

# Evaluasi Kinerja Model *Machine Learning* dalam Klasifikasi Penyakit THT: Studi Komparatif *Naïve Bayes*, *SVM*, dan *Random Forest*

Nur Wachid Adi Prasetya<sup>1\*</sup>, Linda Perdana Wanti<sup>2</sup>, Riyadi Purwanto<sup>3</sup>,  
Isa Bahroni<sup>4</sup>, Rostika Listyaningrum<sup>5</sup>

<sup>1,4</sup>Program Studi Sarjana Terapan Teknologi Rekayasa Multimedia, Politeknik Negeri Cilacap

<sup>2</sup>Program Studi Sarjana Terapan Rekayasa Keamanan Siber, Politeknik Negeri Cilacap

<sup>3,5</sup>Program Studi Teknik Informatika, Politeknik Negeri Cilacap

<sup>1,2,3,4,5</sup>Jln. Dr. Soetomo No.1 Karangcengis Sidakaya, Kabupaten Cilacap, 53212, Indonesia

E-mail: [nwap.pnc@pnc.ac.id](mailto:nwap.pnc@pnc.ac.id)<sup>1</sup>, [linda\\_perdana@pnc.ac.id](mailto:linda_perdana@pnc.ac.id)<sup>2</sup>, [riyadi\\_purwanto@pnc.ac.id](mailto:riyadi_purwanto@pnc.ac.id)<sup>3</sup>,  
[isabahroni@pnc.ac.id](mailto:isabahroni@pnc.ac.id)<sup>4</sup>, [li\\_sa007@pnc.ac.id](mailto:li_sa007@pnc.ac.id)<sup>5</sup>

---

## Abstrak

### Info Naskah:

Naskah masuk: 28 Mei 2025

Direvisi: 26 Juni 2025

Diterima: 2 Juli 2025

Klasifikasi penyakit Telinga, Hidung, dan Tenggorokan (THT) penting dilakukan untuk membantu diagnosis lebih cepat dan akurat. Namun, hingga kini belum ada studi yang secara khusus membandingkan performa algoritma *Naïve Bayes*, *Support Vector Machine* (SVM), dan *Random Forest* dalam kasus penyakit THT. Penelitian ini bertujuan mengevaluasi dan membandingkan ketiga model klasifikasi tersebut dalam mengenali penyakit THT dengan atau tanpa komorbiditas. Data diperoleh dari rekam medis rumah sakit, diolah melalui pra-pemrosesan, seleksi fitur menggunakan ANOVA, serta penyeimbangan data dengan *SMOTE*. Hasil menunjukkan bahwa SVM memiliki kinerja terbaik (akurasi 59%), diikuti *Random Forest* (57%), dan *Naïve Bayes* (48%). SVM unggul karena konsistensi tinggi pada semua metrik evaluasi. Penelitian ini menyimpulkan bahwa pemilihan model klasifikasi sangat mempengaruhi akurasi diagnosis penyakit THT.

---

## Abstract

### Keywords:

ent classification;

comorbidities;

naïve bayes;

svm;

random forest.

Classification of Ear, Nose, and Throat (ENT) diseases is essential to support faster and more accurate diagnosis. However, no prior studies have specifically compared the performance of *Naïve Bayes*, *Support Vector Machine* (SVM), and *Random Forest* algorithms in ENT cases. This study aims to evaluate and compare the three classification models in identifying ENT diseases with or without comorbidities. Medical record data were processed through preprocessing, feature selection using ANOVA, and class balancing with *SMOTE*. The results showed that SVM outperformed the other models with the highest accuracy (59%), followed by *Random Forest* (57%), and *Naïve Bayes* (48%). SVM demonstrated superior performance due to its consistent scores across all evaluation metrics. The study concludes that the choice of classification model significantly impacts the accuracy of ENT disease diagnosis.

---

\*Penulis korespondensi:

Nur Wachid Adi Prasetya

E-mail: [nwap.pnc@pnc.ac.id](mailto:nwap.pnc@pnc.ac.id)

## 1. Pendahuluan

Masalah kesehatan yang berkaitan dengan telinga, hidung, dan tenggorokan (THT) cukup sering dialami banyak orang dan dapat berdampak nyata pada aktivitas sehari-hari. Sebagai contoh, infeksi telinga dapat menyebabkan rasa sakit hingga gangguan pendengaran. Kemudian ada penyakit rhinitis, yang merupakan radang di bagian hidung, biasa muncul dengan gejala bersin terus-menerus dan hidung tersumbat. Ada juga penyakit sinusitis yang membuat area sekitar wajah terasa nyeri, serta radang amandel atau tonsilitis yang membuat sulit menelan dan terasa tidak nyaman saat makan atau minum [1]. Gangguan pada bagian THT dapat meningkatkan kerentanan seseorang terhadap penyakit lain, serta memperbesar risiko infeksi atau masalah medis lainnya. Sebagai contoh, pasien asma sering mengalami rhinitis atau sinusitis karena gangguan tersebut berkaitan dengan peradangan saluran napas. Di sisi lain, kondisi seperti asam lambung yang naik ke tenggorokan dapat memicu iritasi, suara serak, dan memperparah batuk kronis atau radang tenggorokan [2].

Mengelompokkan data dari hasil diagnosa pasien dapat membantu dalam mengidentifikasi kecenderungan dan pola yang ada, mempercepat proses pengambilan keputusan, serta berkontribusi terhadap peningkatan efisiensi operasional dan kualitas pelayanan [3]. Pada era transformasi digital, keputusan yang berkaitan dengan program kesehatan perlu didukung oleh sistem cerdas yang mampu mengolah data dan memberikan prediksi. Dengan bantuan teknologi *Machine Learning*, proses pengambilan keputusan dapat menjadi lebih cepat dan akurat karena sistem dapat menyajikan informasi prediktif secara otomatis tanpa campur tangan manual yang kompleks [4]. Menurut Abellera & Bulusu (2018), kajian *Machine Learning* dalam bidang kecerdasan buatan berfokus pada proses pembelajaran komputasional dan pengenalan pola, menggunakan algoritma seperti *supervised* dan *unsupervised learning* untuk menghasilkan prediksi serta membantu pengambilan keputusan otomatis dari data yang tersedia [4]. Model prediksi kasus penyakit dapat dibangun melalui penerapan *Machine Learning*, yang dalam prosesnya menggunakan berbagai jenis algoritma klasifikasi. Pemanfaatan algoritma klasifikasi disesuaikan dengan kebutuhan untuk memperkirakan level kejadian yang telah disesuaikan ke dalam kategori tertentu [4]. Penerapan algoritma *Machine Learning* dalam klasifikasi penyakit THT memungkinkan sistem mengenali pola dari data klinis, sehingga proses diagnosis dapat dilakukan dengan lebih cepat dan tepat. Dengan sistem klasifikasi yang baik, tenaga medis dapat terbantu dalam mengambil keputusan klinis dan merancang perawatan yang sesuai dengan kondisi pasien [5].

Pada penelitian ini, algoritma klasifikasi digunakan untuk mengklasifikasikan penyakit THT dengan atau tanpa diiringi penyakit lainnya (komorbiditas), menggunakan beberapa metode. Algoritma yang dipakai adalah *Naïve Bayes*, SVM, dan *Random Forest*. Penggunaan beberapa algoritma ini bertujuan untuk membandingkan berbagai algoritma klasifikasi, guna menentukan algoritma yang paling cocok untuk kasus penyakit THT. Penentuan algoritma yang paling akurat dan efisien, membuat kualitas diagnosis dan penanganan pasien dapat ditingkatkan.

Pada beberapa tahun belakangan, studi perbandingan algoritma klasifikasi pada *Machine Learning* di bidang kesehatan telah dilakukan. Lestari dan Homaidi (2024) melakukan perbandingan antara algoritma *Naïve Bayes* dan *Random Forest* dalam pengelompokan kanker payudara, dengan hasil bahwa *Random Forest* menunjukkan ketepatan yang lebih tinggi, yakni 85% [6]. Prasetyo et al. (2024) turut memakai *Random Forest* dan *Support Vector Machine* (SVM) dalam klasifikasi penyakit menular, dan *Random Forest* menunjukkan akurasi sebesar 90%, lebih baik dibandingkan SVM yang hanya mencapai 82% [5]. Depari et al. (2022) membandingkan model *Decision Tree*, *Naïve Bayes*, dan *Random Forest* untuk memprediksi penyakit jantung, dengan hasil bahwa *Random Forest* kembali unggul dengan akurasi 88%, sedangkan *Naïve Bayes* hanya memperoleh 75% [7]. Zuhri et al. (2025) melaksanakan studi perbandingan algoritma untuk mendeteksi diabetes, di mana *Random Forest* meraih akurasi 87%, lebih tinggi dibandingkan *Naïve Bayes* yang hanya mendapatkan 80% [8]. Septhya et al. (2023) mengaplikasikan *Decision Tree* dan SVM untuk klasifikasi kanker paru-paru, dengan SVM mencapai akurasi 85% [9]. Akmal et al. (2023) membandingkan algoritma *Naïve Bayes* dan K-Nearest Neighbors (KNN) dalam klasifikasi penyakit stroke, dengan hasil bahwa KNN menunjukkan akurasi 83%, sementara *Naïve Bayes* hanya 78% [10]. Napiyah et al. (2022) membandingkan metode *Naïve Bayes*, KNN, SVM yang berfokus pada ISPA (Infeksi Saluran Pernapasan Akut), yang secara klinis lebih spesifik ke bagian paru-paru dan saluran pernapasan bawah, dengan hasil SVM memberikan akurasi 99%, dibandingkan KNN (94%) dan *Naïve Bayes* (98%) [11]. Jamil et al. (2024) juga membandingkan metode *Naïve Bayes*, KNN, ANN, SVM, *Random Forest* untuk klasifikasi penyakit THT, dengan hasil KNN, ANN, *Random Forest*, dan SVM sebesar 100%, dibanding *Naïve Bayes* sebesar 98% [12].

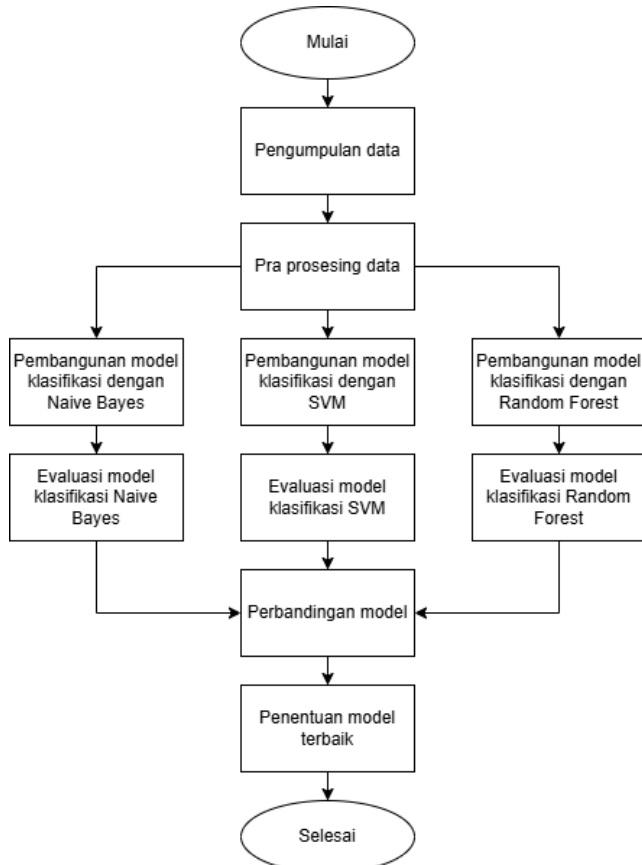
Walaupun sudah banyak penelitian yang melibatkan algoritma seperti *Naïve Bayes*, SVM, dan *Random Forest* dalam klasifikasi penyakit, masih sedikit penelitian yang secara spesifik mengevaluasi efektivitas algoritma-algoritma tersebut untuk kasus penyakit THT (Telinga, Hidung, dan Tenggorokan). Hal ini menunjukkan bahwa topik ini masih terbuka luas untuk diteliti lebih lanjut, terutama karena data pada kasus THT cenderung memiliki kompleksitas dan keragaman yang tinggi. Terlebih, belum ada penelitian yang membandingkan algoritma-algoritma tersebut pada kasus THT dengan komorbiditas, karena keadaan komorbiditas pada THT dipengaruhi oleh banyak faktor.

Pemilihan algoritma *Naïve Bayes*, SVM, dan *Random Forest* untuk komparasi klasifikasi penyakit THT karena berdasarkan penelitian terdahulu, ketiganya telah terbukti unggul dalam domain medis dengan kelebihan masing-masing, yaitu *Naïve Bayes* cepat dan efisien untuk dataset kecil, SVM akurat pada pola kompleks dan data linier, serta *Random Forest* stabil dan unggul dalam akurasi pada dataset besar, sehingga algoritma-algoritma tersebut merepresentasikan keragaman pendekatan klasifikasi yang kuat dan saling melengkapi. Penelitian ini diharapkan dapat memberikan pemahaman mengenai perbandingan algoritma

Naïve Bayes, SVM, dan Random Forest saat diterapkan pada klasifikasi penyakit THT.

**2. Metode**

Alur pelaksanaan penelitian ini adalah seperti pada Gambar 1.



Gambar 1. Alur pelaksanaan penelitian

**2.1 Pengumpulan Data**

Data yang digunakan pada penelitian ini diperoleh dari data rekam medis pasien penyakit THT dari Rumah Sakit Umum Daerah (RSUD) dr. R. Goeteng Taroenadibrata Kabupaten Purbalingga. Berdasarkan data yang diperoleh, diketahui atribut data berupa gejala penyakit adalah seperti pada Tabel 1.

Tabel 1. Data atribut gejala penyakit THT

No	Gejala penyakit THT	
1	G01-THT	telinga berdengung
2	G02-THT	pusing
3	G03-THT	nyeri kepala
4	G04-THT	pendengaran turun
5	G05-THT	nyeri leher
6	G06-THT	hidung buntu
7	G07-THT	bersin
8	G08-THT	nyeri telan
9	G09-THT	tenggorokan panas sakit
10	G10-THT	tenggorokan kering
11	G11-THT	penciuman terganggu
12	G12-THT	iritasi hidung
13	G13-THT	pilek menahun
14	G14-THT	badan panas
15	G15-THT	sulit buka mulut

No	Gejala penyakit THT	
16	G16-THT	ingus darah
17	G17-THT	sakit kepala
18	G18-THT	leher kaku
19	G19-THT	leher bengkak
20	G20-THT	sesak nafas
21	G21-THT	telinga penuh cairan
22	G22-THT	pengelompokan sistol
23	G23-THT	pengelompokan diastol
24	G24-THT	pengelompokan temperature
25	G25-THT	pengelompokan pulse
26	G26-THT	pengelompokan breath
27	G27-THT	pengelompokan height weight

Adapun label target dari data tersebut adalah klasifikasi jenis penyakit THT, yaitu penyakit THT tanpa penyakit penyerta (tanpa kormobiditas), dan penyakit THT dengan penyakit penyerta (dengan kormobiditas), pada Tabel 2.

Tabel 2. Data kelompok penyakit THT sebagai label target

No	Diagnosa penyakit THT	
1	P-THT	THT dengan atau tanpa kormodibitas

**2.2 Pra-Prosesing Data**

Pada penelitian ini, tahap *praprocessing* data yang dilakukan antara lain:

1) Transformasi Data

Proses transformasi data bertujuan untuk menyesuaikan skala pengukuran dari data mentah ke bentuk lain, agar sesuai dengan asumsi yang menjadi dasar analisis varians serta sejalan dengan cara kerja algoritma [13]. Pada penelitian ini, metode transformasi data yang dilakukan adalah dengan metode normalisasi Min-Max. Metode ini mentransformasikan data dari satu rentang nilai ke rentang lain, sambil tetap mempertahankan keterkaitan dengan data aslinya. Umumnya, teknik ini diterapkan untuk menyesuaikan data ke dalam skala antara 0 hingga 1 [14].

2) Distribusi Data

Salah satu tantangan terbesar dalam penerapan algoritma klasifikasi dan *Machine Learning* adalah menangani data dengan distribusi kelas yang tidak merata. Kondisi ini disebut sebagai ketimpangan data, di mana jumlah data dari kelas mayoritas jauh lebih banyak dibandingkan kelas lainnya. Ketidakseimbangan ini sering kali membuat sistem klasifikasi cenderung memihak ke kelas yang lebih besar, sehingga meningkatkan risiko terjadinya kesalahan dalam mengelompokkan data [15].

Masalah jumlah kelas data yang tidak seimbang dapat diatasi dengan cara menyesuaikan distribusi kelas, misalnya dengan memperbanyak data dari kelas minoritas (*oversampling*) atau mengurangi data dari kelas mayoritas (*undersampling*) [15].

3) Seleksi Fitur

Supaya data lebih mudah diproses, fitur-fitur yang tidak terlalu relevan atau tidak memberikan pengaruh besar terhadap isi dokumen biasanya disingkirkan. Dengan menyaring fitur seperti ini, sistem klasifikasi dapat bekerja lebih cepat dan memberikan hasil yang lebih akurat [16].

Penelitian ini menggunakan metode ANOVA untuk menemukan fitur-fitur yang paling berperan dalam

mempengaruhi variabel target dalam proses klasifikasi. Setiap fitur dinilai menggunakan *F-Score*, yang membantu melihat sejauh mana perbedaan antar fitur dibandingkan dengan variasi di dalam satu kelompok. Kemudian, fitur-fitur dengan nilai *F* tertinggi diseleksi melalui teknik *SelectKBest*, agar hanya fitur-fitur yang benar-benar relevan yang digunakan dalam membangun model klasifikasi [17].

### 2.3 Pembangunan Model Klasifikasi

Penelitian ini menerapkan algoritma *Naïve Bayes*, *Support Vector Machine* (SVM), dan *Random Forest* untuk membangun model klasifikasi penyakit THT.

#### 1) *Naïve Bayes*

*Naïve Bayes* adalah algoritma klasifikasi yang digunakan untuk memprediksi kemungkinan suatu peristiwa di masa depan berdasarkan pola yang ditemukan dari data sebelumnya. Algoritma ini bekerja dengan pendekatan probabilitas melalui Teorema Bayes, dan mengasumsikan bahwa setiap fitur dalam data tidak saling memengaruhi, artinya setiap atribut dianggap berdiri sendiri terhadap nilai dari kelas yang ingin diprediksi [18].

Proses klasifikasi menggunakan metode *Naïve Bayes* terdiri dari beberapa langkah utama, antara lain [19]:

- a. Mengestimasi Kemungkinan pada Tiap Kelas  
Probabilitas untuk setiap kategori dihitung berdasarkan data yang sedang dianalisis. Dalam praktik algoritma *Naïve Bayes*, perhitungan ini mencakup kombinasi antara kemungkinan awal (prior) dari suatu kategori dan peluang nilai fitur berdasarkan kelas yang tersedia.
- b. Menggunakan Teorema Bayes  
Teorema Bayes diaplikasikan untuk menggabungkan nilai probabilitas awal tiap kelas dengan hasil perhitungan sebelumnya guna memperoleh probabilitas gabungan untuk setiap prediksi.
- c. Memilih Kelas dengan Kemungkinan Tertinggi  
Setelah semua probabilitas dihitung, sistem akan menentukan kategori dengan nilai probabilitas terbesar sebagai hasil prediksi. Evaluasi performa metode *Naïve Bayes* biasanya dilakukan melalui indikator seperti akurasi, ketepatan (precision), dan sensitivitas (recall) guna mengetahui seberapa efektif model dalam melakukan pengklasifikasian.
- d. Formula Algoritma *Naïve Bayes*  
Rumus perhitungan *Naïve Bayes* juga digunakan dalam proses ini sebagai dasar matematis dari pendekatan tersebut.

$$P(D) = \frac{P(h) * P(h)}{P(D)} \quad (1)$$

Penjelasannya:

*h*: dugaan bahwa suatu data termasuk dalam kelompok atau kelas tertentu.

*D*: data yang belum diketahui termasuk dalam kelas mana.

*P(h)*: kemungkinan awal dari dugaan tersebut sebelum melihat data yang ada.

*P(D)*: seberapa besar peluang kemunculan data tersebut secara umum.

*P(D/h)*: peluang data muncul jika asumsi atau dugaan *h* benar.

*P(h/D)*: seberapa besar kemungkinan dugaan *h* benar jika kita sudah mengetahui data *D*

#### 2) *Support Vector Machine* (SVM)

SVM adalah salah satu algoritma klasifikasi yang digunakan untuk memisahkan data, baik yang bersifat linier maupun tidak linier. Tujuan utama dari metode ini adalah menemukan garis pemisah terbaik (*hyperplane*) agar data bisa terbagi dengan jelas ke dalam dua kategori [20].

Dalam penerapan algoritma SVM, model linier digunakan dengan bantuan rumus khusus untuk menemukan garis pemisah terbaik yang mampu memisahkan data secara optimal, dengan rumus sebagai berikut [21]:

$$Y = w^T x_i + b, \text{ di mana } i = 1, 2, \dots, l \quad (2)$$

Penjelasannya:

*x<sub>i</sub>*: vektor yang berbentuk baris dengan jumlah dimensi sebanyak *k*, di mana *k* merupakan banyaknya fitur atau atribut.

*Y*: target atau label dari data, yang hanya memiliki dua kemungkinan nilai, yaitu -1 dan +1.

*l*: total keseluruhan data yang digunakan.

*w*: parameter bobot yang berfungsi mengatur pengaruh masing-masing fitur dalam model.

*b*: nilai bias, atau kesalahan sistematis dalam prediksi.

Guna mengetahui margin maksimum dalam algoritma SVM, digunakan konsep support vector atau vektor penyangga. Margin ini dihitung melalui persamaan berikut [22]:

$$(w \cdot x_i) + b = 0 \quad (3)$$

Adapun jika sebuah data (*x<sub>i</sub>*) masuk ke dalam kelas negatif atau -1, maka kondisi tersebut dirumuskan dengan [21]:

$$(w \cdot x_i + b) \leq 1, \text{ di mana } Y_i = -1 \quad (4)$$

Sebaliknya, untuk data yang tergolong ke dalam kelas positif atau +1, digunakan perhitungan lain sebagaimana dijelaskan dalam persamaan berikut [21]:

$$(w \cdot x_i + b) \geq 1, \text{ di mana } Y_i = 1 \quad (5)$$

#### 3) *Random Forest*

*Random Forest* adalah kumpulan dari pohon keputusan yang dibentuk menggunakan contoh data yang dipilih secara acak, namun menerapkan aturan pemisahan node yang bervariasi [7][23]. Tujuan dari metode ini adalah membangun pohon keputusan dengan tiga jenis simpul utama yaitu simpul akar (*root node*), simpul percabangan (*internal node*), dan simpul daun (*leaf node*). Prosesnya dilakukan dengan memilih data dan atribut secara acak namun tetap mengikuti aturan tertentu. Simpul akar berada di bagian paling atas dan menjadi titik awal dari seluruh proses pengambilan keputusan. Selanjutnya ada simpul percabangan, yaitu titik di mana pohon bercabang, dengan satu jalur masuk dan minimal dua arah keluar. Terakhir, ada simpul daun atau simpul akhir yang menandai hasil dari

proses klasifikasi. Simpul ini tidak memiliki cabang lanjutan, hanya menerima satu masukan [20].

Dalam membangun pohon keputusan, langkah pertama yang dilakukan biasanya adalah menghitung nilai entropi (*entropy*) dan information gain. Entropi digunakan untuk melihat seberapa berantakan data pada suatu atribut, sedangkan information gain mengukur seberapa banyak informasi baru yang didapat dari pemisahan tersebut. Selain itu, indeks gini juga digunakan untuk menentukan node mana yang paling optimal sebagai pembagi. Proses perhitungannya akan terus berlangsung selama nilai gini belum mencapai nol, dan akan berhenti secara otomatis ketika hasil akhirnya menunjukkan tidak ada lagi ketimpangan data [24].

Untuk menghitung nilai entropi, digunakan rumus berikut [24]:

$$Entropi(S) = \sum_{i=1}^n -p_i p_i \quad (6)$$

Penjelasannya:

$S$  : kumpulan data yang sedang dianalisis

$n$  : berapa banyak kategori atau kelas yang terdapat dalam data tersebut

$p_i$  : seberapa besar kemungkinan setiap kelas ke- $i$  muncul dalam keluaran dari himpunan data  $S$

Sedangkan rumus menentukan information gain adalah sebagai berikut [24]:

$$Information\ gain(A) = Entropi(S) - \sum_{i=1}^n \frac{S_i}{S} \times Entropi(S_i) \quad (7)$$

Penjelasannya:

$A$  : karakteristik atau fitur dari data yang sedang dianalisis

$S$  : Entropi

$|S_i|$  : berapa banyak data yang termasuk ke dalam kategori atau nilai ke- $i$  dari atribut tertentu

$|S|$  : jumlah total keseluruhan data yang terdapat dalam kumpulan data tersebut

### 2.4 Evaluasi Model

Pada tahap evaluasi model ini, data yang dikumpulkan, dibagi menjadi data uji dan data latih. Pembagian dilakukan dengan parameter  $test\_size=0.2$ , artinya 80% data dipakai untuk pelatihan, dan sisanya 20% digunakan untuk pengujian. Fungsi  $train\_test\_split$  dari pustaka *scikit-learn* di Python digunakan untuk menjalankan proses ini.

Pada tahap evaluasi, performa model klasifikasi dinilai menggunakan *Confusion Matrix*, yang disajikan dalam bentuk tabel evaluasi. Melalui tabel ini, beberapa metrik utama seperti akurasi, presisi, recall, dan F-measure dihitung untuk mengukur kualitas prediksi model. Akurasi menunjukkan seberapa besar data yang diklasifikasikan dengan benar, presisi melihat ketepatan prediksi terhadap kelas positif, sementara recall mengukur seberapa baik model mendeteksi kasus positif yang sebenarnya. Kombinasi

metrik ini memberikan gambaran menyeluruh mengenai efektivitas model dalam melakukan klasifikasi secara akurat dan konsisten [25].

Sebelum melakukan *Confusion Matrix* evaluasi seperti akurasi, presisi, recall, dan F-measure, hal pertama yang perlu diketahui adalah nilai TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*). TP adalah jumlah data yang benar-benar termasuk kategori positif dan berhasil dikenali dengan tepat oleh model. TN mencerminkan data negatif yang juga diklasifikasikan dengan benar. Di sisi lain, FP terjadi ketika data negatif keliru diprediksi sebagai positif, dan FN muncul saat data positif justru dianggap sebagai negatif. Keempat komponen ini memainkan peran penting dalam menilai seberapa baik model dapat mengenali pola dan membuat prediksi yang tepat [25].

Adapun rumus untuk menghitung akurasi, presisi, recall, dan F-measure adalah sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$Precision = \frac{TP}{FP + TP} \quad (9)$$

$$Recall = \frac{TP}{FN + TP} \quad (10)$$

$$F - Score\ atau\ F - measure = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \quad (11)$$

## 3. Hasil dan Pembahasan

### 3.1 Transformasi Data

Pada tahap transformasi data, maka dilihat terlebih dulu pelabelan data yang digunakan. Adapun label data yang digunakan adalah pada Tabel 3.

Tabel 3. Label data gejala penyakit THT

No	Gejala	Label
1	telinga berdengung	0: No, 1: Yes
2	pusing	0: No, 1: Yes
3	nyeri kepala	0: No, 1: Yes
4	pendengaran turun	0: No, 1: Yes
5	nyeri leher	0: No, 1: Yes
6	hidung buntu	0: No, 1: Yes
7	bersin	0: No, 1: Yes
8	nyeri telan	0: No, 1: Yes
9	tenggorokan panas sakit	0: No, 1: Yes
10	tenggorokan kering	0: No, 1: Yes
11	penciuman terganggu	0: No, 1: Yes
12	iritasi hidung	0: No, 1: Yes
13	pilek menahun	0: No, 1: Yes
14	badan panas	0: No, 1: Yes
15	sulit buka mulut	0: No, 1: Yes
16	ingus darah	0: No, 1: Yes
17	sakit kepala	0: No, 1: Yes
18	leher kaku	0: No, 1: Yes
19	leher bengkak	0: No, 1: Yes
20	sesak nafas	0: No, 1: Yes
21	telinga penuh cairan	0: No, 1: Yes
22	pengelompokan sistol	1: Hipotensi, 2: Normal, 3: Hipertensi

No	Gejala	Label
23	pengelompokan diastol	1: Hipotensi, 2: Normal, 3: Hipertensi
24	pengelompokan temperature	1: Hipotermia, 2: Normal, 3: Demam
25	pengelompokan pulse	1: Tidak normal, 2: Normal
26	pengelompokan breath	1: Tidak normal, 2: Normal
27	pengelompokan height weight	1: BB kurang, 2: Normal, 3: BB berlebih, 4: Obesitas

Adapun label kelompok diagnosa penyakit THT sebagai atribut target adalah pada Tabel 4. Contoh data adalah seperti pada Tabel 5.

Tabel 4. Label data diagnosa penyakit THT

No	Diagnosa penyakit THT	Label
1	THT dengan atau tanpa kormodibitas	1: THT tanpa kormodibitas, 2: THT dengan kormodibitas

Tabel 5. Contoh data yang digunakan

No	Variabel	Data 1	Data 2	Data 3	Data 4
1	G01-THT	0	0	0	0
2	G02-THT	0	0	0	0
3	G03-THT	0	0	0	0
4	G04-THT	1	1	0	0
5	G05-THT	0	0	0	0
6	G06-THT	0	0	0	0
7	G07-THT	0	0	0	0
8	G08-THT	0	0	0	1
9	G09-THT	0	0	0	1
10	G10-THT	0	0	0	0
11	G11-THT	0	0	0	0
12	G12-THT	0	0	0	0
13	G13-THT	0	0	0	0
14	G14-THT	0	0	0	0
15	G15-THT	0	0	0	0
16	G16-THT	0	0	0	0
17	G17-THT	0	0	0	0
18	G18-THT	0	0	0	0
19	G19-THT	0	0	0	0
20	G20-THT	0	0	0	0
21	G21-THT	0	0	0	0
22	G22-THT	2	2	3	2
23	G23-THT	1	2	2	2
24	G24-THT	1	1	1	1
25	G25-THT	2	1	2	2
26	G26-THT	1	1	2	2
27	G27-THT	3	3	3	1
28	P-THT	1	1	1	2

### 3.2 Distribusi Data

Pada penelitian ini, distribusi data menggunakan metode *oversampling* yaitu *SMOTE* (*Synthetic Minority Oversampling Technique*). Dalam upaya mengatasi ketidakseimbangan distribusi kelas pada data, *SMOTE*

menciptakan sampel baru secara sintesis di antara dua data yang berasal dari kelas minoritas. Dengan langkah ini, proporsi data menjadi lebih seimbang, sehingga algoritma pembelajaran mesin bisa belajar dari data secara lebih adil dan akurat [26].

Berdasarkan hasil evaluasi, didapat hasil distribusi data pada Gambar 2.

[2d] Distribusi Diagnosa (Sebelum SMOTE):

```
1 1483
2 1210
Name: diagnosa, dtype: int64
Distribusi seimbang, SMOTE tidak diterapkan.
```

Gambar 2. Hasil distribusi data *SMOTE* menggunakan python

Dari gambar 2, jumlah data untuk masing-masing kategori diagnosa tergolong cukup merata, sehingga tidak perlu dilakukan penyeimbangan data tambahan seperti *SMOTE*. Model dapat dilatih langsung tanpa resampling karena tidak ada ketimpangan kelas yang signifikan.

### 3.3 Seleksi Fitur

Berdasarkan hasil analisis menggunakan metode ANOVA, diperoleh sejumlah fitur yang memiliki kontribusi paling signifikan terhadap variabel target dalam proses klasifikasi. Penilaian terhadap masing-masing fitur dilakukan melalui perhitungan *F-Score*, yang merefleksikan tingkat perbedaan antar fitur terhadap variasi dalam kelompok data. Pada penelitian ini, sebanyak 15 fitur dengan *F-Score* tertinggi yang akan dipilih untuk tahap pengembangan model klasifikasi.

Tabel F-Score dan P-Value:

	Feature	F-Score	P-Value	Selected
12	pilek_menahun	43.997185	3.960446e-11	True
3	pendengaran_turun	13.113837	2.985360e-04	True
0	telinga_berdengung	8.452878	3.674475e-03	True
6	bersin	8.072787	4.527304e-03	True
26	height_weight	5.778960	1.628635e-02	True
8	tenggorokan_panas_sakit	4.696940	3.030382e-02	True
16	sakit_kepala	4.270685	3.887076e-02	True
21	sistol	3.594522	5.807732e-02	True
25	breath	2.645383	1.039682e-01	True
23	temperature	1.983603	1.591268e-01	True
4	nyeri_leher	1.891279	1.691722e-01	True
1	pusing	1.853301	1.735140e-01	True
17	leher_kaku	1.632818	2.014240e-01	True
11	iritasi_hidung	1.316130	2.513897e-01	True
22	diastol	0.848693	3.570050e-01	True
14	sulit_buka_mulut	0.815858	3.664749e-01	False
13	badan_panas	0.717918	3.969032e-01	False
2	nyeri_kepala	0.528702	4.672165e-01	False
10	penciuman_terganggu	0.326708	5.676516e-01	False
24	pulse	0.293603	5.879656e-01	False
15	ingus_darah	0.243810	6.215078e-01	False
18	leher_bengkak	0.178963	6.722993e-01	False
7	nyeri_telan	0.171752	6.785926e-01	False
5	hidung_buntu	0.131511	7.168985e-01	False
19	sesak_nafas	0.095970	7.567439e-01	False
9	tenggorokan_kering	NaN	NaN	False
20	telinga_penuh_cairan	NaN	NaN	False

Gambar 3. Hasil seleksi fitur dengan *F-Score* tertinggi menggunakan python

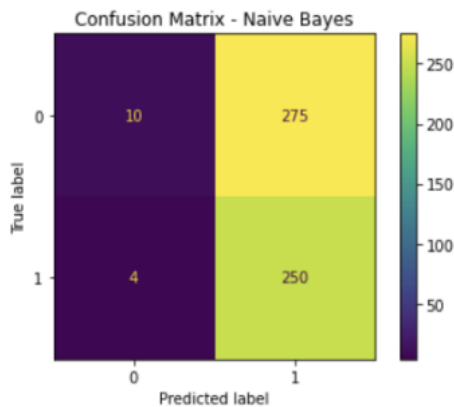
Berdasarkan gambar 3, maka dapat diketahui bahwa fitur-fitur atau atribut gejala penyakit THT yang akan dipilih adalah pada Tabel 6.

Tabel 6. Variabel yang dipilih

No	Gejala penyakit THT yang dipilih	
1	G01-THT	telinga berdengung
2	G02-THT	pusing
3	G04-THT	pendengaran turun
4	G05-THT	nyeri leher
5	G07-THT	bersin
6	G09-THT	tenggorokan panas sakit
7	G12-THT	iritasi hidung
8	G13-THT	pilek menahun
9	G17-THT	sakit kepala
10	G18-THT	leher kaku
11	G22-THT	pengelompokan sistol
12	G23-THT	pengelompokan diastol
13	G24-THT	pengelompokan temperature
14	G26-THT	pengelompokan breath
15	G27-THT	pengelompokan height weight

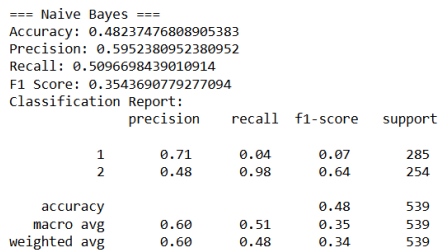
### 3.4 Pembangunan dan Evaluasi Model

Tahap ini dilakukan pembangunan model untuk proses training data yang diperoleh, kemudian dilakukan proses testing guna menentukan akurasi model yang dibangun. Hasil dari evaluasi model-model yang diterapkan adalah pada Gambar 4.



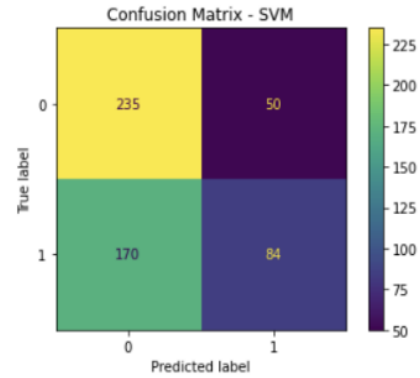
Gambar 4. Confusion Matrix model Naive Bayes

Model *Naive Bayes* ini cenderung sangat bias ke kelas 1, karena hampir semua data diprediksi sebagai kelas 1, sehingga performanya sangat buruk dalam mengenali kelas 0.



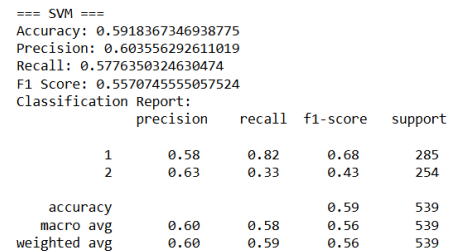
Gambar 5. Classification Report model Naive Bayes

Berdasarkan hasil pada Gambar 5, model *Naive Bayes* menunjukkan akurasi rendah (48%) dan performa sangat buruk pada kelas 1 (*F1-Score* hanya 0.07), menandakan ketidakseimbangan prediksi yang membuat model tidak efektif untuk klasifikasi dua kelas ini.



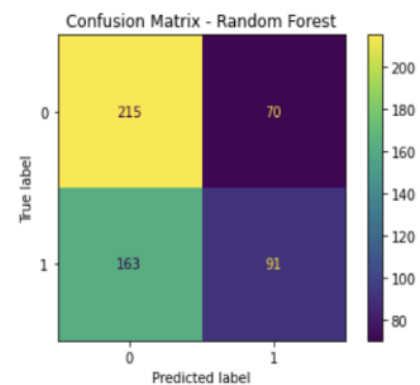
Gambar 6. Confusion Matrix model SVM

Berdasarkan *confusion matrix* pada Gambar 6, model SVM berhasil mengklasifikasikan 235 dari 285 sampel kelas 0 dengan benar, tetapi hanya mengidentifikasi 84 dari 254 sampel kelas 1, menunjukkan performa tinggi untuk kelas 0 namun rendah untuk kelas 1.



Gambar 7. Classification Report model SVM

Dari gambar 7, model SVM menghasilkan akurasi 59% dengan *F1-Score* yang cukup baik untuk kelas 1 (0.68) namun rendah untuk kelas 2 (0.43), menunjukkan model lebih efektif mengenali kelas 1 dibandingkan kelas 2.



Gambar 8. Confusion Matrix model Random Forest

*Confusion matrix* pada Gambar 8 menunjukkan bahwa model *Random Forest* mengklasifikasikan 215 dari 285 sampel kelas 0 dan 91 dari 254 sampel kelas 1 dengan benar, namun masih salah mengklasifikasikan masing-masing 70 dan 163 sampel, yang mencerminkan performa moderat dengan kecenderungan lebih akurat pada kelas 0.

```

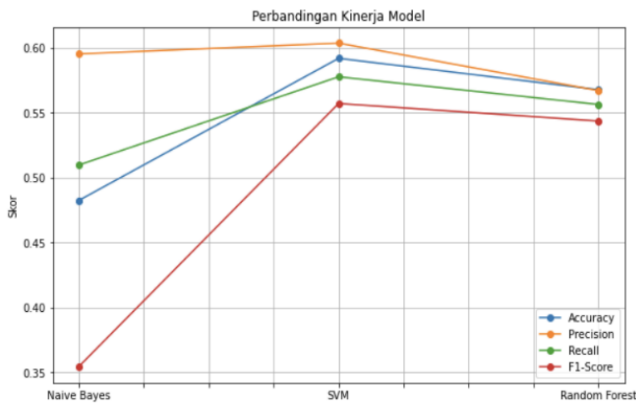
=== Random Forest ===
Accuracy: 0.5677179962894249
Precision: 0.5670002300437083
Recall: 0.5563268407238569
F1 Score: 0.5435606680114122
Classification Report:

```

	precision	recall	f1-score	support
1	0.57	0.75	0.65	285
2	0.57	0.36	0.44	254
accuracy			0.57	539
macro avg	0.57	0.56	0.54	539
weighted avg	0.57	0.57	0.55	539

Gambar 9. Classification Report model Random Forest

Hasil *classification report* untuk model *Random Forest* di atas mencapai akurasi 56,8% dengan *F1-Score* yang lebih baik pada kelas 1 (0,65) dibanding kelas 2 (0,44), menunjukkan performa lebih kuat dalam mengenali kelas 1 namun masih lemah dalam mendeteksi kelas 2. Adapun perbandingan hasil akurasi, presisi, recall, dan F-measure dari ketiga model adalah pada gambar 10



Gambar 10. Komparasi hasil evaluasi model *Naive Bayes*, *SVM*, dan *Random Forest*

Berdasarkan hasil komparasi pada Gambar 10, menunjukkan bahwa model *Naive Bayes* menunjukkan performa yang kurang optimal, dengan akurasi dan recall yang rendah serta ketimpangan besar antara precision dan recall yang menyebabkan *F1-Score* sangat rendah, menandakan ketidakseimbangan model dalam mengenali kelas, khususnya kelas minoritas. Model *SVM* menunjukkan kinerja terbaik di antara ketiga model, dengan nilai tertinggi pada semua metrik (akurasi, presisi, recall, dan F1) serta keseimbangan yang baik antar metrik, menandakan stabilitas yang tinggi dalam memprediksi kedua kelas. Sedangkan model *Random Forest* menunjukkan performa yang lebih baik dibandingkan *Naive Bayes* namun masih di bawah *SVM*, dengan *F1-Score* yang cukup baik berkat keseimbangan relatif antara precision dan recall, meskipun recall yang lebih rendah mengindikasikan masih adanya kegagalan dalam mengenali kelas target.

Dari ketiga model yang diuji, *SVM* terbukti sebagai model terbaik dengan pencapaian tertinggi pada semua metrik dengan tingkat akurasi 59%, sementara *Random Forest* menunjukkan performa menengah dengan tingkat akurasi 57%, dan *Naive Bayes* menjadi yang terendah karena kelemahan signifikan pada *F1-Score*, yang menunjukkan akurasi 48%.

Adapun performa yang relatif rendah dari seluruh model kemungkinan disebabkan karena data penyakit THT dengan atau tanpa komorbiditas merupakan kasus di mana gejala antar jenis penyakit dapat sangat mirip (*overlapping*). Selain itu, meskipun sudah dilakukan seleksi fitur menggunakan *ANOVA F-Score* dan *SelectKBest*, fitur yang digunakan sebagian besar adalah gejala biner (0/1), sehingga data kurang informatif secara individual, dan belum cukup membedakan antara THT dengan atau tanpa komorbiditas. Kemudian walaupun data sudah cukup merata, menjadikan *SMOTE* tidak dilakukan, membuat model lebih dominan memprediksi kelas mayoritas (THT tanpa komorbiditas). Namun demikian, perlu dilakukan penelitian lebih lanjut untuk mengetahui faktor-faktor yang mempengaruhi rendahnya nilai akurasi pada semua algoritma yang dipakai di penelitian ini.

### Ucapan Terimakasih

Penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada Politeknik Negeri Cilacap atas dukungan pendanaan melalui Kontrak Penelitian Internal Tahun Anggaran 2024 (Nomor: 076/PL43/AL.04/2024), yang telah memungkinkan penelitian ini dapat terlaksana dengan baik.

### Daftar Pustaka

- [1] R. Cohen *et al.*, "Antimicrobial treatment of ENT infections," *Infect. Dis. Now*, vol. 53, no. 8, p. 104785, 2023, doi: 10.1016/j.idnow.2023.104785.
- [2] A. Kaspar and S. Pifeleti, "A call to action for the inclusion of ENT/Audiology services in the public health approach to addressing non-communicable diseases in the Pacific Islands," *Public Heal. Pract.*, vol. 2, no. April, p. 100123, 2021, doi: 10.1016/j.puhip.2021.100123.
- [3] I. F. Rahman, W. A. Shestia, S. D. Rama, S. Azzahra, and A. A. Dermawan, "Klasifikasi Diagnosa Pasien Di Klinik Sri Dengan Metode *Decision Tree*," vol. 9, no. 01, 2024.
- [4] R. G. Wardhana, G. Wang, and F. Sibuea, "Penerapan *Machine Learning* Dalam Prediksi Tingkat Kasus Penyakit Di Indonesia," *J. Inf. Syst. Manag.*, vol. 5, no. 1, pp. 40–45, 2023, doi: 10.24076/joism.2023v5i1.1136.
- [5] A. D. Prasetyo, D. M. Zafarani, and M. Ammar, "Klasifikasi Penyakit Menular Dengan Algoritma *Random Forest* Dan *SVM*," *JURIHUM J. Inov. dan Hum.*, vol. 2, no. 3, pp. 416–419, 2024.
- [6] I. I. Lestari and A. Homaidi, "Komparasi Algoritma *Naive Bayes* Dan *Random Forest* Pada Klasifikasi Kanker Payudara," *Gudang J. Multidisiplin Ilmu*, vol. 2, pp. 778–785, 2024.
- [7] D. H. Depari, Y. Widiastiwi, and M. M. Santoni, "Perbandingan Model *Decision Tree*, *Naive Bayes* dan *Random Forest* untuk Prediksi Klasifikasi Penyakit Jantung," *Inform. J. Ilmu Komput.*, vol. 18, no. 3, p. 239, 2022, doi: 10.52958/iftk.v18i3.4694.
- [8] M. R. Zuhri, K. Kusriani, and D. Ariatmanto, "ANALISIS PERBANDINGAN ALGORITMA KLASIFIKASI UNTUK IDENTIFIKASI DIABETES DENGAN MENGGUNAKAN METODE *RANDOM FOREST* DAN *NAIVE BAYES*," *J. Inform. Teknol. dan Sains*, vol. 7, no. 1, pp. 11–20, 2025.
- [9] D. Septhya *et al.*, "Implementasi Algoritma *Decision Tree* dan *Support Vector Machine* untuk Klasifikasi Penyakit Kanker Paru," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 15–19, 2023, doi: 10.57152/malcom.v3i1.591.
- [10] K. Akmal, A. Faqih, and F. Dikananda, "Perbandingan

- Metode Algoritma *Naïve Bayes* Dan K-Nearest Neighbors Untuk Klasifikasi Penyakit Stroke,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 470–477, 2023, doi: 10.36040/jati.v7i1.6367.
- [11] M. Napiyah, R. A. Purnama, M. Raharjo, and W. Bismi, “Komparasi Algoritma Untuk Klasifikasi Penyakit Ispa (Infeksi Saluran Pernapasan Akut),” *J. Infortech*, vol. 4, no. 2, pp. 105–110, 2022, [Online]. Available: <https://ejournal.bsi.ac.id/ejurnal/index.php/infortech/article/view/13644/5720>
- [12] M. Jamil, F. Rozi, and Y. F. Saputra, “Komparasi Kinerja Algoritma *Machine Learning* Untuk Deteksi Penyakit Infeksi Saluran Pernapasan,” *J. Rekayasa Teknol. Inf.*, vol. 8, no. 1, pp. 84–92, 2024.
- [13] D. M. Musa *et al.*, “Penerapan Data Mining Untuk Klasifikasi Data Penjualan Pakan Ternak Terlaris Dengan Algoritma C4.5,” *J. Teknol. Inform. dan Komput.*, vol. 10, no. 1, pp. 168–182, 2024, doi: 10.37012/jtik.v10i1.1985.
- [14] A. R. Aziz, B. Warsito, and A. Prahutama, “Pengaruh Transformasi Data Pada Metode Learning Vector Quantization Terhadap Akurasi Klasifikasi Diagnosis Penyakit Jantung,” *J. Gaussian*, vol. 10, no. 1, pp. 21–30, 2021, doi: 10.14710/j.gauss.v10i1.30933.
- [15] S. Diantika, “Penerapan Teknik *Random Oversampling* Untuk Mengatasi Imbalance Class Dalam Klasifikasi Website Phishing Menggunakan Algoritma *Lightgbm*,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 19–25, 2023, doi: 10.36040/jati.v7i1.6006.
- [16] A. Z. Amrullah, A. Sofyan Anas, and M. A. J. Hidayat, “Analisis Sentimen Movie Review Menggunakan *Naïve Bayes* Classifier Dengan Seleksi Fitur Chi Square,” *Jurnal*, vol. 2, no. 1, pp. 40–44, 2020, doi: 10.30812/bite.v2i1.804.
- [17] A. A. Dhani, T. A. Y. Siswa, and W. J. Pranoto, “Perbaikan Akurasi *Random Forest* Dengan ANOVA Dan *SMOTE* Pada Klasifikasi Data Stunting,” *Teknika*, vol. 13, no. 2, pp. 264–272, 2024, doi: 10.34148/teknika.v13i2.875.
- [18] T. Gori, A. Sunyoto, and H. Al Fatta, “Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 1, pp. 215–224, 2024, doi: 10.25126/jtiik.20241118074.
- [19] W. A. Van Eeden *et al.*, “Predicting the 9-year course of mood and anxiety disorders with automated *Machine Learning* : A comparison between auto-sklearn , *Naïve Bayes* classifier , and traditional logistic regression,” *Psychiatry Res.*, vol. 299, no. October 2020, p. 113823, 2021, doi: 10.1016/j.psychres.2021.113823.
- [20] H. Nalatissifa, W. Gata, S. Diantika, and K. Nisa, “Perbandingan Kinerja Algoritma Klasifikasi *Naïve Bayes*, *Support Vector Machine* (SVM), dan *Random Forest* untuk Prediksi Ketidakhadiran di Tempat Kerja,” *J. Inform. Univ. Pamulang*, vol. 5, no. 4, p. 578, 2021, doi: 10.32493/informatika.v5i4.7575.
- [21] S. Suryani and M. Mustakim, “Estimasi Keberhasilan Siswa dalam Pemodelan Data Berbasis Learning Menggunakan Algoritma *Support Vector Machine*,” *Bull. Informatics Data Sci.*, vol. 1, no. 2, p. 81, 2022, doi: 10.61944/bids.v1i2.36.
- [22] M. G. Pradana, P. H. Saputro, and D. P. Wijaya, “Komparasi Metode *Support Vector Machine* Dan *Naïve Bayes* Dalam Klasifikasi Peluang Penyakit Serangan Jantung,” *Indones. J. Bus. Intell.*, vol. 5, no. 2, p. 87, 2022, doi: 10.21927/ijubi.v5i2.2659.
- [23] D. P. Sinambela, H. Naparini, M. Zulfadhilah, and N. Hidayah, “Implementasi Algoritma *Decision Tree* dan *Random Forest* dalam Prediksi Perdarahan Pascasalin,” *J. Inf. dan Teknol.*, vol. 5, no. 3, pp. 58–64, 2023, doi: 10.60083/jidt.v5i3.393.
- [24] F. Diba, M. S. Lydia, and P. Sihombing, “Analisis *Random Forest* Menggunakan Principal Component Analysis Pada Data Berdimensi Tinggi,” *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2152–2160, 2023, doi: 10.33022/ijcs.v12i4.3329.
- [25] D. Putra and A. Wibowo, “Prediksi Keputusan Minat Penjurusan Siswa SMA Yadika 5 Menggunakan Algoritma *Naïve Bayes*,” *Pros. Semin. Nas. Ris. Dan Inf. Sci.*, vol. 2, pp. 84–92, 2020.
- [26] S. Sidiq, N. S. Mabur, S. T. Informatika, F. Teknik, and U. M. Tangerang, “Pengembangan Model Prediksi Risiko Diabetes Menggunakan Pendekatan *AdaBoost* dan Teknik *Oversampling*,” vol. 4, pp. 13–23, 2025.