

## Klasifikasi Penyakit Demam Berdarah Menggunakan Algoritma Stacking Ensemble Learning

I Kadek Yogi Prayoga<sup>1</sup>, I Made Gede Sunarya<sup>2</sup>, Putu Hendra Suputra<sup>3</sup>

<sup>1,2,3</sup>Fakultas Pascasarjana Ilmu Komputer, Universitas Pendidikan Ganesha  
Jalan Udayana No.11, Singaraja, Indonesia

e-mail: [yogi.prayoga@undiksha.ac.id](mailto:yogi.prayoga@undiksha.ac.id)<sup>1</sup>, [sunarya@undiksha.ac.id](mailto:sunarya@undiksha.ac.id)<sup>2</sup>, [hendra.suputra@undiksha.ac.id](mailto:hendra.suputra@undiksha.ac.id)<sup>3</sup>

Received : March, 2025

Accepted : August, 2025

Published : August, 2025

### Abstract

*Dengue Hemorrhagic Fever (DHF) is a contagious disease that remains a serious public health issue in Indonesia, with a consistent increase in cases each year. One of the main challenges is achieving accurate early diagnosis, as the initial symptoms often resemble other diseases such as influenza or chikungunya. This study aims to develop a DHF classification model using the stacking ensemble learning method to improve diagnostic accuracy compared to single classifier methods. The dataset consists of 650 patient medical records from Puskesmas I Mengwi District, including demographic data and clinical features such as rash, pain, body temperature, bleeding, and laboratory results. The research process involves data preprocessing, the implementation of three base learners (K-Nearest Neighbor, Naïve Bayes, and Decision Tree), and Logistic Regression as the meta-learner. Evaluation using k-fold cross-validation (5-fold and 10-fold) shows that the stacking ensemble achieves the highest accuracy of 84.15%, with a precision of 85.28%, recall of 92.62%, and F1-Score of 88.70%. These results demonstrate that stacking provides better and more stable performance than single classifiers. The proposed model has the potential to support early DHF diagnosis in healthcare facilities, helping medical personnel improve diagnostic accuracy and treatment effectiveness.*

**Keywords:** dengue hemorrhagic fever, classification, machine learning, stacking ensemble learning, confusion matrix

### Abstrak

*Demam Berdarah Dengue (DBD) merupakan penyakit menular yang menjadi masalah serius di Indonesia, dengan tren peningkatan kasus setiap tahun. Salah satu tantangan utama adalah diagnosis dini yang akurat, karena gejala awal DBD sering menyerupai penyakit lain seperti influenza atau chikungunya. Penelitian ini bertujuan membangun model klasifikasi DBD menggunakan metode stacking ensemble learning untuk meningkatkan akurasi diagnosis dibandingkan dengan metode single classifier. Dataset yang digunakan berasal dari 650 rekam medis pasien di Puskesmas I Kecamatan Mengwi, dengan atribut demografi dan gejala klinis seperti bintik merah, nyeri, suhu tubuh, pendarahan, serta hasil pemeriksaan laboratorium. Proses penelitian meliputi preprocessing data, penerapan tiga algoritma sebagai base learner (K-Nearest Neighbor, Naïve Bayes, dan Decision Tree), serta Logistic Regression sebagai meta learner. Evaluasi dengan k-fold cross validation (5-fold dan 10-fold) menunjukkan bahwa stacking ensemble menghasilkan akurasi tertinggi sebesar 84,15%, precision 85,28%, recall 92,62%, dan F1-Score 88,70%. Hasil ini membuktikan bahwa stacking lebih unggul dan stabil dibandingkan single classifier. Model ini berpotensi mendukung diagnosis dini DBD di fasilitas kesehatan, sehingga membantu tenaga medis meningkatkan ketepatan dan efektivitas penanganan pasien.*

**Kata Kunci:** demam berdarah dengue, klasifikasi, machine learning, stacking ensemble learning, confusion matrix

## 1. PENDAHULUAN

Demam Berdarah Dengue (DBD) merupakan salah satu penyakit menular yang menjadi masalah kesehatan global, khususnya di negara-negara tropis seperti Indonesia. Menurut laporan World Health Organization (WHO), lebih dari 390 juta kasus infeksi dengue terjadi setiap tahunnya di seluruh dunia, dengan sekitar 96 juta di antaranya menunjukkan manifestasi klinis. Studi global terkini mencatat bahwa pada tahun 2021 terdapat sekitar 58,96 juta kasus dengue secara global, dengan beban terbesar berada di Asia Tenggara dan Asia Selatan [1], [2]. Tren global menunjukkan bahwa insidensi dengue terus meningkat secara signifikan dalam tiga dekade terakhir, dengan proyeksi akan terus naik hingga tahun 2035 [2].

Di Indonesia, DBD menjadi salah satu masalah kesehatan utama yang menunjukkan tren peningkatan dalam beberapa tahun terakhir. Berdasarkan data Kementerian Kesehatan Republik Indonesia tahun 2024 hingga minggu ke-17, tercatat 88.593 kasus DBD dengan 621 kematian, yang menunjukkan adanya peningkatan signifikan dibandingkan periode sebelumnya [3]. Studi lain menyebutkan bahwa penyebaran DBD di Indonesia sangat dipengaruhi oleh faktor lingkungan, kepadatan penduduk, perubahan iklim, dan rendahnya efektivitas pengendalian vektor [4]. Di tingkat lokal, misalnya di wilayah Puskesmas 1 Kecamatan Mengwi, tercatat lonjakan kasus dari 300 pada tahun 2022 menjadi lebih dari 600 kasus pada tahun 2023. Kondisi ini tidak hanya menjadi beban bagi sistem layanan kesehatan, tetapi juga berdampak besar terhadap aspek sosial dan ekonomi masyarakat [5].

Salah satu tantangan utama dalam penanganan DBD adalah proses diagnosis yang cepat dan akurat, terutama pada tahap awal ketika gejala klinis masih bersifat umum dan menyerupai penyakit lain seperti influenza, chikungunya, atau COVID-19. Keterlambatan diagnosis dapat menyebabkan keterlambatan dalam penanganan medis yang berujung pada peningkatan risiko komplikasi hingga kematian [6], [7]. Keterbatasan metode diagnostik konvensional yang masih bergantung pada tes laboratorium dengan durasi waktu tertentu

menjadi hambatan tersendiri dalam upaya mitigasi dini.

Seiring dengan perkembangan teknologi informasi, metode berbasis *data mining* dan *machine learning* (ML) menjadi solusi potensial untuk mendukung proses diagnosis dini penyakit, termasuk DBD. Melalui pemanfaatan data rekam medis historis, algoritma klasifikasi mampu memetakan pola dan karakteristik pasien yang memiliki risiko lebih tinggi terinfeksi DBD [2], [8]. Beberapa algoritma klasifikasi yang umum digunakan meliputi Decision Tree, Naïve Bayes, K-Nearest Neighbor (KNN), dan Support Vector Machine (SVM) [9]. Namun, penelitian menunjukkan bahwa performa dari metode *single-classifier* seperti KNN dan Naïve Bayes memiliki keterbatasan, seperti sensitivitas terhadap *noise*, *overfitting*, ketidakseimbangan data, serta ketergantungan terhadap parameter tertentu [7], [9]-[12].

Beberapa studi menunjukkan bahwa algoritma KNN hanya mampu mencapai akurasi sebesar 72,31% pada nilai K=2, dan mengalami penurunan hingga 57,97% pada nilai K=3 [7], [11]. Sementara itu, Naïve Bayes mencatat akurasi yang bervariasi antara 64,02% hingga 67% dalam penerapannya pada diagnosis penyakit, termasuk DBD [12], [10]. Keterbatasan ini menyebabkan metode *single-classifier* kurang optimal untuk diterapkan dalam sistem pendukung keputusan di dunia nyata yang menuntut akurasi dan ketepatan yang tinggi.

Sebagai solusi untuk mengatasi kelemahan tersebut, metode *ensemble learning*, khususnya *stacking ensemble*, menjadi pendekatan yang lebih andal. Teknik *stacking* bekerja dengan menggabungkan beberapa *base learner* yang berbeda dan kemudian mengintegrasikan outputnya melalui *meta-learner*, sehingga dapat meningkatkan akurasi dan stabilitas model [13], [14]. Berbeda dengan teknik bagging yang 105 ocus pada pengurangan variansi atau *boosting* yang 105ocus pada pengurangan bias, *stacking* mampu mengoptimalkan kedua aspek tersebut secara bersamaan, sehingga lebih efektif ketika diterapkan pada data medis yang bersifat kompleks dan heterogen [15].

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan model klasifikasi penyakit DBD dengan menggunakan metode *stacking ensemble learning*. Model ini diharapkan dapat meningkatkan akurasi klasifikasi dibandingkan dengan metode *single-classifier*, serta memberikan kontribusi signifikan dalam pengembangan sistem pendukung keputusan yang dapat membantu tenaga medis dalam melakukan diagnosis dini dan penanganan DBD secara lebih efektif.

## 2. METODE PENELITIAN

### 2.1 Desain Penelitian

Penelitian ini merupakan penelitian terapan di bidang data mining dengan pendekatan *supervised learning* untuk klasifikasi penyakit Demam Berdarah Dengue (DBD). Penelitian bertujuan untuk membandingkan kinerja model klasifikasi antara metode *single classifier* dan *stacking ensemble learning* dalam memprediksi penyakit DBD berdasarkan atribut-atribut gejala klinis pasien.

### 2.2 Data Penelitian

Data yang digunakan dalam penelitian ini diperoleh dari rekam medis pasien di Puskesmas I Kecamatan Mengwi. Total data yang digunakan berjumlah 650 data pasien. Data tersebut memuat sejumlah atribut yang digunakan sebagai fitur dalam proses klasifikasi, yaitu jenis kelamin, umur, bitnik merah pada tubuh, nyeri, suhu tubuh, pendarahan pada hidung, pendarahan pada gusi, sistole, diastole, trombosit, leukosit dan hematokrit. Setiap atribut memiliki tipe data numerik atau kategorikal, tergantung pada karakteristik masing-masing variabel.

### 2.3 Preprocessing

Tahapan *preprocessing* dilakukan untuk memastikan data dalam kondisi layak untuk proses modeling. Tahapan preprocessing meliputi:

#### 2.3.1 Label Encoding

Seluruh atribut dalam dataset ini bersifat kategorikal, yang kemudian dikonversi ke dalam format numerik menggunakan metode *label encoding* [11]. Misalnya, atribut jenis kelamin dikonversi menjadi Laki-laki = 1 dan Perempuan = 0; atribut gejala seperti bintik merah pada tubuh, nyeri, pendarahan hidung, dan pendarahan gusi dikonversi dengan skema Ya =

1 dan Tidak = 0. Sementara itu, atribut umur dikelompokkan ke dalam lima kategori umur, dan atribut pemeriksaan seperti suhu tubuh, sistole, diastole, trombosit, leukosit, dan hematokrit dikonversi berdasarkan klasifikasi medis masing-masing. Proses ini bertujuan untuk memastikan semua data dapat diproses oleh algoritma pembelajaran mesin.

Tabel 1: Skema Label Encoding pada Dataset Penelitian

Atribut	Kategori	Label
Jenis Kelamin	Laki-laki / Perempuan	1 / 0
Bintik Merah, Nyeri, Pendarahan (Hidung/Gusi)	Ya /Tidak	1 /0
Umur	Dewasa Awal, Dewasa Akhir, Lansia Awal, Lansia Akhir, Manula	1, 2, 3, 4, 5
Suhu Tubuh	Normal / Tinggi	1 /2
Sistole / Diastole	Normal, Pra-hipertensi, hipertensi 1, hipertensi 2	
Trombosit, Leukosit, Hematokrit	Bukan, Belum Pasti, Pasti	1, 2, 3

#### 2.3.2 Data Selection

*Data selection* adalah langkah yang dilakukan untuk memilih data dari kumpulan data operasional [10]. Dalam hal ini, data yang dipilih berasal dari rekam medis pasien, meliputi parameter seperti jenis kelamin, umur, bitnik merah pada tubuh, nyeri, suhu tubuh, pendarahan pada hidung, pendarahan pada gusi, sistole, diastole, trombosit, leukosit, dan hematokrit. Data ini digunakan untuk membantu proses klasifikasi penyakit demam berdarah.

#### 2.3.3 Handling Missing Value

Data dengan nilai kosong pada atribut kunci dihapus dari dataset untuk menjaga kualitas dan konsistensi data, serta mencegah terjadinya bias atau ketidakakuratan selama proses pelatihan model [16].

### 2.3.4 Handling Imbalanced Data

Pemeriksaan terhadap distribusi kelas dilakukan untuk mengidentifikasi ketidakseimbangan data antara kelas positif (terdiagnosa DBD) dan kelas negatif. Jika ditemukan ketidakseimbangan signifikan, maka diterapkan teknik *Synthetic Minority Oversampling Technique* (SMOTE) untuk menyeimbangkan jumlah data pada masing-masing kelas [17], [18].

### 2.4 Pembagian Data

Data dibagi menggunakan metode *K-Fold Cross Validation* untuk memastikan generalisasi model yang baik. Penelitian ini menggunakan dua skenario pengujian, yaitu:

- 1) *5-Fold Cross Validation*
- 2) *10-Fold Cross Validation*

Setiap fold membagi data menjadi bagian training dan testing secara bergantian, sehingga hasil evaluasi menjadi lebih akurat dan tidak tergantung pada pembagian data tertentu.

### 2.5 Modeling

#### 2.5.1 Single Classifier

Model klasifikasi pertama dilakukan menggunakan tiga algoritma pembelajaran mesin yaitu:

- 1) Naïve Bayes

Algoritma Naïve Bayes adalah model klasifikasi berbasis probabilitas yang menghitung peluang dengan mengukur frekuensi dan kombinasi nilai dari data yang tersedia. Algoritma ini didasarkan pada Teorema Bayes dan mengasumsikan bahwa setiap variabel independen berkontribusi secara terpisah terhadap variabel kelas [19].

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \quad (1)$$

Dimana:

X = Data dengan klasifikasi tak tentu

H = Hipotesis yang menyatakan bahwa data dikategorikan dalam kelas tertentu

P(H|X) = Probabilitas hipotesis H jika diberikan kondisi X (probabilitas posterior)

P(H) = Probabilitas awal hipotesis H (probabilitas sebelumnya)

P(X|H) = Probabilitas X jika diberikan hipotesis H benar

P(X) = Probabilitas dari X

- 2) K-Nearest Neighbor

K-Nearest Neighbor (K-NN) termasuk dalam kategori pembelajaran berbasis instans. Teknik K-NN mengidentifikasi kelompok K objek dalam set data pelatihan yang paling dekat atau paling mirip dengan objek dalam set data baru atau uji. Persamaan dari algoritma KNN sebagai berikut [20], [21].

$$euc = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2)$$

Dimana:

pi = Data latih

qi = Data uji

i = Variable data

n = Dimensi data

- 3) Decision Tree

Pohon keputusan adalah ilustrasi langsung dari metode klasifikasi untuk sejumlah kelas yang terbatas. Simpul daun mewakili kelas yang berbeda, tepi diberi label dengan nilai atribut yang mungkin, dan simpul akar dan internal diberi label dengan nama atribut. Algoritma C4.5 digunakan untuk membangun pohon keputusan. Pohon keputusan ini merupakan salah satu pendekatan yang paling umum dan efisien untuk klasifikasi dan prediksi. Algoritma C4.5 membangun pohon keputusan dengan memanfaatkan data pelatihan yang terdiri dari berbagai kasus atau rekaman (tuple) dalam basis data. Persamaan dari metode Decision Tree ini ada dua, yaitu sebagai berikut [22], [23].

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (3)$$

Dimana:

S = Himpunan kasus

A = Atribut

n = Jumlah komponen atribut A

|S<sub>1</sub>| = Jumlah kasus dalam porsi ke-i

|S| = Jumlah total kasus dalam himpunan S

Sementara untuk persamaan kedua menggunakan persamaan sebagai berikut.

$$Entropy(S) = \sum_{i=1}^n (-p_i) * \log_2 p_i \quad (4)$$

Keterangan:

S = Kumpulan kasus

n = Jumlah partisi dari S

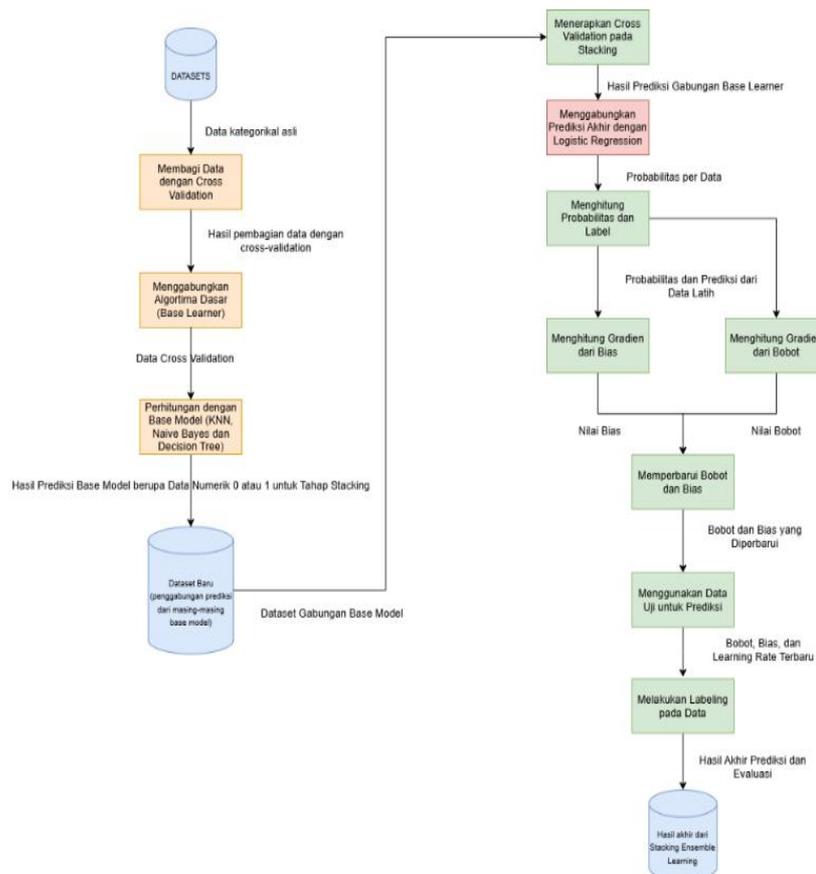
$\pi_i$  = Probabilitas yang diperoleh dari kasus  $i$  dibagi dengan total kasus

### 2.5.2 Stacking Ensemble Learning

*Stacking ensemble learning* adalah teknik penggabungan beberapa model dasar (*base learner*) dengan satu model akhir (*meta learner*) untuk meningkatkan akurasi prediksi. Metode ini bekerja dengan mengintegrasikan hasil prediksi dari beberapa *base learner* yang berbeda [24].

4) *Base Learner*  
*Base learner* yang digunakan yaitu K-Nearest Neighbor (KNN), Naïve Bayes, dan Decision Tree (C4.5).

5) *Meta Learner*  
 Logistic Regression, dipilih karena memiliki kemampuan generalisasi yang baik dan dapat menangani dataset dengan atribut numerik maupun kategorikal. Proses *stacking ensemble learning* ditunjukkan pada gambar 1.



Gambar 1. Proses Modeling dengan Metode Stacking Ensemble Learning

Proses dari model *stacking ensemble learning* dilakukan dengan beberapa langkah sebagai berikut:

- 1) Dataset dilatih pada masing-masing algoritma *base learner*.
- 2) Setiap *base learner* menghasilkan *output* prediksi (label atau probabilitas).
- 3) Hasil prediksi dari *base learner* digunakan untuk membentuk dataset baru.

4) Dataset baru tersebut digunakan sebagai input bagi *meta learner* Logistic Regression.

5) *Meta learner* mempelajari pola dari hasil-hasil prediksi *base learner* dan menghasilkan prediksi akhir yang lebih akurat.

### 2.6 Software dan Tools

Proses *modeling* dan evaluasi dilakukan menggunakan perangkat lunak berikut:

- 1) Python 3.x

- Scikit-Learn untuk implementasi algoritma *machine learning* (KNN, Naïve Bayes, Decision Tree, Logistic Regression, dan Stacking)
- Pandas dan NumPy untuk pengolahan data

### 2.7 Evaluasi Model

Evaluasi dilakukan dengan menggunakan confusion matrix untuk mengukur performa model. Metode evaluasi dilakukan pada masing-masing model *single classifier* (KNN, Decision Tree dan Naïve Bayes) dan *stacking ensemble learning* (Logistic Regression).

Tabel 1: Confusion Matrix

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

Evaluasi dilakukan untuk membandingkan performa antara metode *single classifier* dan *stacking ensemble learning*, sehingga dapat dianalisis sejauh mana peningkatan akurasi dan performa yang dihasilkan oleh metode *stacking*.

Tingkat akurasi dari pengklasifikasi pada data uji ditunjukkan oleh persentase *instance (tuple)* yang berhasil diklasifikasikan dengan benar. Rumus berikut menghitung *accuracy*, *precision* dan *recall* yang diperoleh dari *confusion matrix* [25].

- Akurasi (*Accuracy / AC*)asio prediksi yang benar terhadap total prediksi yang dibuat.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (5)$$

- Precision* pada proporsi kelas yang diprediksi positif secara tepat dibandingkan dengan semua kelas yang diprediksi sebagai positif.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

- Recall* merujuk pada proporsi kelas yang diprediksi positif secara tepat dibandingkan

dengan semua kelas yang sebenarnya positif.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

- F-measure* merupakan metrik kinerja yang menggabungkan nilai presisi dan recall menjadi satu ukuran

$$F1 - score = 2 \frac{precision \times recall}{precision + recall} \quad (8)$$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Perhitungan Manual dengan Stacking Ensemble Learning

Dataset yang dihasilkan merupakan kombinasi prediksi dari masing-masing *base learner*. Setelah terbentuk dataset baru ini, langkah berikutnya adalah mempersiapkan *data training* untuk *meta-learner* pada model *stacking ensemble*. Dataset ini akan digunakan untuk melatih *meta-learner* yang bertugas menggabungkan *output base learners* menjadi prediksi akhir yang lebih akurat.

Tabel 2: Data Latih Stacking Ensemble Learning

Data	X <sub>1</sub> (Naïve Bayes)	X <sub>2</sub> (KNN)	X <sub>3</sub> (Decision Tree)	Y (Target)
1	0	1	0	1
2	1	0	1	1
3	0	1	1	1
4	1	0	0	0
5	1	1	0	1
6	0	0	1	0
7	1	1	1	1
8	0	1	0	1

Setelah menentukan data latih tahap selanjutnya yaitu membuat inialisasi awal bobot dan nilai *bias*, dimana nilai nilainya pada tabel.

Tabel 3: Pembobotan Awal

Bobot (w)			Bias (b)	Learning rate (η)
w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>		
0	0	0	0	0.1

Pada tahapan selanjutnya lakukan perhitungan dari nilai  $\hat{Y}_i$  atau nilai prediksi dari *Logistic Regression* yang dihitung melalui kombinasi prediksi *base learner*. Dalam perhitungan ini

akan dilakukan perhitungan hingga iterasi kedua dan perbarui nilai bobot serta nilai bias setiap iterasinya, untuk mendapatkan hasil yang lebih akurat lakukan iterasi menggunakan kode program. Persamaan untuk mencari nilai  $\hat{Y}_i$  adalah sebagai berikut.

$$\hat{Y}_i = \frac{1}{1 + e^{-(w_1.X_{i1} + w_2.X_{i2} + w_3.X_{i3} + b)}} \quad (9)$$

### 1) Iterasi Pertama

Lakukan perhitungan untuk masing – masing data tersebut, contoh perhitungan adalah sebagai berikut.

$$\hat{Y}_1 = \frac{1}{1 + e^{-(0.0+0.1+0.0+0)}} = \frac{1}{1 + e^0} = 0.5$$

Untuk hasil dari perhitungan masing – masing data dapat dilihat pada tabel berikut.

Tabel 4: Hasil Iterasi Pertama

Data	$\hat{Y}_i$
X <sub>1</sub>	0.5
X <sub>2</sub>	0.5
X <sub>3</sub>	0.5
X <sub>4</sub>	0.5
X <sub>5</sub>	0.5
X <sub>6</sub>	0.5
X <sub>7</sub>	0.5
X <sub>8</sub>	0.5

Selanjutnya hitung nilai gradien untuk  $w_1$ ,  $w_2$ ,  $w_3$  dan nilai *bias*  $b$ , untuk persamaan yang digunakan sebagai berikut, untuk bobot dan *bias*  $b$ .

$$w_j \frac{\partial L}{\partial w_j} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i) \cdot X_{ij} \quad (10)$$

$$\frac{\partial L}{\partial b} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i) \quad (11)$$

Hitung gradien  $w_1$ :

$$\frac{\partial L}{\partial w_1} = \frac{1}{8} \begin{bmatrix} (0.5 - 1).0 + (0.5 - 1).1 + \\ (0.5 - 1).0 + (0.5 - 0).1 + \\ (0.5 - 1).1 + (0.5 - 0).0 + \\ (0.5 - 1).1 + (0.5 - 1).0 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_1} = \frac{1}{8} [0 - 0.5 + 0 + 0.5 - 0.5 +] = \frac{1}{8} (-1) = -0.125$$

Hitung gradien  $w_2$ :

$$\frac{\partial L}{\partial w_2} = \frac{1}{8} \begin{bmatrix} (0.5 - 1).1 + (0.5 - 1).0 + \\ (0.5 - 1).1 + (0.5 - 0).0 + \\ (0.5 - 1).1 + (0.5 - 0).0 + \\ (0.5 - 1).1 + (0.5 - 1).1 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_2} = \frac{1}{8} [-0.5 + 0 - 0.5 + 0 - 0.5 +] = \frac{1}{8} (-2.5) = -0.3125$$

Hitung gradien  $w_3$ :

$$\frac{\partial L}{\partial w_3} = \frac{1}{8} \begin{bmatrix} (0.5141 - 1).0 + (0.5125 - 1).1 + \\ (0.5172 - 1).1 + (0.5094 - 0).0 + \\ (0.5172 - 1).0 + (0.5094 - 0).1 + \\ (0.5203 - 1).1 + (0.5141 - 1).0 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_3} = \frac{1}{8} [0 - 0.4875 - 0.4828 +] = \frac{1}{8} (-0.9406) = -0.1175$$

Hitung gradien  $b$ :

$$\frac{\partial L}{\partial b} = \frac{1}{8} \begin{bmatrix} (0.5141 - 1) + (0.5125 - 1) + \\ (0.5172 - 1) + (0.5094 - 0) + \\ (0.5172 - 1) + (0.5094 - 0) + \\ (0.5203 - 1) + (0.5141 - 1) \end{bmatrix}$$

$$\frac{\partial L}{\partial b} = \frac{1}{8} \begin{bmatrix} (-0.4859) + (-0.4875) + \\ (-0.4828) + 0.5094 + \\ (-0.4828) + 0.5094 + \\ (-0.4797) + (-0.4859) \end{bmatrix} = \frac{1}{8} (-0.8858) = -0.1107$$

Setelah mendapatkan nilai gradien dari masing – masing parameter  $w_1$ ,  $w_2$ ,  $w_3$  dan *bias*, tahap selanjutnya perbarui nilai tersebut dengan menggunakan nilai *learning rate* 0.1. Untuk perhitungannya lengkapnya sebagai berikut:

Perbarui  $w_1$ :

$$w_1 = w_1 - \eta \cdot \frac{\partial L}{\partial w_1} = 0 - 0.1 \cdot (-0.125) = 0.0125$$

Perbarui  $w_2$ :

$$w_2 = w_2 - \eta \cdot \frac{\partial L}{\partial w_2} = 0 - 0.1 \cdot (-0.3125) = 0.03125$$

Perbarui  $w_3$ :

$$w_3 = w_3 - \eta \cdot \frac{\partial L}{\partial w_3} = 0 - 0.1 \cdot (-0.125) = 0.0125$$

Perbarui *bias*  $b$ :

$$b = b - \eta \cdot \frac{\partial L}{\partial b} = 0 - 0.1 \cdot (-0.25) = 0.025$$

### 2) Iterasi Kedua

Pada perhitungan iterasi 2 ini akan menggunakan nilai  $w_1$ ,  $w_2$ ,  $w_3$  dan *bias* yang sudah diperbarui pada iterasi 1. Setelah

mendapatkan nilai terbaru, dilanjutkan dengan melakukan perhitungan sama dengan yang dilakukan pada iterasi pertama namun mengganti parameternya.

$$\hat{Y}_i = \frac{1}{1 + e^{-(w_1.Xi1+w_2.Xi2+w_3.Xi3+b)}}$$

$$\hat{Y}_i = \frac{1}{1 + e^{-((0.0125 \times 0)+(0.03125 \times 1)+(0.0125 \times 0)+0.025)}}$$

$$\hat{Y}_i = \frac{1}{1 + e^{-(0.05625)}} = \frac{1}{1.9453} = 0.5140$$

Selanjutnya lakukan perhitungan pada semua data pada dataset tersebut, sehingga untuk hasilnya dapat dilihat pada tabel 5.

Tabel 5: Hasil Iterasi Kedua

Data	$\hat{Y}_i$
X <sub>1</sub>	0.5141
X <sub>2</sub>	0.5125
X <sub>3</sub>	0.5172
X <sub>4</sub>	0.5094
X <sub>5</sub>	0.5172
X <sub>6</sub>	0.5094
X <sub>7</sub>	0.5203
X <sub>8</sub>	0.5141

Selanjutnya lakukan perhitungan gradien untuk nilai  $w_1$ ,  $w_2$ ,  $w_3$  dan  $bias$ , untuk persamaannya sama pada iterasi pertama.

Hitung gradien  $w_1$ :

$$\frac{\partial L}{\partial w_1} = \frac{1}{8} \begin{bmatrix} (0.5141 - 1).0 + (0.5125 - 1).1 + \\ (0.5172 - 1).0 + (0.5094 - 0).1 + \\ (0.5172 - 1).1 + (0.5094 - 0).0 + \\ (0.5203 - 1).1 + (0.5141 - 1).0 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_1} = \frac{1}{8} \begin{bmatrix} 0 - 0.4875 + 0 + \\ 0.5094 - 0.428 + \\ 0 - 0.4797 + 0 \end{bmatrix} = \frac{1}{8}(-0.8858) = -0.7750$$

Hitung gradien  $w_2$ :

$$\frac{\partial L}{\partial w_2} = \frac{1}{8} \begin{bmatrix} (0.5141 - 1).1 + (0.5125 - 1).0 + \\ (0.5172 - 1).1 + (0.5094 - 0).0 + \\ (0.5172 - 1).1 + (0.5094 - 0).0 + \\ (0.5203 - 1).1 + (0.5141 - 1).1 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_2} = \frac{1}{8} \begin{bmatrix} -0.4859 + 0 - 0.4828 + \\ 0 - 0.4828 + \\ 0 - 0.4797 - 0.4859 \end{bmatrix} = \frac{1}{8}(-2.4171) = -0.3021$$

Hitung gradien  $w_3$ :

$$\frac{\partial L}{\partial w_3} = \frac{1}{8} \begin{bmatrix} (0.5141 - 1).0 + (0.5125 - 1).1 + \\ (0.5172 - 1).1 + (0.5094 - 0).0 + \\ (0.5172 - 1).0 + (0.5094 - 0).1 + \\ (0.5203 - 1).1 + (0.5141 - 1).0 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_3} = \frac{1}{8} \begin{bmatrix} 0 - 0.4875 - 0.4828 + \\ 0 + 0 + 0.5094 - \\ 0.4797 + 0 \end{bmatrix} = \frac{1}{8}(-0.9406) = -0.1175$$

Hitung  $bias$   $b$ :

$$\frac{\partial L}{\partial b} = \frac{1}{8} \begin{bmatrix} (0.5141 - 1) + (0.5125 - 1) + \\ (0.5172 - 1) + (0.5094 - 0) + \\ (0.5172 - 1) + (0.5094 - 0) + \\ (0.5203 - 1) + (0.5141 - 1) \end{bmatrix}$$

$$\frac{\partial L}{\partial b} = \frac{1}{8} \begin{bmatrix} (-0.4859) + (-0.4875) + \\ (-0.4828) + 0.5094 + \\ (-0.4828) + 0.5094 + \\ (-0.4797) + (-0.4859) \end{bmatrix} = \frac{1}{8}(-0.8858) = -0.1107$$

Setelah mendapatkan nilai gradien dari masing – masing parameter  $w_1$ ,  $w_2$ ,  $w_3$  dan  $bias$ , tahap selanjutnya perbarui nilai tersebut dengan menggunakan nilai  $learning$  rate 0.1. Untuk perhitungannya lengkapnya sebagai berikut:

Perbarui  $w_1$ :

$$w_1 = w_1 - \eta \cdot \frac{\partial L}{\partial w_1} = 0.0125 - 0.1 \cdot (-0.7750) = 0.0900$$

Perbarui  $w_2$ :

$$w_2 = w_2 - \eta \cdot \frac{\partial L}{\partial w_2} = 0.03125 - 0.1 \cdot (-0.3021) = 0.06146$$

Perbarui  $w_3$ :

$$w_3 = w_3 - \eta \cdot \frac{\partial L}{\partial w_3} = 0.0125 - 0.1 \cdot (-0.1175) = 0.02425$$

Perbarui  $bias$   $b$ :

$$b = b - \eta \cdot \frac{\partial L}{\partial b} = 0.025 - 0.1 \cdot (-0.1107) = 0.03607$$

Dari nilai – nilai tersebut akan digunakan dalam perhitungan iterasi selanjutnya. Untuk nilai lengkapnya dapat dilihat pada tabel 6.

Tabel 6: Pembobotan Iterasi Kedua

Bobot ( $w$ )			Bias ( $b$ )	Learn ing rate ( $\eta$ )
$w_1$	$w_2$	$w_3$		
0.0900	0.0614	0.024	0.0360	0.1
	6	25	7	

Selanjutnya setelah melakukan  $training$  pada dataset tersebut sehingga mendapatkan nilai bobot dan  $bias$ , dilanjutkan dengan uji coba pada data uji, untuk data uji yang digunakan dapat dilihat pada tabel 7.

Tabel 7: Data Uji

X <sub>1</sub> (Naïve Bayes)	X <sub>2</sub> (KNN)	X <sub>3</sub> (Decision Tree)	Y(Target)
0	1	0	1

Selanjutnya data uji tersebut dilakukan perhitungan dengan persamaan (5) pada *Logistic Regression* dengan menggunakan nilai terbaru yang sudah diperoleh dari iterasi kedua, sehingga perhitungannya dapat dilihat sebagai berikut.

$$\hat{Y}_i = \frac{1}{1 + e^{-((0.0900 x_0) + (0.06146 x_1) + (0.02425 x_0) + 0.03607)}}$$

$$\hat{Y}_i = \frac{1}{1 + e^{-(-0.09753)}} = \frac{1}{1 + 0.9071}$$

$$\hat{Y}_i = \frac{1}{1.9071} = 0.5243$$

Bandingkan probabilitas  $P(Y = 1)$  dengan ambang batas 0.5. Dikarenakan hasil dari perhitungan pada data uji lebih besar dari 0.5 maka data uji tersebut akan diberikan label 1 (positif).

### 3.2 Hasil Uji

Studi ini menggunakan *k-fold cross validation* untuk pelatihan dan pengujian. *Cross validation k-fold* adalah metode statistik yang digunakan untuk menilai performansi model atau algoritma yang dikembangkan. Selama fase pelatihan, kumpulan data dibagi menjadi data pelatihan dan data pengujian. Model dilatih dengan data pelatihan dan divalidasi dengan data pengujian menggunakan *cross validation k-fold* dengan 5 dan *k-fold* 10.

Tabel 8: Pengujian 450 Data dengan Cross Validation

Pengujian K-Fold	Metode	Accuracy	Precision	Recall	F1 Score
5	KNN	72.00%	78.50%	80.19%	78.95%
	Naïve Bayes	79.33%	80.25%	<b>91.04%</b>	85.15%
	Decision Tree	69.33%	77.82%	75.48%	76.26%
	Logistic Regression	76.22%	76.24%	76.22%	75.29%

10	Stacking Ensemble Learning	<b>81.55%</b>	<b>83.62%</b>	90.05%	<b>86.39%</b>
	KNN	70.88%	77.56%	79.50%	78.14%
	Naïve Bayes	80.22%	81.10%	<b>91.42%</b>	85.76%
	Decision Tree	69.11%	77.75%	74.53%	75.62%
	Logistic Regression	76.89%	76.89%	76.89%	75.84%
	Stacking Ensemble Learning	<b>80.88%</b>	<b>82.70%</b>	90.13%	<b>86.04%</b>

Tabel 9: Pengujian 650 Data dengan Cross Validation

Pengujian K-Fold	Metode	Accuracy	Precision	Recall	F1 Score
5	KNN	77.69%	81.21%	87.25%	84.03%
	Naïve Bayes	83.23%	82.85%	<b>94.69%</b>	88.35%
	Decision Tree	74.61%	81.85%	80.15%	80.98%
	Logistic Regression	76.00%	75.26%	76.00%	74.87%
	Stacking Ensemble Learning	<b>84.15%</b>	<b>85.11%</b>	92.63%	<b>88.70%</b>
10	KNN	78.15%	81.42%	87.78%	84.29%
	Naïve Bayes	82.92%	82.79%	<b>94.61%</b>	88.11%
	Decision Tree	73.07%	79.78%	80.64%	80.10%

Logistic Regression	77.3 8%	77.1 0%	77.3 8%	76.2 0%
Stacking Ensemble Learning	<b>84.1</b> <b>5%</b>	<b>85.2</b> <b>8%</b>	92.6 2%	<b>88.6</b> <b>2%</b>

Hasil pengujian yang ditampilkan pada Tabel 9 dan Tabel 10 menunjukkan bahwa metode *stacking ensemble learning* konsisten menghasilkan performa yang lebih baik dibandingkan dengan algoritma *single classifier* lainnya, baik dari sisi *accuracy*, *precision*, *recall*, maupun F1-Score.

Secara spesifik, metode Naïve Bayes memiliki nilai *recall* yang paling tinggi dibandingkan metode lain pada semua skenario pengujian. *Recall* yang tinggi pada Naïve Bayes menunjukkan bahwa model ini sangat sensitif terhadap deteksi kasus positif (DBD), sehingga potensi untuk mendeteksi pasien yang benar-benar terkena DBD menjadi lebih tinggi. Namun, hal ini sering kali diikuti dengan *precision* yang sedikit lebih rendah, karena Naïve Bayes cenderung mengorbankan ketepatan demi tidak melewatkan kasus positif.

Di sisi lain, *stacking ensemble learning* berhasil mencapai keseimbangan yang sangat baik antara *precision* dan *recall*, serta memiliki F1-Score yang lebih tinggi. Hal ini menunjukkan bahwa metode *stacking* mampu menggabungkan kekuatan dari beberapa base learner (KNN, Naïve Bayes, Decision Tree) dan mengoptimalkannya melalui *meta-learner* Logistic Regression. Dengan demikian, *stacking* tidak hanya baik dalam mendeteksi kasus positif, tetapi juga tetap menjaga ketepatan prediksi terhadap data.

Hasil *stacking* juga menunjukkan stabilitas yang lebih baik pada semua variasi jumlah data (450 dan 650) dan variasi k-fold (5 dan 10), yang mengindikasikan bahwa metode ini memiliki kemampuan generalisasi yang lebih baik.

### 3.3 Pembahasan

Hasil dari pengujian yang dilakukan memperlihatkan bahwa metode *stacking*

*ensemble learning* secara konsisten memberikan hasil yang lebih baik dibandingkan dengan *single classifier*. Ini sejalan dengan penelitian oleh, yang menyatakan bahwa metode *stacking* mampu meningkatkan performa klasifikasi pada permasalahan lain, seperti klasifikasi topik vaksin COVID-19.

Performa Naïve Bayes yang memiliki *recall* tinggi juga sejalan dengan temuan penelitian sebelumnya oleh, yang menunjukkan bahwa Naïve Bayes memiliki keunggulan dalam mendeteksi kelas positif, meskipun *precision*-nya relatif lebih rendah.

Dalam konteks klasifikasi penyakit DBD, *recall* yang tinggi adalah metrik yang sangat penting. Hal ini karena kesalahan dalam bentuk *false negative* (pasien yang sebenarnya terkena DBD namun diklasifikasikan negatif) berpotensi sangat berbahaya, mengingat DBD merupakan penyakit yang dapat berkembang cepat dan menyebabkan kematian jika tidak ditangani dengan segera. Oleh karena itu, model dengan *recall* tinggi membantu memastikan pasien dengan DBD dapat segera terdeteksi dan ditangani.

Namun, *precision* yang tinggi juga tetap penting untuk menghindari diagnosis yang salah pada pasien yang tidak menderita DBD. Inilah keunggulan dari metode *stacking*, yang mampu menjaga balance antara *recall* dan *precision*.

Dari sisi *novelty*, penelitian ini menawarkan kontribusi baru dengan menerapkan metode *stacking ensemble learning* pada klasifikasi penyakit DBD menggunakan data riil dari Puskesmas I Kecamatan Mengwi. Kombinasi algoritma KNN, Naïve Bayes, dan Decision Tree sebagai *base learner* serta Logistic Regression sebagai *meta-learner* belum banyak digunakan dalam penelitian serupa di bidang kesehatan, khususnya untuk klasifikasi penyakit DBD.

Kontribusi ilmiah dari penelitian ini tidak hanya terletak pada penerapan metode *ensemble* yang mampu meningkatkan akurasi, tetapi juga pada bagaimana metode ini dapat diaplikasikan dalam mendukung proses diagnosis penyakit DBD di fasilitas kesehatan tingkat pertama. Dengan sistem yang akurat dan andal, proses skrining dan diagnosis DBD dapat dilakukan lebih cepat dan tepat, sehingga membantu

menurunkan angka kesalahan diagnosis dan meningkatkan efektivitas penanganan pasien.

#### 4. KESIMPULAN

Penelitian ini membuktikan bahwa penerapan metode *stacking ensemble learning* dalam klasifikasi penyakit Demam Berdarah Dengue (DBD) mampu meningkatkan performa prediksi dibandingkan dengan metode *single classifier* seperti K-Nearest Neighbor, Naïve Bayes, dan Decision Tree.

Metode *stacking* berhasil memberikan hasil yang lebih stabil dan akurat karena menggabungkan keunggulan dari beberapa model dasar. Model ini tidak hanya mampu mendeteksi lebih banyak kasus positif DBD secara benar (*recall* tinggi), tetapi juga menjaga ketepatan dalam prediksi (*precision* tinggi), sehingga mampu meminimalkan kesalahan dalam diagnosis.

Peningkatan performa model menunjukkan bahwa pendekatan *stacking* lebih efektif dalam menangkap pola kompleks dari data gejala klinis DBD. Hal ini sangat penting dalam konteks medis, di mana akurasi diagnosis berkontribusi langsung terhadap ketepatan penanganan pasien.

Secara praktis, model yang dikembangkan dalam penelitian ini berpotensi digunakan sebagai alat bantu diagnosis awal DBD di fasilitas kesehatan tingkat pertama, seperti Puskesmas, untuk membantu tenaga medis dalam proses skrining dan pengambilan keputusan lebih cepat dan lebih tepat.

Untuk pengembangan ke depan, penelitian ini dapat diperluas dengan pengujian pada dataset yang lebih besar dan lebih bervariasi, serta dengan eksplorasi kombinasi algoritma lain dan teknik *balancing data* yang lebih canggih untuk meningkatkan performa model secara lebih baik.

#### DAFTAR PUSTAKA

[1] Y. Liang and X. Dai, "The global incidence and trends of three common flavivirus infections (Dengue, yellow fever, and Zika) from 2011 to 2021," *Front Microbiol*, vol. 15, Aug. 2024, doi: 10.3389/fmicb.2024.1458166.

[2] S.-X. Zhang *et al.*, "Global, regional, and national burden of dengue, 1990–2021: Findings from the global burden of disease study 2021," *Decoding Infection and Transmission*, vol. 2, p. 100021, 2024, doi: 10.1016/j.dcit.2024.100021.

[3] Rokom, "Waspada DBD di Musim Kemarau," Kementerian Kesehatan. Accessed: Dec. 26, 2024. [Online]. Available: <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20240616/0045767/waspada-dbd-di-musim-kemarau/>

[4] N. D. Rinarto *et al.*, "Community Empowerment in Prevention of Dengue Hemorrhagic Fever (DHF) in Coastal Areas," *Journal of Community Engagement in Health and Nursing*, vol. 1, no. 2, pp. 61–65, Jul. 2023, doi: 10.30643/jcehn.v1i2.305.

[5] N. Tian *et al.*, "Dengue Incidence Trends and Its Burden in Major Endemic Regions from 1990 to 2019," *Trop Med Infect Dis*, vol. 7, no. 8, p. 180, Aug. 2022, doi: 10.3390/tropicalmed7080180.

[6] M. Du, W. Jing, M. Liu, and J. Liu, "The Global Trends and Regional Differences in Incidence of Dengue Infection from 1990 to 2019: An Analysis from the Global Burden of Disease Study 2019," *Infect Dis Ther*, vol. 10, no. 3, pp. 1625–1643, Sep. 2021, doi: 10.1007/s40121-021-00470-2.

[7] M. Y. Matdoan, "Penerapan Metode K-Nearest Neighbor untuk Mengklasifikasi Penyebaran Kasus Demam Berdarah Dengue (DBD) di Kabupaten Maluku Tenggara," *Square: Journal of Mathematics and Mathematics Education*, vol. 4, no. 2, pp. 75–82, Oct. 2022, doi: 10.21580/square.2022.4.2.13056.

[8] S. Shirazi, H. Baziyad, and H. Karimi, "An Application-Based Review of Recent Advances of Data Mining in Healthcare," *Journal of Biostatistics and Epidemiology*, Aug. 2020, doi: 10.18502/jbe.v5i4.3864.

[9] J. Mantik, N. Kadek, W. Patrianingsih, K. Setemen, D. Gede, and H. Divayana, "Comparison Analysis Of Naïve Bayes And K-Nearest Neighbor Methods On The Prediction Of Academic Potential In Smk Ti Bali Global Badung," 2021.

- [10] I. G. A. P. Mahendra, I. M. A. Wirawan, and I. G. A. Gunadi, "Enhancement performance of the Naïve Bayes method using AdaBoost for classification of diabetes mellitus dataset type II," *International Journal of Advances in Applied Sciences*, vol. 13, no. 3, pp. 733–742, Sep. 2024, doi: 10.11591/ijaas.v13.i3.pp733-742.
- [11] G. A. J. Satvika, I. N. Sukajaya, and I. G. A. Gunadi, "Improving k-nearest neighbor performance using permutation feature importance to predict student success in study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 3, pp. 1835–1844, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1835-1844.
- [12] Y. Handayani and A. R. Hakim, "Classification of Naive Bayes Algorithm on Dengue Hemorrhagic Fever and Typhoid Fever Based on Hematology Results," *Journal of Applied Intelligent System*, vol. 8, no. 1, pp. 94–99, Feb. 2023, doi: 10.33633/jais.v8i1.7547.
- [13] A. K. Putri and H. Suparwito, "Uji Algoritma Stacking Ensemble Classifier pada Kemampuan Adaptasi Mahasiswa Baru dalam Pembelajaran Online," 2023.
- [14] R. Jayapermana, A. Aradea, and N. I. Kurniati, "Implementation of Stacking Ensemble Classifier for Multi-class Classification of COVID-19 Vaccines Topics on Twitter," *Scientific Journal of Informatics*, vol. 9, no. 1, pp. 8–15, May 2022, doi: 10.15294/sji.v9i1.31648.
- [15] E. Togami *et al.*, "Epidemiology of dengue reported in the World Health Organization Western Pacific Region, 2013–2019," *Western Pacific Surveillance and Response Journal*, vol. 14, no. 1, pp. 12–27, Jan. 2023, doi: 10.5365/wpsar.2023.14.1.973.
- [16] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *J Big Data*, vol. 8, no. 1, p. 140, Oct. 2021, doi: 10.1186/s40537-021-00516-9.
- [17] R. Sena and S. Ben Hamida, "ACTIVE SMOTE for Imbalanced Medical Data Classification," 2024, pp. 81–97. doi: 10.1007/978-3-031-51664-1\_6.
- [18] N. A. Azhar, M. S. Mohd Pozi, A. M. Din, and A. Jatowt, "An Investigation of SMOTE Based Methods for Imbalanced Datasets with Data Complexity Analysis (Extended Abstract)," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, IEEE, May 2024, pp. 5735–5736. doi: 10.1109/ICDE60146.2024.00499.
- [19] K. S. Y. Pande, D. G. H. Divayana, and G. Indrawan, "Comparative analysis of naïve bayes and knn on prediction of forex price movements for gbp/usd currency at time frame daily," *J Phys Conf Ser*, vol. 1810, no. 1, p. 012012, Mar. 2021, doi: 10.1088/1742-6596/1810/1/012012.
- [20] S. H. A. Samsudin, N. M. Sabri, N. Isa, and U. F. M. Bahrin, "Sentiment Analysis on Acceptance of New Normal in COVID-19 Pandemic using Naïve Bayes Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 9, 2022, doi: 10.14569/IJACSA.2022.0130968.
- [21] P. P. Vedanty, M. W. A. Kesiman, I. M. G. Sunarya, and I. G. A. A. D. Indradewi, "Identification of Leaf Diseases of Medicinal Plants Using K-Nearest Neighbor Based on Color, Texture, and Shape Features," in *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/ICAICTA59291.2023.10390034.
- [22] A. C. Dewi and G. Fajar Shidik, "Early Detection for Determinants of Risky Behavior in Cervical Cancer Cases through the C4.5 Algorithm in Indonesia," in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2022, pp. 198–203. doi: 10.1109/iSemantic55962.2022.9920459.
- [23] T. A. Pakasi, L. A. Utami, A. Surniandari, H. Rachmi, and D. Nurlela, "Comparison of C4.5 and Naïve Bayes algorithm to determine recommendations of patients receiving the Covid-19 vaccine at Cimanggis Jaya clinic," 2023, p. 030009. doi: 10.1063/5.0128770.

- [24] O. Asmae, S. Saleh, R. Abdelhadi, and B. Bachir, "Enhancing Parkinson's Disease Diagnosis: A Stacking Ensemble Approach Leveraging Machine Learning Techniques," in *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, IEEE, May 2024, pp. 1–7. doi: 10.1109/IRASET60544.2024.10549375.
- [25] I. M. G. Sunarya, I Wayan Treman, and Putu Zasya Eka Satya Nugraha, "Classification of Rice Growth Stage on UAV Image Based on Convolutional Neural Network Method," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 12, no. 1, pp. 146–155, May 2023, doi: 10.23887/janapati.v12i1.60959.