	0,023-1,996

3.3. Persiapan Data (*Data Preparation*)

Tabel 4. Persiapan Data

Syntaks	Stress
<pre>#hilangkan plant_id dan timestamp df = df.drop(columns = ['Timestamp', 'Plant_ID']) df.head(10)</pre>	Langkah-langkah persiapan data dimulai dengan penghapusan kolom yang tidak relevan, yaitu Timestamp dan Plant_ID, karena hanya berfungsi sebagai identifikasi dan tidak memiliki nilai prediktif terhadap status kesehatan tanaman.
<pre># Definisi Fitur (X) dan Target (y) fitur_kolom = ['Soil_Moisture', 'Ambient_Temperature', 'Soil_Temperature', 'Humidity', 'Light_Intensity', 'Soil_pH', 'Nitrogen_Level', 'Phosphorus_Level', 'Potassium_Level', 'Chlorophyll_Content', 'Electrochemical_Signal'] target_kolom = 'Plant_Health_Status' X = df[fitur_kolom] y = df[target_kolom]</pre>	Selanjutnya, dilakukan pemisahan antara fitur (X) dan target (y), di mana fitur mencakup parameter fisiologis tanaman seperti kelembaban tanah, suhu, pH, kandungan nitrogen, dan lainnya, sedangkan target adalah <i>Plant_Health_Status</i> .
<pre># Encoding Target le = LabelEncoder() y_encoded = le.fit_transform(y) # Untuk melihat mapping: print(le.classes_) print(f"Kelas asli: {le.classes_}") print(f"Kelas terencode: {np.unique(y_encoded)}")</pre>	Tahap ketiga adalah <i>encoding label target</i> menggunakan <i>LabelEncoder</i> , karena model klasifikasi membutuhkan data numerik, misalnya label " <i>Healthy</i> " diubah menjadi 0. Terakhir, dilakukan pembagian data menjadi 80% data latih dan 20% data validasi menggunakan <i>train_test_split</i> dengan parameter <i>stratify</i> , agar distribusi kelas tetap seimbang antara data latih dan data validasi

3.4. Permodelan (*Modeling*)

Pendekatan pemodelan langsung dengan menggunakan *library Python* seperti *Scikit-Learn* yang menjadi pilihan utama dalam pengembangan sistem cerdas berbasis data karena mampu menyederhanakan proses implementasi algoritma *machine learning* secara signifikan. *Library* ini menyediakan fungsi-fungsi siap pakai yang telah teruji, efisien, dan didukung oleh komunitas ilmiah, sehingga pengguna tidak perlu lagi membuat kode perhitungan manual yang rumit dan rentan kesalahan.

Tabel 5. Implementasi Model

Syntaks	Deskripsi
<pre># IMPLEMENTASI MODEL KNN # Inisialisasi model KNN (contoh dengan k=5) knn_model = KNeighborsClassifier(n_neighbors=5) # Latih model # Menggunakan X_train dan X_val secara langsung karena scaling dilewati knn_model.fit(X_train, y_train) # Prediksi pada data validasi y_pred_knn = knn_model.predict(X_val) # Evaluasi kinerja KNN and classification report print("Accuracy KNN: ", accuracy_score(y_val, y_pred_knn)) print("Classification Report KNN:") print(classification_report(y_val, y_pred_knn, target_names=le.classes_))</pre>	Pada implementasi ini, model K-NN diinisialisasi dengan parameter $k=5$, yang berarti algoritma akan mempertimbangkan lima data tetangga terdekat dalam menentukan kelas dari suatu data uji. Model dilatih menggunakan data pelatihan X_{train} dan y_{train} , kemudian dilakukan prediksi terhadap data <i>testing</i> X_{val} . Evaluasi performa dilakukan dengan menghitung akurasi dan menghasilkan <i>classification report</i> yang mencakup <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> . Algoritma K-NN tidak memerlukan proses pelatihan eksplisit karena prediksi dilakukan secara langsung menggunakan data pelatihan, sehingga cocok untuk dataset berskala kecil hingga menengah dengan fitur numerik.
<pre>#implementasi model naive bayes gnb_model = GaussianNB() gnb_model.fit(X_train, y_train) y_pred_gnb = gnb_model.predict(X_val) print("Accuracy Naive Bayes: ", accuracy_score(y_val, y_pred_gnb)) print("Classification Report Naive Bayes:") print(classification_report(y_val, y_pred_gnb, target_names=le.classes_))</pre>	Dalam implementasi ini, digunakan varian <i>Gaussian Naive Bayes</i> (<i>GaussianNB</i>) yang mengasumsikan bahwa distribusi data setiap fitur mengikuti distribusi normal. Model dilatih pada data pelatihan X_{train} dan y_{train} , kemudian digunakan untuk memprediksi kelas dari data <i>testing</i> X_{val} . Evaluasi performa dilakukan dengan menghitung tingkat akurasi dan menghasilkan <i>classification report</i> . <i>Naive Bayes</i> dikenal sebagai model yang efisien dan cepat, terutama ketika diterapkan pada dataset berukuran besar, serta memiliki performa yang cukup baik meskipun asumsi independensi fitur tidak sepenuhnya terpenuhi..

3.5. Evaluasi (Evaluation)

Tabel 6. Hasil Evaluasi Model

Laporan Evaluasi					Deskripsi
Accuracy KNN: 0.5291666666666667 Classification Report KNN: <pre> precision recall f1-score support Healthy 0.47 0.57 0.52 60 High Stress 0.66 0.67 0.67 100 Moderate Stress 0.39 0.33 0.35 80 accuracy 0.53 240 macro avg 0.51 0.52 0.51 240 weighted avg 0.52 0.53 0.52 240 </pre>					Model KNN menunjukkan performa yang kurang optimal dalam mengklasifikasikan status kesehatan tanaman, dengan akurasi hanya sebesar 52,9%. Meskipun cukup baik dalam mengenali kelas High Stress (f1-score: 0.67), model ini lemah dalam membedakan kelas Moderate Stress (f1-score: 0.35), yang berdampak pada nilai rata-rata keseluruhan. Nilai macro average dan weighted average f1-score berada di kisaran 0.51–0.52, menandakan bahwa model tidak konsisten dalam menangani seluruh kelas secara seimbang.
Accuracy Naive Bayes: 0.7625 Classification Report Naive Bayes: <pre> precision recall f1-score support Healthy 0.77 0.78 0.78 60 High Stress 0.83 0.79 0.81 100 Moderate Stress 0.68 0.71 0.70 80 accuracy 0.76 240 macro avg 0.76 0.76 0.76 240 weighted avg 0.77 0.76 0.76 240 </pre>					Sebaliknya, model Naive Bayes memberikan hasil evaluasi yang jauh lebih baik dengan akurasi mencapai 76,25%. Model ini mampu mengklasifikasikan ketiga kelas dengan lebih seimbang, terutama kelas High Stress yang memiliki f1-score tertinggi sebesar 0.81. F1-score untuk kelas lainnya juga tergolong baik, dengan macro average dan weighted average keduanya berada pada nilai 0.76. Kinerja yang konsisten ini menunjukkan bahwa Naive Bayes lebih efektif dalam menangani pola distribusi data yang ada, sehingga lebih layak dijadikan model klasifikasi utama untuk kasus ini.

3.6. Penyebaran Hasil (Deployment)

Pada tahap *deployment*, model KNN dan *Naive Bayes* yang telah dilatih disimpan dalam bentuk file .pkl menggunakan pustaka *pickle*. Penyimpanan ini memungkinkan model digunakan kembali tanpa perlu proses pelatihan ulang. Model yang tersimpan kemudian digunakan untuk memprediksi status kesehatan tanaman berdasarkan data sensor terbaru. Data input berupa file CSV berisi hasil pengukuran sensor, kemudian dilakukan proses prediksi dan hasilnya disimpan dalam file `result_test_plant_data.csv`. Tahap ini sangat penting untuk mengintegrasikan model ke dalam sistem nyata yang digunakan oleh pengguna, baik melalui antarmuka aplikasi web, desktop, maupun *pipeline* otomatisasi data.

Tabel 7. Deployment

Laporan Evaluasi	Deskripsi
<pre> #saya ingin menyimpan model knn dan naive bayes import pickle pickle.dump(knn_model, open("knn_model.pkl", "wb")) pickle.dump(gnb_model, open("gnb_model.pkl", "wb")) </pre>	Model <i>K-Nearest Neighbor</i> (<code>knn_model</code>) dan <i>Naive Bayes</i> (<code>gnb_model</code>) yang telah dilatih sebelumnya disimpan dalam bentuk file .pkl menggunakan <code>pickle.dump()</code> . File <code>knn_model.pkl</code> dan <code>gnb_model.pkl</code> ini menyimpan struktur dan parameter model sehingga dapat digunakan kembali tanpa perlu melakukan pelatihan ulang. Langkah ini penting dalam <i>deployment</i> agar model siap digunakan di lingkungan produksi.
<pre> #implementasi model knn_model = pickle.load(open("knn_model.pkl", "rb")) gnb_model = pickle.load(open("gnb_model.pkl", "rb")) </pre>	Untuk menggunakan kembali model yang sudah disimpan, dilakukan proses loading dengan <code>pickle.load()</code> . Model KNN dan <i>Naive Bayes</i> dimuat dari file .pkl, dan siap digunakan untuk prediksi data baru. Ini adalah langkah kunci dalam <i>deployment</i> karena model dimanfaatkan untuk prediksi real-time atau batch di aplikasi atau sistem akhir.

IV. KESIMPULAN

Penelitian ini menunjukkan bahwa metode *Naive Bayes* lebih efektif dibandingkan *K-Nearest Neighbor* dalam hal klasifikasi kesehatan tanaman berdasarkan data lingkungan yang dianalisis. Hasil evaluasi menunjukkan bahwa *Naive Bayes* mencapai akurasi sebesar 76,25%, dengan kinerja yang lebih seimbang dalam mengenali berbagai status kesehatan tanaman, terutama dalam mendeteksi stres tinggi dengan f1-score tertinggi 0.81. Pendekatan probabilistik yang digunakan oleh *Naive Bayes* membuatnya lebih mampu mengatasi variabilitas data dan ketergantungan terhadap distribusi fitur, sehingga mampu menyediakan prediksi yang lebih akurat dan andal dalam konteks pemantauan tanaman secara otomatis. Implementasi metode ini diharapkan dapat memperkuat sistem pengelolaan pertanian berbasis data,

mendukung pengambilan keputusan yang cepat dan tepat, serta meningkatkan produktivitas pertanian secara berkelanjutan.

Hal yang dapat mengoptimalkan kinerja algoritma *Naive Bayes* yakni melalui *tuning* parameter dan fitur yang relevan, serta mempertimbangkan penggunaan teknik validasi lain seperti *cross-validation* untuk memastikan kestabilan dan generalisasi model. Selain itu, penggabungan atau perbandingan dengan algoritma lain, seperti *Random Forest* atau *Neural Network*, juga dapat dilakukan untuk mendapatkan hasil yang lebih komprehensif dan optimal dalam monitoring kesehatan tanaman berbasis data sensor.

REFERENSI

- [1] A. L. Adiredjo and L. Soetopo, *Pemuliaan Ketahanan Genetik Tanaman*. Universitas Brawijaya Press, 2021.
- [2] F. and A. Organization, "Climate change fans spread of pests and threatens plants and crops, new FAO study Pests destroy up to 40 percent of global crops and cost \$220 billion of losses." Accessed: Feb. 02, 2025. [Online]. Available: <https://www.fao.org/newsroom/detail/Climate-change-fans-spread-of-pests-and-threatens-plants-and-crops-new-FAO-study>
- [3] P. P. Vedanty, "Perbandingan Metode K-Nearest Neighbor dan Naive Bayes Dalam Identifikasi Penyakit Daun Tanaman Obat," 2024, *Universitas Pendidikan Ganesha*.
- [4] A. S. Huda, R. M. Awangga, and R. N. S. Fathonah, *Prediksi Penerimaan Pegawai Baru Dengan Metode Naive Bayes*, vol. 1. Kreatif, 2020.
- [5] M. R. Haditama, "Analisis dan pembuatan dashboard prediksi kelulusan mahasiswa menggunakan metode random forest, naïve bayes dan support vector machine," 2023, *Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta*.
- [6] M. Huda and M. Kom, *Algoritma Data Mining: Analisis Data Dengan Komputer*. bisakimia, 2019.
- [7] G. A. P. Febriyanti and A. Baita, "Comparison of Support Vector Machine and Decision Tree Algorithm Performance with Undersampling Approach in Predicting Heart Disease Based on Lifestyle," *J. Appl. Informatics Comput.*, vol. 9, no. 2, pp. 318–327, 2025.
- [8] I. Maulita and A. Wahid, "Prediksi Magnitudo Gempa Menggunakan Random Forest, Support Vector Regression, XGBoost, LightGBM, dan Multi-Layer Perceptron Berdasarkan Data Kedalaman dan Geolokasi (Predicting Earthquake Magnitude Using Random Forest, Support Vector Regression, XGBoost)," *J. Pendidik. Dan Teknol. Indones.*, vol. 4, pp. 221–232, 2024.
- [9] N. Rahayu, "Aggarwal, CC (2018). Artificial Neural Network and Deep Learning. Springer. Anderson, JA (1995). An Introduction to Neural Networks. The MIT Press. Bahdanau, D., Cho, K., & Bengio, Y.(2015). Neural Machine Translation," *Deep Learn. Teor. Algoritm. dan Apl.*, vol. 9, p. 56, 2025.
- [10] N. A. Permana and H. Bunyamin, "Perbandingan Logistic Regression dengan Random Forest dalam Memprediksi Sentimen Pada IMDb Movie Review," *J. Strateg. Maranatha*, vol. 6, no. 2, pp. 391–399, 2024.
- [11] G. S. Mahendra *et al.*, *Tren Teknologi AI: Pengantar, Teori, dan Contoh Penerapan Artificial Intelligence di Berbagai Bidang*. PT. Sonpedia Publishing Indonesia, 2024.
- [12] R. M. Sari, V. Tasril, S. Wahyuni, and S. E. Putri, *Klasifikasi Forecasting Menggunakan Algoritma Naive Bayes*. Serasi Media Teknologi, 2024.
- [13] R. M. Sari, *Klasifikasi Data Mining*. Serasi Media Teknologi, 2024.
- [14] J. R. Sitinjak, M. H. H. Ichsan, and E. Setiawan, "Penerapan Metode Naive Bayes dalam Sistem Pendeteksi Kualitas Tanah pada Tanaman Kedelai," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, pp. 2617–2622, 2023.
- [15] B. Aribowo and S. Fairuz, *Panduan Praktis Machine Learning Klasifikasi Menggunakan Python: Diandra Kreatif*. Diandra Kreatif, 2024.
- [16] G. Urva *et al.*, *PENERAPAN DATA MINING DI BERBAGAI BIDANG: Konsep, Metode, dan Studi Kasus*. PT. Sonpedia Publishing Indonesia, 2023.
- [17] U. Sa'adah, M. Y. Rochayani, D. W. Lestari, and D. A. Lusua, *Kupas Tuntas Algoritma Data Mining dan Implementasinya Menggunakan R*. Universitas Brawijaya Press, 2021.
- [18] Y. Resti, C. Irsan, M. T. Putri, I. Yani, A. Ansyori, and B. Suprihatin, "Identification of corn plant diseases and pests based on dig[1] Y. Resti, C. Irsan, M. T. Putri, I. Yani, A. Ansyori, and B. Suprihatin, "Identification of corn plant diseases and pests based on digital images using multinomial naïve bayes and k-nearest n," *Sci. Technol. Indones.*, vol. 7, no. 1, pp. 29–35, 2022.