

## Kombinasi Algoritma Gaussian Naïve Bayes Dan Adaboost Untuk Meningkatkan Akurasi Dalam Klasifikasi Penyakit Diabetes

Fitri Handayani\*<sup>1</sup>, Rahmad Firdaus<sup>2</sup>, Ashari Wahyudi<sup>3</sup>, Hasanatul Fu'adah Amran<sup>4</sup>, Reny Medikawati Taufiq<sup>5</sup>

<sup>1,2,3,4,5</sup>Teknik Informatika, Ilmu Komputer, Universitas Muhammadiyah Riau

<sup>1</sup>fitrihandayani@umri.ac.id, <sup>2</sup>rahmadfirdaus@umri.ac.id, <sup>3</sup>210401031@student.umri.ac.id, <sup>4</sup>Hasanatul@umri.ac.id,

<sup>5</sup>renymedikawati@umri.ac.id

### Abstract

*Diabetes mellitus is a chronic metabolic disease that can lead to serious complications if not detected early. This study aims to improve the accuracy of diabetes classification by combining the Naïve Bayes algorithm and Adaboost using ensemble learning techniques. Ensemble learning is a machine learning method that enhances model accuracy by combining predictions from several different models. This technique integrates models that may perform suboptimally individually to form a more superior model. Adaboost assigns higher weights to samples that are difficult to classify, making it effective in handling complex and imbalanced data. The dataset used in this study is from Sylhet Diabetes Hospital, Bangladesh, consisting of questionnaire data verified by doctors. Evaluation using the Confusion Matrix shows that the combination of Naïve Bayes and Adaboost significantly improves the accuracy of diabetes classification. This model achieved an accuracy of 96.1% with an 80:20 data split, higher than the standalone Naïve Bayes (87.69%). The highest precision (100%) was recorded at the 80:20 data split, with recall stable at 93.7%–94%, and the highest F1-Score at 96.7%. These results indicate that the combination of both algorithms through ensemble learning techniques complements each other and enhances classification performance, making it more effective in identifying diabetes.*

*Keywords: adaboost, confusion matrix, diabetes mellitus, ensemble learning, naïve bayes*

### Abstrak

Diabetes mellitus adalah penyakit metabolik kronis yang dapat menyebabkan komplikasi serius jika tidak terdeteksi dini. Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi diabetes dengan menggabungkan algoritma Gaussian Naïve Bayes dan Adaboost menggunakan teknik ensemble learning. Ensemble learning adalah metode dalam pembelajaran mesin yang meningkatkan akurasi model dengan menggabungkan prediksi dari beberapa model yang berbeda. Teknik ini mengintegrasikan model-model yang mungkin memiliki performa kurang optimal secara individu untuk membentuk model yang lebih unggul. Adaboost memberikan bobot lebih besar pada sampel yang sulit diklasifikasikan, sehingga efektif dalam menangani data yang kompleks dan tidak seimbang. Dataset yang digunakan berasal dari Sylhet Diabetes Hospital, Bangladesh, yang berisi data kuesioner yang telah diverifikasi oleh dokter. Evaluasi menggunakan Confusion Matrix menunjukkan bahwa kombinasi Gaussian Naïve Bayes dan Adaboost meningkatkan akurasi klasifikasi diabetes secara signifikan. Model ini mencapai akurasi 96.1% pada pembagian data 80:20, lebih tinggi dibandingkan Naïve Bayes tunggal (87.69%). Presisi tertinggi (100%) tercatat pada pembagian data 80:20, dengan recall stabil pada 93.7%–94%, dan F1-Score tertinggi sebesar 96.7%. Hasil ini menunjukkan bahwa kombinasi kedua algoritma melalui teknik ensemble learning dapat saling melengkapi dan meningkatkan performa klasifikasi, menjadikannya lebih efektif dalam identifikasi diabetes.

Kata kunci: adaboost, confusion matrix, diabetes mellitus, ensemble learning, naïve bayes

©This work is licensed under a Creative Commons Attribution - ShareAlike 4.0 International License

### 1. Pendahuluan

Diabetes mellitus (DM) merupakan penyakit metabolik kronis yang ditandai oleh tingginya kadar glukosa dalam darah akibat gangguan pada produksi atau efektivitas insulin. Penyakit ini menyebabkan berbagai komplikasi serius jika tidak terdeteksi dan ditangani sejak dini. Menurut data dari Organisasi Kesehatan Dunia (WHO), diabetes menempati urutan ketujuh sebagai penyebab kematian tertinggi di dunia, dengan sekitar 1,5 juta kematian setiap tahunnya. Lebih dari 422 juta orang di seluruh dunia menderita diabetes, dengan mayoritas kasus terjadi di negara-negara berpenghasilan rendah dan menengah [1]. Proyeksi global menunjukkan bahwa jumlah penderita diabetes akan terus meningkat pesat, dengan *International Diabetes Federation* (IDF) memprediksi bahwa pada

tahun 2030, jumlah penderita diabetes akan mencapai 643 juta orang, dan pada tahun 2045, jumlahnya akan melonjak menjadi 783 juta orang[2].

Di Indonesia, prevalensi diabetes juga mengalami peningkatan yang signifikan. Berdasarkan data IDF, pada tahun 2021 jumlah penderita diabetes di Indonesia mencapai 19,5 juta orang, menempatkan Indonesia sebagai negara kelima dengan jumlah penderita diabetes terbanyak di dunia. Angka ini diperkirakan akan melonjak menjadi 28,6 juta orang pada tahun 2045. Diabetes sering kali disebut sebagai "ibu dari segala penyakit" karena penyakit ini dapat memicu berbagai kondisi medis lain yang berpotensi meningkatkan risiko komplikasi serius, seperti penyakit jantung, gagal ginjal, gangguan penglihatan, dan stroke. Komplikasi-komplikasi tersebut tidak

hanya menurunkan kualitas hidup penderita, tetapi juga meningkatkan beban biaya perawatan kesehatan yang tinggi, yang berkontribusi pada peningkatan beban ekonomi dan sosial secara signifikan[3].

Di tingkat global maupun nasional, diabetes tidak hanya menjadi masalah kesehatan masyarakat, tetapi juga merupakan masalah yang membebani sistem kesehatan dan ekonomi. Penyakit ini mengganggu aktivitas produktif masyarakat, mengurangi kualitas hidup penderita, dan meningkatkan risiko penyakit lain. Oleh karena itu, upaya deteksi dini menjadi sangat penting, karena semakin cepat diabetes terdeteksi, semakin besar kemungkinan untuk mencegah komplikasi yang lebih serius. Meskipun diabetes memiliki gejala-gejala umum seperti sering merasa haus, sering buang air kecil, penurunan berat badan yang drastis, luka yang sulit sembuh, dan sering merasa lelah, banyak orang yang tidak menyadarinya, sehingga mereka terlambat mendapatkan perawatan yang diperlukan[3].

Di Indonesia, telah dilakukan berbagai upaya untuk menangani diabetes melitus. Program-program tersebut meliputi penyuluhan mengenai pencegahan, serta pelaksanaan program Kesehatan Dasar Masyarakat (KDM) yang menyediakan fasilitas pemeriksaan kadar gula darah secara rutin dengan biaya yang ditanggung oleh negara. Seseorang dinyatakan mengidap diabetes melitus jika hasil pemeriksaan menunjukkan kadar gula darah  $\geq 200$  mg/dL ketika diperiksa sewaktu atau dua jam setelah makan, dan  $\geq 126$  mg/dL jika diperiksa setelah puasa. Selain itu, pemerintah juga telah menyusun peta jalan (roadmap) untuk pengendalian kasus diabetes dan deteksi dini penyakit diabetes melitus[4].

Seiring dengan meningkatnya angka penderita diabetes, kebutuhan akan metode deteksi dini menjadi semakin mendesak. Deteksi dini sangat penting untuk meminimalkan risiko komplikasi yang lebih parah dan membantu dalam pengelolaan penyakit [5]. Teknologi machine learning (pembelajaran mesin) telah muncul sebagai pendekatan inovatif yang dapat meningkatkan ketepatan dan kecepatan dalam diagnosis dini penyakit diabetes. *Machine learning*, yang merupakan bagian dari kecerdasan buatan, memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa perlu diprogram secara eksplisit. Di bidang kesehatan, machine learning dapat menganalisis data medis yang kompleks secara cepat untuk mengidentifikasi pola-pola yang membantu dalam pengambilan keputusan klinis dan diagnosis yang lebih tepat [5].

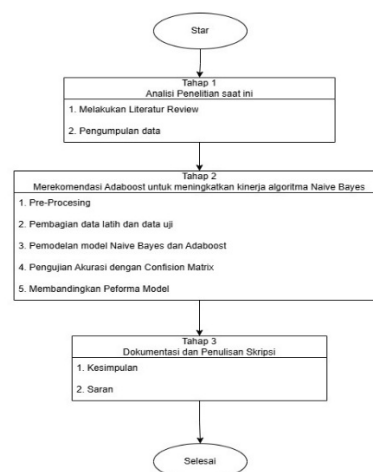
Beberapa penelitian sebelumnya menunjukkan bahwa berbagai algoritma machine learning dapat digunakan untuk mendeteksi diabetes dengan akurasi yang tinggi. Sebagai contoh, penelitian terdahulu menunjukkan bahwa algoritma *Random Forest* mampu mencapai akurasi hingga 97,88% [6], sementara penelitian oleh Septiani dan Rohwadi menemukan bahwa algoritma

*Decision Tree* mencapai akurasi 95,96%[5]. Selain itu, penelitian dari Siridion dan Siregar membuktikan bahwa pendekatan berbasis machine learning, khususnya algoritma supervised learning seperti *Random Forest*, *Support Vector Machine* (SVM), *K-Nearest Neighbor* (K-NN), dan *Decision Tree*, terbukti efektif dalam mengidentifikasi risiko diabetes dengan tingkat akurasi yang tinggi. *Random Forest* bahkan mencapai akurasi tertinggi dalam prediksi diabetes, yaitu 98,71%, mengungguli algoritma lainnya [4]. Penelitian lain juga menunjukkan bahwa *Decision Tree* memiliki akurasi tertinggi yaitu 95,96%, diikuti oleh *Naive Bayes* dengan akurasi 87,69%, sementara *Neural Network* menunjukkan akurasi yang paling rendah, yaitu 61,54%. Penelitian ini menunjukkan bahwa *Naive Bayes* memiliki tingkat akurasi sekitar 87,69%, yang masih rendah dibandingkan dengan *Decision Tree*[5].

Untuk meningkatkan kinerja algoritma *Naive Bayes*, beberapa studi telah mengusulkan penerapan metode penguatan, seperti *Adaboost*, yang dapat digunakan untuk meningkatkan akurasi algoritma klasifikasi, termasuk *Naive Bayes*. *Adaboost* telah terbukti meningkatkan akurasi *Naive Bayes* hingga 7,42% untuk prediksi stroke[7]. Hasil penelitian lain menunjukkan bahwa dengan menggunakan *Adaboost*, akurasi *Naive Bayes* meningkat dari 0.7608 menjadi 0.7694 [8]. Oleh karena itu, penelitian ini bertujuan untuk mengoptimalkan algoritma *Naive Bayes* menggunakan teknik *Adaboost* guna meningkatkan akurasi prediksi diabetes. Dengan pendekatan ini, diharapkan *Naive Bayes* dapat menjadi lebih handal sebagai alat deteksi dini diabetes yang cepat dan akurat, serta mendukung upaya penanganan dan pengendalian diabetes secara lebih efektif.

## 2. Metode Penelitian

Penelitian ini meliputi tiga tahapan penting, yakni analisis penelitian, merekomendasikan *Adaboost* untuk meningkatkan kinerja algoritma *Naive Bayes*, dan dokumentasi penulisan skripsi. Rincian mengenai langkah-langkah tersebut dapat ditemukan dalam urutan yang terperinci pada Gambar 1 di bawah ini:



Gambar 1. Alur Kerja Penelitian

2.1. Analisis Penelitian Saat Ini

2.1.1 Literature review

Pada tahap ini, langkah pertama yang dilakukan adalah studi literatur terkait dengan penggunaan machine learning dalam klasifikasi diabetes. Peneliti mempelajari penelitian terdahulu mengenai algoritma yang telah digunakan dalam diagnosis diabetes, seperti *Naïve Bayes*, *Decision Tree*, dan *Random Forest*, serta teknik penguatan seperti *Adaboost*. Setelah itu, peneliti menentukan pendekatan yang paling sesuai untuk penelitian ini, yaitu kombinasi *Naïve Bayes* dengan *Adaboost* untuk meningkatkan akurasi klasifikasi.

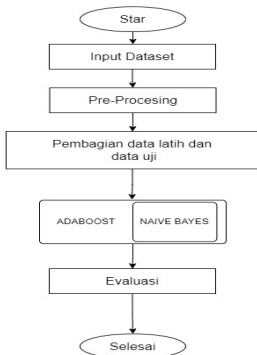
2.1.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah *dataset diabetes* yang diambil dari *Sylhet Diabetes Hospital*, Bangladesh. *Dataset* ini terdiri dari data pasien yang telah diverifikasi oleh dokter dan mencakup 17 atribut, antara lain usia, jenis kelamin, gejala-gejala terkait diabetes, dan hasil klasifikasi (positif atau negatif). Data yang digunakan berjumlah 520 sampel, dengan masing-masing sampel memiliki 17 atribut. Terlihat pada tabel 1 di bawah ini:

Tabel 1. *Dataset*

No.	Atribut	Contoh Nilai
1	Age	16-90
2	Gender	Male, Female
3	Polyuria	Yes, No
4	Polydipsia	Yes, No
5	sudden weight loss	Yes, No
6	weakness	Yes, No
7	Polyphagia	Yes, No
8	Genital thrush	Yes, No
9	visual blurring	Yes, No
10	Itching	Yes, No
11	Irritability	Yes, No
12	delayed healing	Yes, No
13	partial paresis	Yes, No
14	muscle stiffness	Yes, No
15	Alopecia	Yes, No
16	Obesity	Yes, No
17	class	Positive, Negative

2.2 Rekomendasi metode *Adaboost* untuk meningkatkan kinerja algoritma *Naïve bayes*



Gambar 2. Rekomendasi Metode *Adaboost* untuk Meningkatkan Kinerja Algoritma *Naïve Bayes*

2.2.1 Input Data

Tahap input data merupakan tahapan pertama yang dilakukan pada proses ini, data di inputkan kedalam google colab untuk di olah dan di proses ke tahap selanjutnya. Berikut contoh beberapa data yang di tampilkan setelah proses input pada gambar 3.3 di bawah ini:

No	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
1	40	Male	No	Yes	No	Yes	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Positive
2	88	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
3	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
4	45	Male	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No	Positive
5	60	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive
6	45	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No	No	Positive
7	47	Male	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No	No	Positive
8	66	Male	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No	No	Positive
519	33	Female	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Negative
520	42	Male	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Negative

Gambar 3. Input Data

2.2.2 Pre-Processing

Pra-pemrosesan adalah langkah penting sebelum data digunakan dalam model. Tahapan pra-pemrosesan yang dilakukan meliputi:

1. *Encoding Data Kategorikal*: Data bertipe kategorikal perlu diubah menjadi data numerik karena model tidak mampu memproses data dalam bentuk kategorikal secara langsung [9] Kolom-kolom dalam dataset yang memiliki nilai kategorikal, seperti jenis kelamin (*Male/Female*) dan gejala-gejala diabetes (misalnya *polyuria*, *polydipsia*), diubah menjadi format numerik menggunakan metode *LabelEncoder*. Proses ini memastikan bahwa model machine learning dapat memproses data dalam format yang sesuai.

Gambar 4. *Encode*

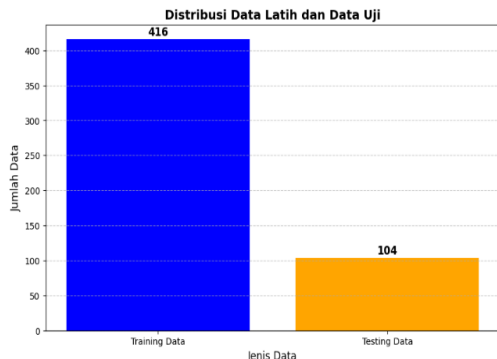
Terlihat pada Gambar 4 seluruh tipe data di ganti jadi angka, agar dapat digunakan oleh algoritma *machine learning*.

2. *Pembagian Fitur dan Label*: Data kemudian dipisahkan menjadi dua bagian, yaitu fitur (X) dan label (y). Fitur (X) terdiri dari semua kolom kecuali kolom terakhir, yang merupakan label atau target (positif atau negatif). Proses ini sangat penting karena kualitas data yang buruk dapat menghasilkan kesimpulan yang tidak akurat dan berdampak pada kualitas keputusan yang diambil berdasarkan data tersebut[8].

Gambar 5. Pembagian Fitur dan Label

2.2.3 Pembagian data latih dan data uji

Dataset dibagi menjadi dua bagian, yaitu data latih (80%) dan data uji (20%). Pembagian ini dilakukan dengan menggunakan fungsi *train\_test\_split* dari *library scikit-learn* dengan parameter *stratify* untuk memastikan distribusi label yang konsisten di kedua subset data.



Gambar 6. Pembagian data latih dan data uji (80:20)

Pada Gambar 6 sesuai perbandingan 80:20 terlihat jumlah data latih berjumlah 416 data, sedangkan data uji berjumlah 104 data.

### 2.2.4 Pemodelan

Setelah tahap preprocessing selesai, dilanjutkan dengan tahap pelatihan model. Tahap ini akan melakukan klasifikasi menggunakan algoritma Naïve Bayes dengan metode Adaboost.

#### 1. Naïve Bayes

Algoritma *Naive bayes* adalah metode klasifikasi statistik yang digunakan untuk memperkirakan kemungkinan sebuah data tergolong dalam kelas tertentu berdasarkan perhitungan probabilitas. Algoritma ini didasarkan pada Teorema Bayes yang dikembangkan oleh Thomas Bayes dan dikenal karena kemampuannya memproses database besar dengan cepat dan akurat. Dalam *machine learning*, *Naive bayes* sering diterapkan karena mampu memberikan akurasi tinggi melalui perhitungan yang sederhana[10]. Algoritma *Naive bayes* hanya mendukung atribut dengan tipe data diskrit atau yang telah didiskritisasi, sehingga tidak mendukung atribut dengan nilai kontinu (numerik). Selain itu, algoritma ini mengasumsikan bahwa semua atribut bersifat independen dan dapat berkontribusi terhadap prediksi dari atribut target[11]. Rumus teorema Naïve Bayes seperti yang di jelaskan pada rumus 1 di bawah ini [11]:

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

Dimana:

- X = Data dengan *class* yang belum diketahui
- H = Hipotesis data X merupakan suatu *class* spesifik
- P(H|X) = Probabilitas hipotesis H berdasarkan kondisi x (posteriori prob.)
- P(H) = Probabilitas hipotesis H (prior prob.)
- P(X|H) = Probabilitas X berdasarkan kondisi tersebut
- P(X) = Probabilitas dari X

#### 2. Adaboost

Algoritma *Adaptive Boosting (Adaboost)* membentuk pengklasifikasi yang kuat dengan menggabungkan sejumlah pengklasifikasi sederhana (lemah)nya, setiap sampel diberi bobot yang sama. Setelah setiap langkah klasifikasi, bobot sampel yang diklasifikasikan dengan benar akan dikurangi, sedangkan bobot sampel yang salah akan ditingkatkan. Proses ini terus berlanjut hingga mencapai ambang batas tertentu atau jumlah siklus maksimum[12]. Algoritma *Ensemble AdaBoost* termasuk dalam metode boosting. Secara umum, algoritma boosting memiliki performa yang lebih unggul dibandingkan bagging, meskipun tidak selalu konsisten di semua kasus. Boosting mampu meningkatkan kinerja pengklasifikasi dalam berbagai situasi, termasuk saat menghadapi data yang tidak seimbang[13]. *Adaboost* dapat dideskripsikan melalui persamaan rumus 2 di bawah in[14]:

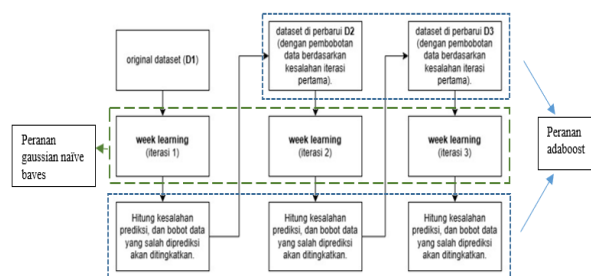
$$f(x) = \sum_{t=1}^t \alpha_t h_t(x) \quad (2)$$

Dimana:

- $\alpha_t$  : Tingkat pembelajaran (learning rate)
- $h_t$  : Pengklasifikasian dasar/ lemah
- F(x) : Pengklasifikasian kuat

#### 3. Ensemble learning

*Ensemble learning* merupakan salah satu teknik dalam pembelajaran mesin yang bertujuan untuk meningkatkan kinerja dan akurasi model dengan menggabungkan prediksi dari berbagai model atau algoritma yang berbeda. Teknik ini mengintegrasikan beberapa model atau algoritma yang mungkin memiliki performa kurang optimal secara individu untuk membentuk model yang memiliki performa lebih unggul dibandingkan dengan model tunggal[15]. Model pembelajaran yang di gunakan pada pemodelan adaboost ini merupakan pembelajaran berurutan, di mana berbagai model dibuat secara berurutan dan kesalahan model sebelumnya dipelajari oleh penerusnya. Hal ini bertujuan untuk mengeksploitasi ketergantungan antar model dengan memberikan bobot yang lebih tinggi.



Gambar 7. Pemodelan

Pada gambar 7 terdapat langkah langkah pemodelan, dimulai pada iterasi pertama, *Gaussian Naive Bayes* sebagai week learning diterapkan pada dataset asli (D1) untuk menghasilkan prediksi. Setelah itu, kesalahan prediksi dihitung dan bobot data yang salah diprediksi akan ditingkatkan. Hal ini bertujuan

agar model lebih memfokuskan perhatian pada data yang sulit diprediksi di iterasi berikutnya. Pada iterasi kedua, dataset diperbarui menjadi D2 dengan pembobotan berdasarkan kesalahan prediksi yang terjadi pada iterasi pertama. *Gaussian Naive Bayes* kemudian diterapkan pada dataset yang telah diperbarui ini (D2). Kemudian, kesalahan prediksi dihitung kembali dan bobot data yang masih salah diprediksi akan ditingkatkan lebih lanjut. Pada iterasi ketiga, dataset diperbarui lagi menjadi D3 dengan pembobotan yang lebih kuat berdasarkan kesalahan yang terjadi di iterasi sebelumnya. *Gaussian Naive Bayes* diterapkan sekali lagi pada dataset yang diperbarui ini (D3).

Akhirnya, pada tahap prediksi akhir, hasil prediksi akan dihasilkan dari gabungan model-model yang dilatih pada setiap iterasi. Prediksi ini akan lebih akurat karena setiap iterasi memberikan bobot yang lebih kuat pada data yang salah diprediksi, sehingga model secara keseluruhan menjadi lebih baik dalam menangani data yang sulit diprediksi. Gabungan model ini menghasilkan prediksi yang lebih kuat dan lebih akurat dibandingkan jika hanya mengandalkan satu model saja.

Kode Program

```
base_estimator = GaussianNB()
# AdaBoost classifier with Naive Bayes as weak
Learner
nb_adaboost_model =
AdaBoostClassifier(estimator=base_estimator,n_e
stimators=7,random_state=42,learning_rate=0.8)
```

Adapun parameter utama yang di gunakan pada pemodelan ini yaitu :

- a. *estimator=base\_estimator*: *Gaussian Naive bayes* digunakan sebagai *weak learner*.
- b. *n\_estimators=7*: Model akan melakukan 7 iterasi untuk meningkatkan akurasi prediksi dari *base\_estimator* (karna dengan 7 kali iterasi sudah terjadi peningkatan maksimal, walaupun iterasi di tingkatan tapi tetap saja hasilnya akan sama dengan 7 kali iterasi).
- c. *random\_state=42*: Menetapkan nilai acak (random seed) untuk memastikan hasil yang konsisten dan dapat direproduksi. Angka 42 digunakan sebagai nilai acak yang umum dan banyak dipakai.
- d. *learning\_rate=0.8*: Parameter ini mengontrol seberapa besar kontribusi setiap pembelajar lemah terhadap model akhir. Semakin kecil nilai learning rate, semakin kecil pengaruh tiap pembelajar, namun model mungkin membutuhkan lebih banyak estimators untuk bekerja dengan baik. Nilai 0.8 memberikan kontribusi yang cukup besar tanpa mendominasi pembelajaran.

Selanjutnya di lakukan pelatihan data menggunakan data yang sudah di pisah menjadi data latih sebelumnya.

Kode Program

```
y_pred = nb_adaboost_model.predict(X_test)
```

Dari *code* diatas terlihat proses pelatihan model *Adaboost* dimulai. Model *Adaboost Classifier* yang telah didefinisikan sebelumnya menggunakan *Gaussian Naive Bayes* sebagai estimator dasar untuk setiap iterasi *boosting*. Proses pelatihan dimulai dengan memberikan data fitur ( $X_{train}$ ) dan label target ( $y_{train}$ ) ke dalam model. Selama pelatihan, *Adaboost* bekerja secara iteratif dalam beberapa langkah (dalam kasus ini, sebanyak 7 iterasi, sesuai dengan parameter *n\_estimators=7*).

Pada setiap iterasi, model dasar (*GaussianNB*) dilatih menggunakan data yang ada, dan bobot diberikan lebih tinggi pada data yang sebelumnya salah diklasifikasikan oleh model sebelumnya. Hal ini memungkinkan model untuk lebih fokus pada data yang lebih sulit untuk diprediksi.

Setelah 7 iterasi selesai, model *Adaboost* menghasilkan sebuah ensemble yang terdiri dari 7 model dasar dengan bobot masing-masing, yang secara kolektif meningkatkan akurasi prediksi. Setelah proses ini selesai, model sudah siap untuk digunakan dalam membuat prediksi atau evaluasi lebih lanjut menggunakan data baru. Dengan demikian, proses fit() mengoptimalkan model dengan cara menggabungkan kekuatan dari beberapa model sederhana untuk menghasilkan model yang lebih kuat dan akurat.

3. Hasil dan Pembahasan

3.1. Hasil Evaluasi

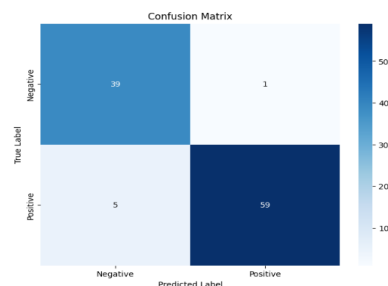
Evaluasi menggunakan Confusion Matrix bertujuan untuk mengevaluasi kinerja model dengan menghitung True Positives, True Negatives, False Positives, dan False Negatives seperti pada Tabel 2 [12]:

Tabel 2. Confusion Matrix

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

Dari sini, metrik seperti akurasi, *precision*, *recall*, dan *F1-score* dihitung.

3.1.1 Hasil Evaluasi Model *Adaboost*

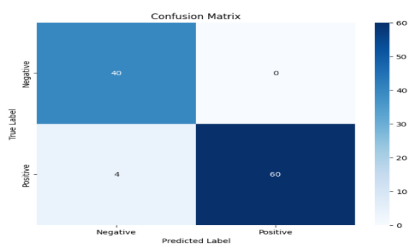


Gambar 8. Hasil Evaluasi Model *Adaboost*

Gambar 8 merupakan hasil evaluasi model *Adaboost* dimana:

True Positive (TP): Jumlah sampel kelas positif yang diprediksi dengan benar sebagai positif berjumlah 59.  
True Negative (TN): Jumlah sampel kelas negatif yang diprediksi dengan benar sebagai negatif berjumlah 39.  
False Positive (FP) (Error Type I): Jumlah sampel kelas negatif yang diprediksi salah sebagai positif berjumlah 1.  
False Negative (FN) (Error Type II): Jumlah sampel kelas positif yang diprediksi salah sebagai negatif berjumlah 5.

### 3.1.2 Hasil Evaluasi Model Kombinasi *Adaboost* dan *Naïve Bayes*



Gambar 9. Hasil evaluasi model Kombinasi *Adaboost* dan *Naïve Bayes*

Gambar 9 merupakan hasil evaluasi kombinasi *Adaboost* dan *Naïve Bayes* dimana:

True Positive (TP): Jumlah sampel kelas positif yang diprediksi dengan benar sebagai positif berjumlah 60.  
True Negative (TN): Jumlah sampel kelas negatif yang diprediksi dengan benar sebagai negatif berjumlah 40.  
False Positive (FP) (Error Type I): Jumlah sampel kelas negatif yang diprediksi salah sebagai positif berjumlah 0.  
False Negative (FN) (Error Type II): Jumlah sampel kelas positif yang diprediksi salah sebagai negatif berjumlah 4.

Hasil evaluasi model menggunakan confusion matrix untuk data uji yang terdiri dari dua kelas: Positif (diabetes) dan Negatif (bukan diabetes) dan di dapatkan hasil akurasi, precision, recall, dan F1-score dihitung.

Tabel 3. Hasil Confusion Matrix

	Adaboost	Gaussian Naïve Bayes + Adaboost
Akurasi	94.20%	<b>96.10%</b>
Precision	98.30%	100%
Recall	92.10%	93.70%
F1-Score	95.10%	96.70%

Pada tabel 2 di jelaskan hasil dari evaluasi confusion matrix diantaranya:

Akurasi: Model yang menggunakan *Gaussian Naïve Bayes* dan *Adaboost* mencapai akurasi sebesar 96.1%, jauh lebih tinggi dibandingkan *Adaboost* tunggal yang hanya mencapai 94.20%.

Precision: Precision tertinggi tercatat pada kedua model (100%), yang menunjukkan bahwa model mampu mengidentifikasi semua data positif dengan tepat tanpa kesalahan.

Recall: Recall stabil antara 93.7% hingga 94% pada kedua model, yang mengindikasikan bahwa model

cukup baik dalam mengidentifikasi kasus positif diabetes.

F1-Score: Model dengan kombinasi *Gaussian Naïve Bayes* dan *Adaboost* memiliki F1-score tertinggi sebesar 96.7%, yang menunjukkan keseimbangan antara precision dan recall yang lebih baik.

### 3.2. Perbandingan Performa Model

Berdasarkan hasil peforma model yang kita dapatkan pada tahap sebelum nya kita bisa membandingkan dengan hasil penelitian sebelumnya yang pernah dilakukan oleh (Septiani and Marlina 2021) dengan menggunakan algoritma *Naïve Bayes* yang mendapatkan akurasi 87,69%.

Tabel 4. Perbandingan Hasil Akurasi

Algoritma	Akurasi
Naïve Bayes (80:20)(Septiani and Marlina 2021)	87.69%
Adaboost (Our purpose)	94.20%
Adaboost + GaussianNB (Our purpose)	96.10%

Pada Tabel 3 Perbandingan Akurasi, perbandingan yang di lakukan berdasarkan nilai akurasi yang di dapat dari penelitian sebelum nya yang menggunakan algoritma *Naïve Bayes*,dibandingkan dengan yang menggunakan algoritma *adaboost* dan kombinasi aloritma *gaussian naïve bayes* dengan *adaboost* .

## 4. Kesimpulan

Peningkatan performa yang signifikan terlihat ketika algoritma *Gaussian Naïve Bayes* digabungkan dengan *Adaboost*. Model kombinasi ini berhasil meningkatkan akurasi hingga 96.1%, yang lebih tinggi dibandingkan dengan penggunaan masing-masing algoritma secara terpisah. Hasil ini menunjukkan bahwa kombinasi kedua algoritma dapat mengatasi kelemahan masing-masing model dan saling melengkapi untuk meningkatkan kinerja dalam klasifikasi penyakit diabetes. Oleh karena itu, penggunaan kombinasi *Gaussian Naïve Bayes* dan *Adaboost* memberikan kontribusi yang besar dalam meningkatkan performa klasifikasi, menjadikannya lebih efektif untuk mengidentifikasi kasus diabetes dengan akurasi yang lebih tinggi.

Penelitian ini dapat dikembangkan lebih lanjut dengan mengeksplorasi metode ensemble lain, seperti Random Forest atau XGBoost, serta menguji performa model pada dataset yang lebih kompleks dan beragam, baik dari segi ukuran, jumlah fitur, maupun tingkat ketidak seimbangan kelas. Penelitian juga dapat mencoba kombinasi *Adaboost* dengan algoritma dasar lain, seperti Decision Tree, untuk membandingkan hasilnya.

Selanjutnya, penelitian dapat diperluas implementasi praktis, disarankan untuk membuat sistem klasifikasi berbasis model yang dihasilkan, seperti aplikasi berbasis web atau perangkat lunak desktop, yang dapat digunakan langsung oleh pengguna untuk klasifikasi

otomatis pada data baru. Sistem ini dapat dirancang agar mudah diakses, user-friendly, dan efisien secara komputasi.

### Daftar Rujukan

- [1] WHO, "Diabetes," 2024. [https://www.who.int/health-topics/diabetes#tab=tab\\_3](https://www.who.int/health-topics/diabetes#tab=tab_3)
- [2] IDF, "International Diabetes Federation," 2024. <https://idf.org/our-network/regions-and-members/western-pacific/members/indonesia/>
- [3] Kemenkes RI, "Saatnya mengatur si Manis," 2024. <https://sehatnegeriku.kemkes.go.id/baca/blog/20240110/5344736/saatnya-mengatur-si-manis/>
- [4] S. T. Siridion and B. Siregar, "ANALISIS KLASIFIKASI DIAGNOSA PENYAKIT DIABETES MELITUS BERDASARKAN KOMPARASI ALGORITMA SUPERVISED LEARNING," *Multidisciplinary Sci. Journal*, vol. 2, no. 3, pp. 1006–1014, 2024, [Online]. Available: <https://doi.org/10.57185/mutiara.v2i2.159>
- [5] W. D. Septiana and U. Rohwadi, "Optimasi Algoritma Genetika Pada Algoritma C4.5 Untuk Deteksi Dini Penyakit Diabetes," *J. AKRAB JUARA*, vol. 6, no. 5, pp. 221–229, 2021.
- [6] W. Apriliah, I. Kurniawan, M. Baydhowi, and T. Haryati, "Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest," *Sistemasi*, vol. 10, no. 1, p. 163, 2021, doi: 10.32520/stmsi.v10i1.1129.
- [7] H. Mukaromah, "Komparasi Teknik Bagging Dan Adaboost Pada Decision Tree Dan Naive Bayes Untuk Prediksi Stroke," *JUPITER J. Penelit. Ilmu dan Teknol. ...*, pp. 167–180, 2024, [Online]. Available: <https://jurnal.polsri.ac.id/index.php/jupiter/article/view/8448%0Ahttps://jurnal.polsri.ac.id/index.php/jupiter/article/download/8448/3043>
- [8] L. Pebrianti, F. Aulia, H. Nisa, and K. Saputra, "Informasi Implementasi Metode Adaboost untuk Mengoptimasi Klasifikasi Penyakit Diabetes dengan Algoritma Naive Bayes," *J. Sist. dan Teknol.*, vol. 7, no. 2, pp. 122–127, 2022, [Online]. Available: <http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/8627%0Ahttp://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/download/8627/4296>
- [9] A. A. Karim, M. A. Prasetyo, and M. R. Saputra, "Perbandingan Metode Random Forest, K-Nearest Neighbor, dan SVM Dalam Prediksi Akurasi Pertandingan Liga Italia," *Pros. Semin. Nas. Teknol. dan Sains*, vol. 2, pp. 377–342, 2023, [Online]. Available: <http://www.football-data.co.uk>
- [10] M. Rizal, M. Z. Syahaf, S. R. Priyambodo, and Y. Rhamdani, "Optimasi Algoritma Naive Bayes Menggunakan Forward Selection Untuk Klasifikasi Penyakit Ginjal Kronis," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 5, no. 1, pp. 71–80, 2023, doi: 10.53580/naratif.v5i1.200.
- [11] A. Ridwan, "Penerapan Algoritma Naive Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, 2020, doi: 10.47970/siskom-kb.v4i1.169.
- [12] T. Tanti, P. Sirait, and A. Andri, "Optimalisasi Kinerja Klasifikasi Melalui Seleksi Fitur dan AdaBoost dalam Penanganan Ketidakseimbangan Kelas," *J. Media Inform. Budidarma*, vol. 5, no. 4, p. 1377, 2021, doi: 10.30865/mib.v5i4.3280.
- [13] U. Indahyanti, N. L. Azizah, and H. Setiawan, "Pendekatan Ensemble Learning Untuk Meningkatkan Akurasi Prediksi Kinerja Akademik Mahasiswa," *J. Sains dan Inform.*, vol. 8, no. 2, pp. 160–169, 2022, doi: 10.34128/jsi.v8i2.459.
- [14] A. Nur Rais and W. Warjiyono, "Optimasi Akurasi Klasifikasi Pada Prediksi Smoke Detection dengan Menggunakan Algoritma Adaboost," *J. Sist. Komput. dan Inform.*, vol. 4, no. 2, p. 343, 2022, doi: 10.30865/json.v4i2.5154.
- [15] M. Rama, H. Suryanto, and D. W. Utomo, "Pembelajaran Ensemble Untuk Klasifikasi Ulasan Pelanggan E-commerce Menggunakan Teknik Boosting," vol. 15, no. 02, pp. 238–244, 2024, doi: 10.35970/infotekmesin.v15i2.2314.