Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

Perbandingan Kinerja Metode Naïve Bayes dan Random Forest untuk Klasifikasi Penyakit Diabetes Berdasarkan Data Medis

Rendy Risqi Pradana, Yani Parti Astuti*

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia
Email: [1]111202113345@mhs.dinus.ac.id, [2,*]yanipartiastuti@dns.dinus.ac.id
Email Penulis Korespondensi: yanipartiastuti@dns.dinus.ac.id
Submitted: **27/05/2025**; Accepted: **15/06/2025**; Published**: 15/06/2025**

Abstract−Diabetes mellitus is a non-communicable disease with a prevalence that continues to rise in Indonesia. Conventional diagnostic processes often face challenges, including delays and high costs. This study aims to compare the performance of the Naive Bayes and Random Forest algorithms in classifying diabetes using the Pima Indians Diabetes Dataset. To address class imbalance, the dataset was processed using the Synthetic Minority Over-sampling Technique (SMOTE). Performance evaluation was conducted using accuracy, precision, recall, and F1-score metrics. The results showed that the Random Forest algorithm achieved an accuracy of 79.5%, precision of 79.6%, recall of 79.5%, and F1-score of 79.5%. In contrast, the Naive Bayes algorithm achieved an accuracy of 76.5%, precision of 76.5%, recall of 76.5, and F1-score of 76.5%. These findings indicate that while Random Forest is superior in handling complex data with higher predictive accuracy, Naive Bayes remains effective for simpler implementations due to its computational efficiency. This study contributes to the development of intelligent decision support systems for earlier and more accurate diabetes detection, potentially reducing the burden on healthcare systems.

Keywords: Diabetes Mellitus; Naive Bayes; Random Forest; SMOTE; Decision Support System.

Abstract−Diabetes mellitus merupakan penyakit tidak menular yang prevalensinya terus meningkat di Indonesia. Proses diagnosis secara konvensional sering menghadapi berbagai tantangan, seperti keterlambatan dan biaya yang tinggi. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Naive Bayes dan Random Forest dalam klasifikasi diabetes dengan menggunakan dataset Pima Indians Diabetes. Untuk mengatasi ketidakseimbangan kelas, dataset diproses menggunakan teknik Synthetic Minority Over-sampling Technique (SMOTE). Evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa algoritma Random Forest memperoleh akurasi sebesar 79,5%, presisi 79,6%, recall 79,5%, dan F1-score 79,5%. Sementara itu, algoritma Naive Bayes memperoleh akurasi 76,5%, presisi 76,5%, recall 76,5%, dan F1-score 76,5%. Temuan ini menunjukkan bahwa Random Forest unggul dalam menangani data yang kompleks dengan akurasi prediksi yang lebih tinggi, sedangkan Naive Bayes tetap efektif untuk implementasi yang lebih sederhana karena efisiensi komputasinya. Studi ini berkontribusi dalam pengembangan sistem pendukung keputusan cerdas untuk deteksi dini diabetes yang lebih cepat dan akurat, sehingga dapat membantu mengurangi beban pada sistem layanan kesehatan.

Keywords: Diabetes Mellitus; Naive Bayes; Random Forest; SMOTE; Decision Support System.

1. PENDAHULUAN

Diabetes melitus merupakan penyakit tidak menular yang prevalensinya terus meningkat di Indonesia, memberikan dampak signifikan terhadap kualitas hidup dan beban sistem kesehatan nasional[1]. Penyakit ini dapat menyebabkan komplikasi serius seperti penyakit jantung, stroke, gagal ginjal, dan kerusakan saraf, sehingga deteksi dini dan pengelolaan yang tepat menjadi [2], [3]. Data Badan Pusat Statistik (BPS) menunjukkan peningkatan prevalensi diabetes dari 279.345 kasus pada tahun 2021 menjadi 435.512 pada tahun 2022, dan diperkirakan mencapai 605.570 kasus pada tahun 2023 [1] Faktor risiko seperti obesitas, gaya hidup sedentari, dan pola makan tidak sehat turut berkontribusi terhadap tren ini, terutama di kelompok usia 45 tahun ke atas dan wilayah perkotaan [2], [4].

Proses diagnosis tradisional diabetes, seperti pemeriksaan glukosa darah, seringkali menghadapi tantangan seperti keterlambatan dan biaya tinggi [5]. Oleh karena itu, pendekatan alternatif berbasis kecerdasan buatan (AI) dan pembelajaran mesin menjadi solusi yang menjanjikan untuk meningkatkan akurasi dan kecepatan deteksi dini[6], [7]. Dua algoritma yang umum digunakan adalah Random Forest, yang mengombinasikan pohon keputusan untuk meningkatkan akurasi prediksi , dan Naive Bayes, yang menggunakan pendekatan probabilistik dengan asumsi independensi fitur . Kedua algoritma ini telah terbukti efektif dalam mengelola data medis yang kompleks [8].

Selain itu, dalam mengatasi masalah ketidakseimbangan kelas pada dataset, yang merupakan tantangan umum dalam analisis medis, teknik SMOTE (Synthetic Minority Over-sampling Technique) dapat digunakan untuk meningkatkan kinerja model. SMOTE menghasilkan contoh sintetis dari kelas yang kurang terwakili, membantu model untuk mempelajari fitur-fitur dari kedua kelas secara lebih seimbang [9]. Hal ini sangat penting untuk mencegah model lebih fokus pada kelas mayoritas dan mengabaikan kelas minoritas, seperti pasien yang terdiagnosis diabetes. Beberapa studi telah menunjukkan bahwa SMOTE dapat meningkatkan kinerja model prediksi diabetes secara signifikan, terutama dalam meningkatkan recall dan sensitivity pada dataset yang tidak seimbang [10], [11].

Klasifikasi diabetes menggunakan pembelajaran mesin melibatkan pemrosesan data medis yang mencakup berbagai parameter, seperti kadar glukosa darah, tekanan darah, BMI, serta faktor risiko lainnya. Algoritma seperti Naive Bayes dan Random Forest digunakan untuk memprediksi kemungkinan seseorang menderita diabetes berdasarkan data medis tersebut. Penggunaan algoritma ini dapat membantu dalam diagnosis dini, yang sangat penting untuk mencegah komplikasi serius dan mengurangi angka kesalahan diagnosis yang sering terjadi dalam metode tradisional [6].

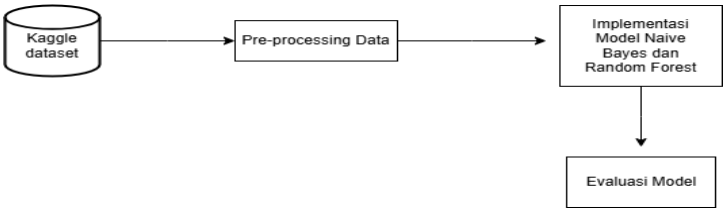
Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

Penelitian sejenis telah dilakukan oleh beberapa peneliti dengan fokus pada penggunaan algoritma klasifikasi dalam prediksi diabetes. Hasanah dan Munggaran (2021) membandingkan algoritma Naive Bayes dan Random Forest menggunakan dataset publik dari Kaggle. Hasil penelitian mereka menunjukkan bahwa Random Forest menghasilkan akurasi yang lebih tinggi dibandingkan Naive Bayes, terutama pada pembagian data 70:30 dan 90:10 [12].

Sementara itu, Huda dan Ula (2024) membandingkan lima algoritma klasifikasi yaitu Naive Bayes, Logistic Regression, Random Forest, SVM, dan KNN. Berdasarkan hasil pengujian, Random Forest mencatat akurasi tertinggi, yaitu sebesar 81%, sehingga dinilai paling efektif dalam memprediksi diabetes mellitus [13].

Berdasarkan penelitian-penelitian terdahulu tersebut, tujuan dari penelitian ini adalah untuk membandingkan kinerja Random Forest dan Naive Bayes dalam klasifikasi diabetes menggunakan dataset publik dari Kaggle, dengan fokus pada parameter medis seperti kadar glukosa darah, tekanan darah, dan indeks massa tubuh (BMI) [14]. Hasil penelitian diharapkan dapat memberikan wawasan tentang algoritma yang lebih efektif dan efisien, serta berkontribusi pada pengembangan sistem pendukung keputusan di bidang kesehatan [15]. Dengan memanfaatkan data terbaru dan teknologi pembelajaran mesin, penelitian ini diharapkan dapat meningkatkan deteksi dini dan pengelolaan diabetes, mengurangi beban ekonomi dan sosial, serta memperluas akses layanan kesehatan di Indonesia [2], [4].

2. METODOLOGI PENELITIAN



Gambar 1. Alur Program

Penelitian ini memanfaatkan data medis seperti kadar glukosa, tekanan darah, BMI, dan faktor risiko lainnya untuk mengklasifikasikan status diabetes. Pada Gambar 1, alur program dimulai dengan pengambilan dataset dari Kaggle yang berisi data pasien. Selanjutnya, dilakukan pre-processing data yang meliputi pembersihan, penanganan nilai yang hilang, dan normalisasi agar data siap digunakan. Setelah itu, diimplementasikan dua algoritma yaitu Naive Bayes dan Random Forest untuk melakukan klasifikasi. Tahapan terakhir adalah evaluasi model menggunakan metrik akurasi, precision, recall, dan F1-score guna membandingkan efektivitas kedua algoritma.

2.1 Pengumpulan Data

Penelitian ini menggunakan Pima Indians Diabetes Database (PIDD) yang diperoleh dari platform Kaggle yang Bisa di diakses melalui link berikut ini : https://www.kaggle.com/code/shrutimechlearn/step-by-step-diabetes-classification. Dataset ini memiliki beberapa keterbatasan, di antaranya bahwa seluruh individu yang terdaftar dalam dataset ini adalah perempuan berusia minimal 21 tahun dengan latar belakang etnis Pima Indian. Dataset ini mencakup berbagai fitur medis, antara lain Pregnancies (jumlah kehamilan), Glucose (kadar glukosa darah), BloodPressure (tekanan darah), SkinThickness (ketebalan kulit), Insulin (kadar insulin darah), BodyMassIndex (BMI), DiabetesPedigreeFunction (riwayat keluarga diabetes), Age (usia), dan Outcome (status diabetes: 0 untuk tidak terinfeksi, 1 untuk terinfeksi). Fitur-fitur tersebut digunakan untuk memprediksi kemungkinan seseorang terkena diabetes berdasarkan kondisi medis mereka.

2.2 Preprocessing Data

Preprocessing data adalah tahap yang sangat penting dalam proses machine learning untuk memastikan bahwa data yang digunakan dalam model adalah bersih, terstruktur dengan baik, dan siap digunakan. Tahapan preprocessing ini meliputi beberapa langkah penting yang dirancang untuk mengatasi masalah-masalah seperti distribusi data yang tidak merata, nilai yang hilang, duplikasi data, dan deteksi outlier. Dalam penelitian ini, proses preprocessing dilakukan melalui serangkaian tahapan yang dijelaskan sebagai berikut:

a. Eksplorasi Data (Exploratory Data Analysis - EDA)
   Exploratory Data Analysis (EDA) merupakan langkah pertama untuk memahami karakteristik data. Dalam tahap ini, dilakukan analisis untuk memeriksa distribusi data, identifikasi nilai yang hilang, serta deteksi outlier. Statistik deskriptif digunakan untuk memberikan gambaran umum mengenai distribusi data dalam dataset. Fungsi df.describe() digunakan untuk menghitung statistik dasar, seperti mean, standar deviasi, nilai minimum, nilai maksimum, dan kuartil. Rumus untuk menghitung mean dan standar deviasi adalah sebagai berikut:
   1. Mean

$$mean = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

   Dimana data x_i adalahnilai data dan n adalah jumlah data.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

2. Standar Deviasi

$$Standar\ Deviation = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \quad (2)$$

Dimana μ adalah nilai rata-rata dari data.
Untuk memeriksa distribusi setiap fitur, dilakukan visualisasi menggunakan histogram dan Kernel Density Estimation (KDE) yang dapat dihasilkan dengan sns.histplot(). Langkah ini bertujuan untuk memahami apakah data terdistribusi normal atau memiliki distribusi yang sangat miring (skewed). Visualisasi distribusi penting untuk mengetahui apakah data perlu diproses lebih lanjut (misalnya, transformasi logaritmik untuk mengurangi skewness), Outlier dapat sangat memengaruhi hasil analisis dan model machine learning. Dalam penelitian ini, outlier dideteksi menggunakan boxplot dan Interquartile Range (IQR). Rumus untuk menghitung IQR dan mendeteksi outlier adalah sebagai berikut:
IQR dihitung dengan:

$$IQR = Q3 - \mathrm{Q1} \quad (3)$$

Dimana Q1 adalah quartil pertama dan Q3 quartil ketiga.
Outlier dapat dideteksi dengan rumus.

$$Lower\ Bound = Q1 - 1.5 \times IQR$$
$$Upper\ Bound = Q3 + 1.5 \times IQR \quad (4)$$

Data yang berada di luar batas bawah atau batas atas ini dianggap sebagai outlier, dan langkah pembersihan data dilakukan untuk mengatasi nilai-nilai ekstrem tersebut.

b. Validasi Data
Validasi data adalah langkah untuk memastikan bahwa data yang digunakan dalam model machine learning memiliki kualitas yang baik. Beberapa hal yang diperiksa selama validasi data adalah :
1. Tipe Data: Memeriksa tipe data untuk setiap fitur dalam dataset dengan menggunakan df.info(). Hal ini untuk memastikan bahwa kolom-kolom numerik dan kategorikal telah teridentifikasi dengan benar.
2. Missing Value: Nilai yang hilang dihitung dengan df.isnull().sum(). Jika ditemukan nilai yang hilang, langkah penanganan missing value dilakukan, baik dengan cara imputasi atau penghapusan baris yang memiliki nilai hilang.
3. Data Duplikat: Pengecekan data duplikat dilakukan dengan menggunakan df.duplicated().sum(). Jika ditemukan data duplikat, baris tersebut akan dihapus untuk menjaga keakuratan model.

c. Penanaganan Missing Value
Pada dataset ini, tidak ditemukan missing value atau data duplikat, sehingga langkah ini tidak diperlukan. Namun, jika ditemukan missing value, penanganan dilakukan dengan dua cara:
1. Imputasi: Nilai yang hilang digantikan dengan nilai yang paling sering muncul (mode) untuk data kategorikal, atau rata-rata (mean) untuk data numerik.
2. Penghapusan Baris: Baris yang mengandung nilai hilang dapat dihapus jika proporsi data yang hilang cukup besar untuk memengaruhi analisis.

Untuk outlier, nilai-nilai yang melebihi batas yang dihitung dengan menggunakan IQR akan diganti dengan nilai batas bawah atau atas. Ini untuk memastikan bahwa model tidak terpengaruh oleh data yang sangat ekstrem.

d. Feature Selection dan Standarisasi
Proses pemilihan fitur (feature selection) dilakukan untuk memilih fitur yang paling relevan dalam memprediksi target. Dalam penelitian ini, fitur dipilih menggunakan teknik Mutual Information, yang mengukur ketergantungan antara fitur dan target. Rumus untuk menghitung Mutual Information adalah sebagai berikut.

$$\mathrm{I(X,Y)} = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \quad (5)$$

Dimana p(x,y) adalah probabilitas bersama dan p(x), p(y) adalah probabilitas marginal dari fitur X dan target Y.
Setelah fitur yang relevan dipilih, data distandarisasi menggunakan StandardScaler untuk memastikan bahwa setiap fitur memiliki rata-rata 0 dan standar deviasi 1. Hal ini penting untuk menghindari bias pada model yang sensitif terhadap skala fitur. Rumus untuk standarisasi adalah sebagai berikut.

$$z = \frac{x-\mu}{\sigma} \quad (6)$$

Dimana x adalah nilai fitur, μ adalah rata-rata fitur, σ adalahcstandar deviasi fitur.
Proses standarisasi ini memastikan bahwa setiap fitur memiliki skala yang seragam, sehingga model seperti Naive Bayes dan Random Forest dapat beroperasi lebih efisien tanpa dipengaruhi oleh skala fitur yang berbeda.

e. SMOTE (Synthetic Minority Over-sampling Technique)
Pada dataset ini, SMOTE digunakan untuk menangani ketidakseimbangan kelas pada variabel target Outcome (0 = tidak diabetes, 1 = diabetes). Dataset ini mengalami ketidakseimbangan kelas, dimana jumlah pasien yang tidak terkena diabetes lebih banyak dibandingkan dengan pasien yang terkena diabetes. SMOTE adalah teknik yang

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

digunakan untuk menyeimbangkan kelas dengan cara menghasilkan contoh sintetis dari kelas yang minoritas (dalam hal ini, pasien yang terkena diabetes. Dengan menggunakan SMOTE, jumlah pasien yang terdiagnosis diabetes akan meningkat, sehingga model akan mendapatkan representasi yang lebih baik dari kelas minoritas dan dapat mengklasifikasikan kedua kelas dengan lebih akurat.

2.3 Naive Bayes dan Random Forest

Pada tahap ini, dilakukan eksperimen dengan berbagai konfigurasi model untuk membandingkan performa klasifikasi menggunakan algoritma Naïve Bayes dan Random Forest. Eksperimen ini mencakup dua kondisi utama, yaitu model yang dikembangkan tanpa penerapan metode penyeimbangan data dan model yang menggunakan metode SMOTE untuk menangani ketidakseimbangan kelas pada dataset. Kedua metode Naïve Bayes dan Random Forest akan dijelaskan sebagai berikut.

a. Naïve Bayes

Naive Bayes merupakan algoritma klasifikasi yang didasarkan pada prinsip teorema Bayes, dengan anggapan bahwa setiap fitur dalam dataset memiliki sifat independen satu sama lain, meskipun asumsi independensi ini jarang terjadi dalam praktik. Walaupun asumsi independensi ini jarang sesuai dengan data di dunia nyata, algoritma ini tetap mampu memberikan hasil yang efektif dalam berbagai aplikasi, termasuk dalam klasifikasi penyakit. Secara matematis, Naive Bayes mengasumsikan bahwa untuk setiap kelas C, probabilitas untuk mendapatkan nilai fitur x1,x2,...,xn dapat dihitung menggunakan rumus Bayes sebagai berikut:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (7)$$

Dimana P(C|X) adalah probabilitas kelas C diberikan fitur X, P(X|C) adalah likelihood, probabilitas data X diberikan kelas C, P(C) adalah prior probability dari kelas C,P(X) adalah probabilitas total data X[16].

Asumsi independensi dalam Naive Bayes menyatakan bahwa setiap fitur xi bersifat independen satu sama lain, sehingga likelihood P(X|C) dapat dihitung sebagai produk dari probabilitas untuk masing-masing fitur:

$$P(X|C) = \prod_{i=1}^{n} P(x_i|C) \quad (8)$$

Dengan demikian, probabilitas akhir untuk setiap kelas C dapat dihitung, dan kelas dengan probabilitas tertinggi akan dipilih sebagai prediksi[17][18].

b. Random Forest

Random Forest adalah algoritma ensemble yang terdiri dari sekumpulan pohon keputusan (decision trees). Setiap pohon keputusan dilatih menggunakan subset acak dari data dan fitur, dan hasil prediksi dari semua pohon keputusan kemudian digabungkan untuk menghasilkan prediksi akhir. Random Forest menggunakan prinsip bagging (bootstrap aggregating), yang mengurangi variansi model dengan menggabungkan banyak model sederhana untuk membentuk satu model yang lebih kuat[19].

Random Forest membentuk pohon keputusan melalui tiga langkah utama: (1) mengambil subset data secara acak dengan teknik bootstrap, (2) memilih subset acak fitur pada setiap node untuk mengurangi korelasi antar pohon, dan (3) menghasilkan prediksi akhir berdasarkan voting mayoritas dari seluruh pohon dalam model[19], [20]. Secara matematis, jika N adalah jumlah pohon dalam Random Forest, maka prediksi untuk kelas C diberikan oleh mayoritas suara (voting) dari seluruh pohon Keputusan:

$$\hat{C} = arg\ \max_{c} + \sum_{i=1}^{N} I(C_i = c) \quad (9)$$

Dimana Ci adalah prediksi dari pohon keputusan ke-I,I(Ci=c) adalah indikator yang bernilai 1 jika prediksi pohon ke-i adalah kelas c, dan 0 jika bukan.

C adalah kelas hasil prediksi mayoritas[19]om Forest adalah kemampuannya dalam menangani data besar dan kompleks, serta mengurangi risiko overfitting yang sering terjadi pada pohon keputusan tunggal. Selain itu, Random Forest juga dapat memberikan informasi tentang pentingnya fitur dalam prediksi, yang sangat berguna dalam interpretasi model[20]. Namun, kekurangan Random Forest adalah konsumsi memori yang tinggi dan waktu pelatihan yang lebih lama dibandingkan dengan algoritma yang lebih sederhana seperti Naive Bayes[19] .

2.4 Evaluasi Model

Pada tahap evaluasi, penelitian ini menggunakan confusion matrix. Confusion matrix menunjukkan berapa banyak prediksi yang benar (*True Positive* dan *True Negative*) dan berapa banyak prediksi yang salah (*False Positive* dan *False Negative*).

Tabel 1. Confusion Matrix

	Prediksi Positif (1)	Prediksi Negatif (0)
Aktual Positif (1)	True Positive (TP)	False Negative (FN)
Aktual Negatif (0)	False Positive (FP)	True Negative (TN)

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

Confusion Matrix adalah tabel yang menilai performa model klasifikasi dengan membandingkan prediksi dan label aktual. Tabel 1 membagi hasil ke dalam empat kategori: True Positive (TP) dan True Negative (TN) untuk prediksi yang benar, serta False Positive (FP) dan False Negative (FN) untuk prediksi yang salah. Dari Confusion Matrix, dapat dihitung metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk menilai kinerja model secara menyeluruh. Dari Confusion Matrix, kita dapat menghitung beberapa metrik evaluasi untuk memahami performa model. Berikut adalah metrik utama yang sering digunakan :
Accuracy (Akurasi): Mengukur seberapa banyak prediksi yang benar dari total prediksi.

$$Accuracy = \frac{TP + TN}{TP+TN + FP +FN} \tag{10}$$

Precision (Presisi): Mengukur proporsi prediksi positif yang benar.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

Recall (Sensitivitas atau True Positive Rate): Mengukur seberapa baik model mendeteksi kelas positif.

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

F1-Score: Rata-rata harmonis antara presisi dan recall, digunakan untuk dataset yang tidak seimbang.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

Confusion Matrix sangat berguna untuk memahami performa model pada tingkat granular. Terutama dalam kasus dataset yang tidak seimbang, metrik seperti *Precision* dan *Recall* dapat memberikan wawasan yang lebih baik dibandingkan hanya melihat *Accuracy*. Dengan memanfaatkan Confusion Matrix, kita dapat mengevaluasi model sesuai dengan kebutuhan spesifik, seperti memprioritaskan deteksi positif (*Recall*) atau menghindari prediksi positif palsu (*Precision*).

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil yang diperoleh dari setiap tahapan yang telah dijelaskan sebelumnya dalam bagian metode penelitian. Setiap langkah yang diuraikan dalam metode penelitian memberikan kontribusi yang signifikan terhadap pencapaian hasil-hasil tersebut. Penjelasan mendalam mengenai hasil yang diperoleh pada setiap tahap ini akan memberikan gambaran yang jelas tentang bagaimana proses penelitian dijalankan dan bagaimana data yang terkumpul dapat memberikan pemahaman yang lebih baik terhadap topik yang diteliti. Berikut adalah uraian lebih rinci mengenai hasil yang diperoleh dari masing-masing tahapan tersebut.

3.1 Pengumpulan Data

Data yang saya gunakan dalam penelitian ini diperoleh dari *Pima Indians Diabetes Database*, yang disediakan oleh *National Institute of Diabetes and Digestive* and *Kidney Diseases*, dan dapat diakses melalui *platform Kaggle*. Rincian data yang digunakan dapat dilihat pada Tabel 2.

Tabel 2 Cuplikan Dataset

No	Preg	Glu	BP	Skin	Ins	BMI	DPF	Age	Out
1	6	148	72	35	0	33,6	0,627	50	1
2	1	85	66	29	0	26,6	0,351	31	0
3	8	183	64	0	0	23,3	0,672	32	1
4	1	89	66	23	94	28,1	0,167	21	0
5	0	137	40	35	168	43,1	2,288	33	1
…	…	…	…	…	…	…	…	…	…
764	10	101	76	48	180	32,9	0,171	63	0
765	2	122	70	27	0	36,8	0,340	27	0
766	5	121	72	23	112	26,2	0,245	30	0
767	1	126	60	0	0	30,1	0,49	47	1
768	1	93	70	31	0	30,4	0,315	23	0

Dataset Pima Indians Diabetes terdiri dari 768 baris data dan 9 kolom. Dataset ini mencakup 8 kolom yang berfungsi sebagai fitur, yaitu: kolom *Pregnancies* yang menunjukkan jumlah kehamilan, *Glucose* yang mencatat kadar glukosa dalam darah, *BloodPressure* yang mengukur tekanan darah, *SkinThickness* yang mengindikasikan ketebalan kulit, *Insulin* yang merujuk pada kadar insulin dalam tubuh, *BodyMassIndex* (BMI) yang menghitung indeks massa tubuh, *DiabetesPedigreeFunction* (DPF) yang menunjukkan riwayat keluarga terkait diabetes, dan *Age* yang mencatat usia individu. Selain itu, terdapat 1 kolom target bernama *Outcome* yang menunjukkan hasil diagnosis apakah

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

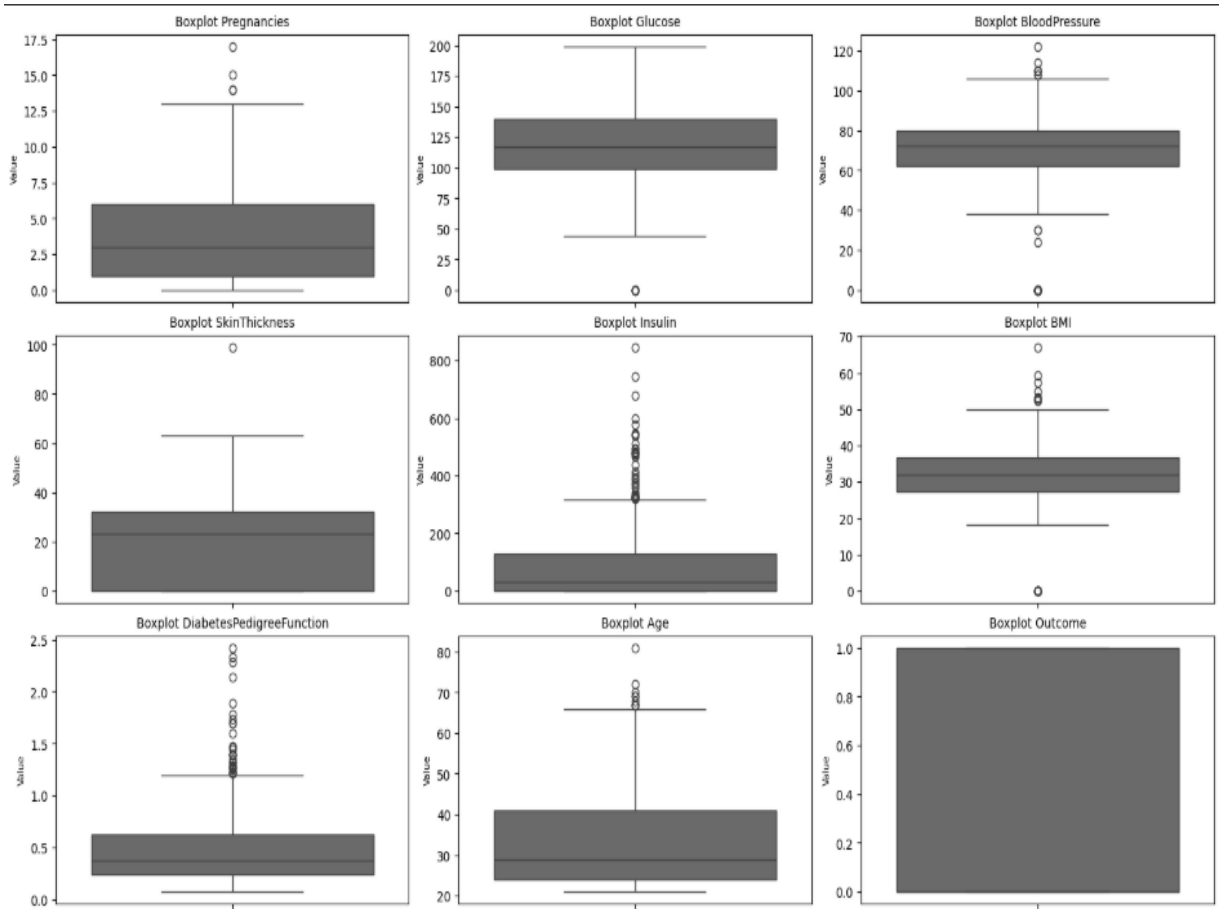
seseorang terdiagnosis diabetes (1) atau tidak (0). Penjelasan lebih rinci mengenai setiap kolom dalam dataset ini dapat ditemukan pada Tabel 3.

Tabel 3. Deskripsi Atribut Dataset

Atribut	Keterangan/Deskripsi
Preg	Jumlah Kehamilan
Glu	Kadar glukosa plasma setelah tes toleransi glukosa 2 jam (mg/dl)
BP	Tekanan darah diastolic (mm Hg)
Skin	Ketebalan lipatan kulit triceps (mm)
Ins	Kadar insulin serum setelah toleransi glukosa 2 jam (mu U/ml)
BMI	Indeks masa tubuh (kg/$\boldsymbol{m^2}$)
DPF	Riwayat diabetes dalam keluarga
Age	Usia
Out	Positif diabetes (1) dan negative diabetes (0)

3.2 Preprocessing Data

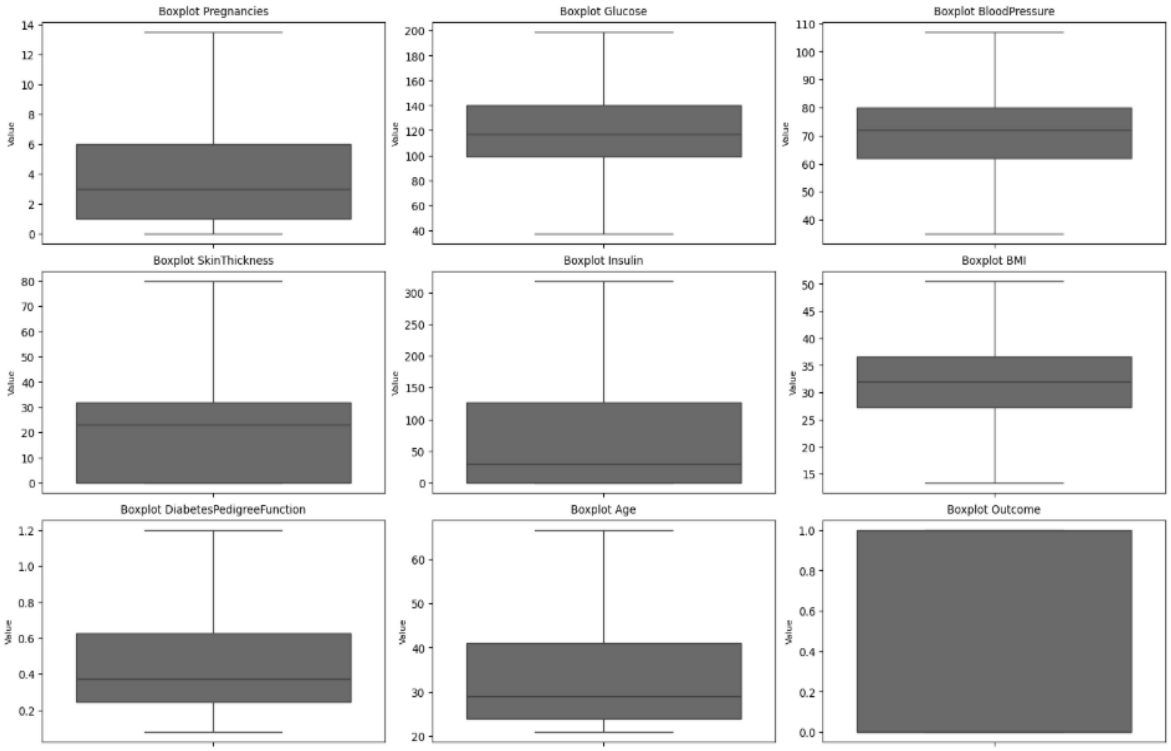
Pada tahap awal, dilakukan pemeriksaan konsistensi dan kelengkapan dataset yang terdiri dari beberapa fitur medis seperti Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, dan Outcome. Analisis outlier menggunakan boxplot (Gambar 2) menunjukkan bahwa kolom Insulin memiliki banyak nilai ekstrem, sedangkan Glucose dan BMI juga mengandung outlier meskipun lebih sedikit.



Gambar 2. Hasil Outlier

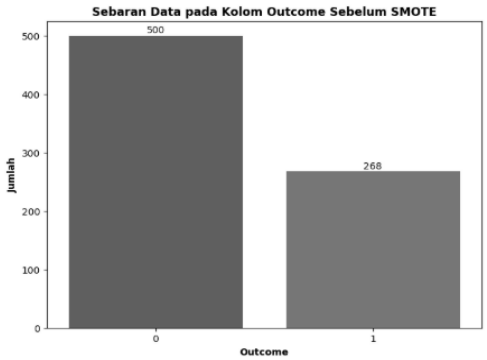
Untuk menangani outlier tersebut, dilakukan penggantian nilai yang berada di luar batas normal menggunakan Interquartile Range (IQR), yang membatasi nilai-nilai ekstrim dan menggantikannya dengan batas bawah atau batas atas yang wajar. Selain itu, setelah outlier ditangani, data kemudian distandarisasi menggunakan StandardScaler. Langkah ini penting karena algoritma pembelajaran mesin seperti Naive Bayes dan Random Forest sangat sensitif terhadap skala data. Hasil dari standarisasi data ini dapat dilihat pada Gambar 3, yang menunjukkan bahwa nilai-nilai pada kolom seperti Glucose, BMI, dan Age kini berada dalam rentang yang lebih seragam, memudahkan proses pelatihan model.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446



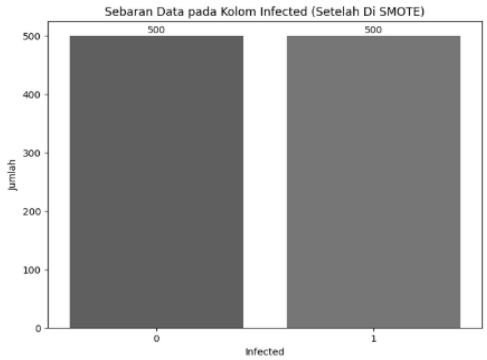
Gambar 3. Hasil Penanganan Outlier

Tahap berikutnya adalah penanganan ketidakseimbangan kelas. Distribusi kelas sebelum penerapan SMOTE (Synthetic Minority Over-sampling Technique) menunjukkan bahwa kelas 0 (tidak diabetes) jauh lebih dominan dibandingkan kelas 1 (diabetes). Gambar 4 menunjukkan distribusi data sebelum SMOTE, dengan kelas 0 memiliki 500 sampel, sementara kelas 1 hanya memiliki 268 sampel.



Gambar 4. Data Sebelum Smote

Setelah penerapan SMOTE, distribusi kelas menjadi seimbang, dengan jumlah sampel kelas 0 dan kelas 1 masing-masing sebanyak 500. Gambar 5 menunjukkan distribusi data setelah SMOTE, di mana kedua kelas kini memiliki jumlah yang setara, yang diharapkan dapat meningkatkan kinerja model.



Gambar 5. Data Setelah Smote

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

Selain itu, untuk memilih fitur yang paling relevan dengan target, digunakan teknik Mutual Information, yang mengukur ketergantungan antara setiap fitur dengan target (Outcome). Berdasarkan hasil pemilihan fitur, Tabel 4 menunjukkan bahwa fitur Glucose dan BMI memiliki skor tertinggi, yang berarti keduanya memiliki hubungan yang lebih kuat dengan target. Sebaliknya, fitur DiabetesPedigreeFunction memiliki skor yang sangat rendah, sehingga fitur ini kurang relevan dan dapat diabaikan dalam pemodelan lebih lanjut.

Tabel 4. Nilai Standardscaler

Feature	Score
Glu	0,111596
BMI	0,065587
Age	0,063141
Ins	0,034483
Preg	0,009835
BP	0,008388
Skin	0,008080
DPF	0,000000

Langkah-langkah preprocessing ini, yang meliputi penanganan outlier, normalisasi, penyeimbangan kelas dengan SMOTE, serta pemilihan fitur, memberikan fondasi yang kuat untuk tahap pemodelan lebih lanjut. Dengan data yang telah dibersihkan dan dipersiapkan dengan baik, model pembelajaran mesin dapat dilatih lebih efisien dan memberikan hasil yang lebih akurat

3.3 Implementasi Model

Dalam penelitian ini, model yang digunakan untuk mengklasifikasikan pasien yang menderita diabetes (kelas 1) dan yang tidak menderita diabetes (kelas 0) adalah Naive Bayes dan Random Forest. Kedua model ini dipilih karena keduanya telah terbukti memiliki performa yang baik dalam masalah klasifikasi, terutama dalam menangani dataset dengan banyak fitur seperti pada dataset diabetes ini. Naive Bayes dipilih karena kemampuannya untuk bekerja dengan baik pada dataset yang memiliki fitur-fitur yang bersifat independen satu sama lain. Sementara itu, Random Forest dipilih karena kemampuannya untuk menangani data dengan banyak fitur serta mengurangi risiko overfitting melalui penggunaan ensemble learning, yang menggabungkan hasil dari banyak pohon Keputusan.

Penelitian ini menggunakan dua model klasifikasi, yaitu Gaussian Naive Bayes dan Random Forest, untuk mengklasifikasikan pasien yang menderita diabetes (kelas 1) dan tidak menderita diabetes (kelas 0). Gaussian Naive Bayes dipilih karena kemampuannya menangani fitur independen dan sederhana dalam implementasi, dengan parameter default seperti priors=None dan var_smoothing=1e-9.

Sementara itu, Random Forest digunakan karena keunggulannya dalam mengelola dataset berdimensi tinggi dan mengurangi overfitting melalui teknik ensemble learning. Model ini diimplementasikan menggunakan RandomForestClassifier dari pustaka scikit-learn, dengan parameter utama n_estimators=100, random_state=42, dan class_weight='balanced' untuk menangani ketidakseimbangan kelas. Untuk meningkatkan kinerja model terhadap data yang tidak seimbang, diterapkan metode SMOTE (Synthetic Minority Over-sampling Technique). SMOTE menyeimbangkan distribusi kelas dengan menghasilkan sampel sintetis untuk kelas minoritas, sehingga memungkinkan model belajar lebih baik dalam mengenali pola pada pasien diabetes. Bisa dilihat pada Tabel 5.

Tabel 5. Parameter Model

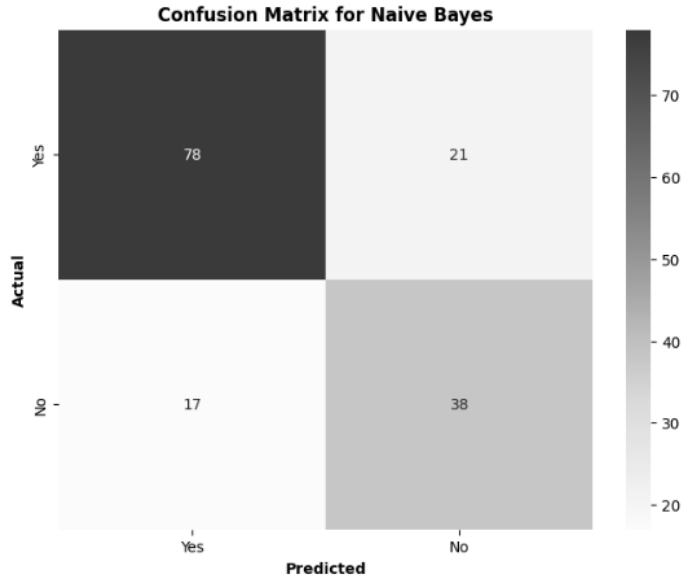
Model	Parameter	Nilai
Naïve Bayes	priors	None
	var_smoothing	1e-9
Random Forest	n_estimators	100
	random_state	42
	class_weight	balanced

3.4 Evaluasi Model

Pada tahap evaluasi, model yang diterapkan untuk mengklasifikasikan pasien diabetes adalah Naive Bayes dan Random Forest. Evaluasi dilakukan menggunakan Confusion Matrix untuk melihat bagaimana baiknya model dalam memprediksi kelas 0 (tidak diabetes) dan kelas 1 (diabetes) sebelum dan setelah penerapan SMOTE (Synthetic Minority Over-sampling Technique). SMOTE diterapkan untuk menangani masalah ketidakseimbangan kelas, di mana jumlah kelas 0 jauh lebih banyak daripada kelas 1.

Sebelum SMOTE diterapkan, dataset menunjukkan ketidakseimbangan kelas yang signifikan, di mana kelas mayoritas (kelas 0) mendominasi. Berdasarkan Confusion Matrix untuk Naive Bayes dan Random Forest, hasil evaluasi sebelum SMOTE dapat dilihat pada gambar berikut:

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446



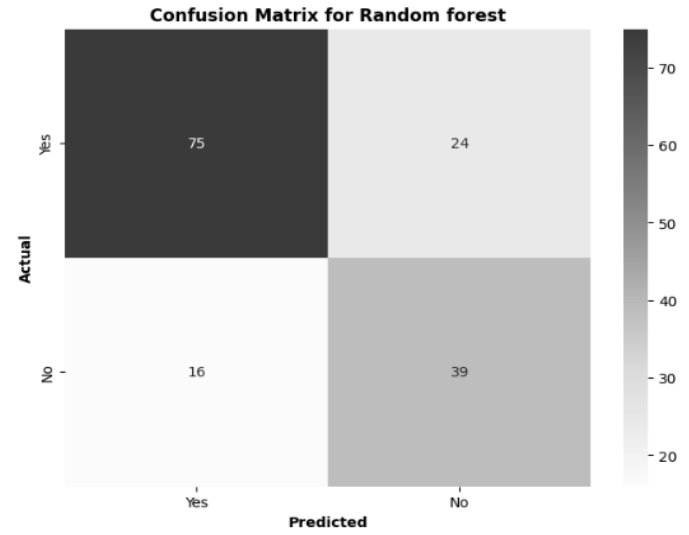
Gambar 6. Confusion Matrix Naïve Bayes sebelum Smote

Dari Gambar 6 bisa dilihat Confusion matrix untuk model Naive Bayes sebelum SMOTE menunjukkan bahwa ada 78 True Positives (TP), yaitu data positif yang diprediksi dengan benar sebagai positif. Sebanyak 21 False Positives (FP), yaitu data negatif yang salah diprediksi sebagai positif, dan 17 False Negatives (FN), yaitu data positif yang salah diprediksi sebagai negatif. Terakhir, ada 38 True Negatives (TN), yaitu data negatif yang diprediksi dengan benar sebagai negatif. Hasil ini menggambarkan kinerja model dalam mendeteksi data positif dan negatif.

Tabel 6. Hasil Evaluasi Naïve Bayes

Parameter	Nilai
Accuracy	75.3%
Precision	75.8%
Recall	75.3%
F1-Score	75.5%

Pada Tabel 6 model ini memiliki Accuracy sebesar 75.3%, yang menunjukkan bahwa 75.3% dari seluruh data berhasil diprediksi dengan benar. Precision sebesar 75.8% mengindikasikan bahwa 75.8% dari prediksi positif yang dilakukan oleh model adalah benar. Recall mencapai 75.3%, yang berarti model berhasil mendeteksi 75.3% dari kasus positif yang sebenarnya. Terakhir, F1-Score sebesar 75.5% menunjukkan keseimbangan antara precision dan recall dalam model, mengindikasikan kinerja yang seimbang dalam memprediksi dan mendeteksi data positif.



Gambar 7. Confusion Matrix Random Forest sebelum Smote

Pada Gambar 7, Confusion matrix untuk model Random Forest menunjukkan bahwa model berhasil memprediksi 75 True Positives (TP) dengan benar, yang berarti 75 kasus positif diprediksi dengan tepat. Namun, model juga menghasilkan 24 False Positives (FP), di mana 24 data negatif salah diprediksi sebagai positif, serta 16

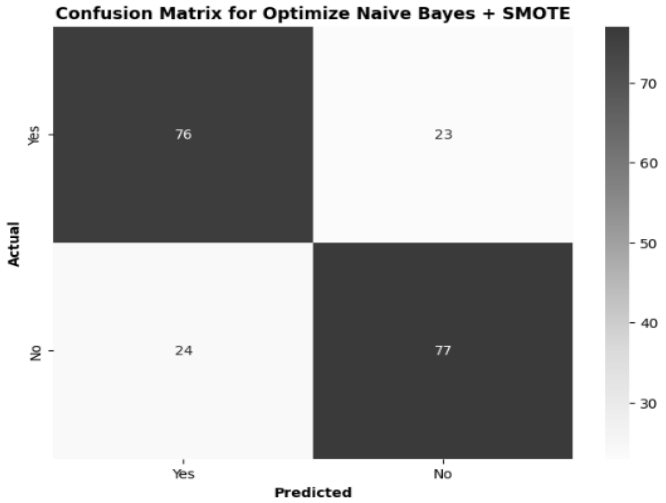
Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

False Negatives (FN), di mana 16 data positif salah diprediksi sebagai negatif. Di sisi lain, model berhasil memprediksi 39 True Negatives (TN) dengan benar, yaitu data negatif yang diprediksi sebagai negatif. Secara keseluruhan, confusion matrix ini menggambarkan bahwa model lebih efektif dalam mendeteksi True Positives, tetapi masih ada sejumlah False Positives dan False Negatives yang menunjukkan bahwa model ini dapat ditingkatkan lebih lanjut, khususnya dalam mengurangi kesalahan prediksi pada kedua kelas.

Tabel 7. Hasil Evaluasi Random

Parameter	Nilai
Accuracy	74%
Precision	75.1%
Recall	74%
F1-Score	74.4%

Tabel 7 menunjukkan hasil evaluasi model berdasarkan empat parameter utama, yaitu Accuracy, Precision, Recall, dan F1-Score, yang semuanya dikonversi ke dalam persen. Berdasarkan hasil tersebut, model memiliki accuracy sebesar 74%, yang menunjukkan persentase data yang diprediksi dengan benar. Precision berada pada 75.1%, yang mengindikasikan tingkat keakuratan model dalam mengidentifikasi kelas positif dibandingkan dengan jumlah prediksi positif yang dihasilkan. Recall tercatat sebesar 74%, menggambarkan kemampuan model dalam mendeteksi kelas positif dari seluruh data yang sebenarnya positif. Sedangkan F1-Score yang memiliki nilai 74.4%, merupakan rata-rata harmonis dari precision dan recall, memberikan gambaran umum mengenai keseimbangan antara keduanya. Secara keseluruhan, hasil ini menunjukkan kinerja model yang cukup baik meskipun masih ada ruang untuk perbaikan, terutama dalam mengoptimalkan keseimbangan antara precision dan recall.



Gambar 8. Confusion Matrix Naïve Bayes Setelah Smote

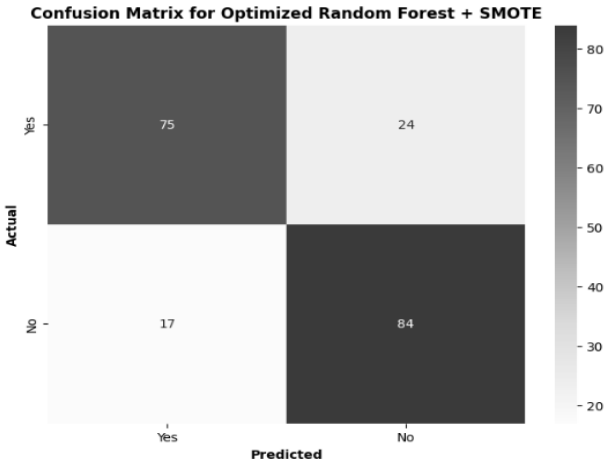
Gambar 8 menunjukkan confusion matrix model Naive Bayes yang dioptimalkan dengan SMOTE. Model menghasilkan 76 True Positives dan 77 True Negatives, menunjukkan kemampuan klasifikasi yang cukup baik. Namun, masih terdapat 23 False Positives dan 24 False Negatives yang mencerminkan kesalahan prediksi. Hasil ini menunjukkan bahwa meskipun kinerja model cukup baik, masih ada ruang untuk perbaikan guna meningkatkan akurasi.

Tabel 8. Hasil Evaluasi Naïve Bayes dengan SMOTE

Parameter	Nilai
Accuracy	76.5%
Precision	76.5%
Recall	76.5%
F1-Score	76.5%

Tabel 8 menunjukkan hasil evaluasi model Naive Bayes setelah diterapkan SMOTE dengan nilai accuracy, precision, recall, dan F1-score masing-masing sebesar 76.5%. Hal ini mengindikasikan bahwa model memiliki kinerja yang seimbang dalam hal memprediksi kelas positif dan negatif. Dengan accuracy sebesar 76.5%, model berhasil memprediksi 76.5% data dengan benar. Precision yang juga 76.5% menunjukkan bahwa dari semua prediksi positif, 76.5% adalah benar-benar positif, sementara recall yang sama menunjukkan bahwa 76.5% dari data positif berhasil terdeteksi oleh model. Terakhir, F1-score yang juga mencapai 76.5% menunjukkan keseimbangan yang baik antara precision dan recall, menandakan bahwa model cukup efektif dalam mendeteksi kelas positif tanpa terlalu banyak kesalahan.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446



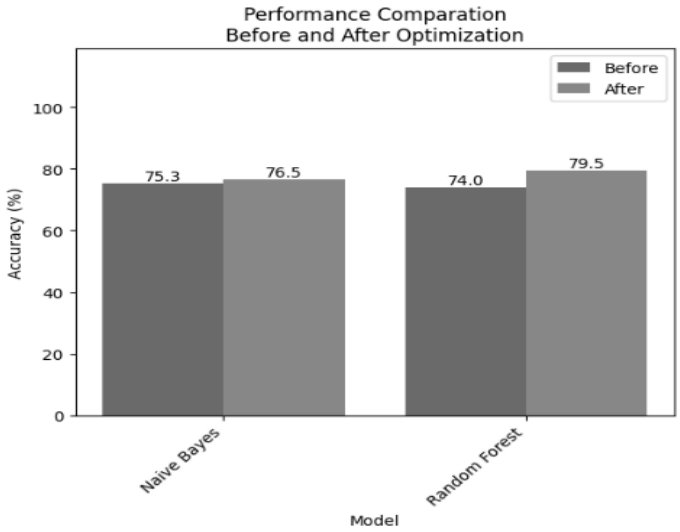
Gambar 9. Confusion Matriks Random Forest Setelah Smote

Confusion matrix pada Gambar 9 menunjukkan hasil klasifikasi model Random Forest yang telah dioptimalkan dengan SMOTE. Model berhasil mengklasifikasikan 75 data sebagai True Positive (TP) dan 84 sebagai True Negative (TN). Namun, terdapat 24 False Positive (FP) dan 17 False Negative (FN), yang menunjukkan masih adanya kesalahan prediksi. Secara keseluruhan, model menunjukkan kinerja yang baik, meskipun akurasinya masih dapat ditingkatkan dengan mengurangi kesalahan klasifikasi.

Tabel 9. Hasil Random Forest dengan SMOTE

Parameter	Nilai
Accuracy	79.5%
Precision	79.6%
Recall	79.5%
F1-Score	79.5%

Tabel 9 menunjukkan hasil evaluasi model Random Forest yang dioptimalkan dengan SMOTE dengan nilai accuracy, precision, recall, dan F1-score masing-masing sebesar 79.5%. Ini mengindikasikan bahwa model memiliki kinerja yang baik, dengan accuracy yang menunjukkan prediksi benar pada 79.5% data. Precision dan recall yang hampir sama menunjukkan kemampuan model dalam mendeteksi kelas positif dan meminimalkan kesalahan dalam prediksi positif. F1-score yang juga mencapai 79.5% menunjukkan keseimbangan yang baik antara precision dan recall. Secara keseluruhan, model menunjukkan performa yang solid meskipun ada sedikit ruang untuk perbaikan dalam mengurangi kesalahan klasifikasi.



Gambar 10. Diagram Sebelum dan Sesudah di Smote

Gambar 10 menunjukkan perbandingan akurasi antara model Naive Bayes dan Random Forest sebelum dan setelah diterapkannya SMOTE (Synthetic Minority Over-sampling Technique).

a. Pada model Naive Bayes, akurasi meningkat dari 75.3% sebelum SMOTE diterapkan menjadi 76.5% setelah SMOTE diterapkan.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

b. Pada model Random Forest, akurasi meningkat dari 74.0% sebelum SMOTE menjadi 79.5% setelah SMOTE diterapkan.

Penerapan SMOTE meningkatkan kinerja model, khususnya Random Forest, dengan menyeimbangkan distribusi kelas sehingga lebih mampu mengenali pasien diabetes. Dari tiga skenario pembagian data (70:30, 80:20, dan 90:10), skenario 80:20 memberikan hasil terbaik karena seimbang antara pelatihan dan pengujian. Sebaliknya, skenario 90:10 kurang optimal karena model cenderung overfitting

Tabel 10. Split Data Sebelum dan Sesudah Smote

Keterangan	Naïve Bayes			Random Forest		
	70 : 30	80 : 20	90 : 10	70 : 30	80 : 20	90 : 10
Sebelum SMOTE	72%	75,3%	66%	74%	74%	68%
Sesudah SMOTE	74%	76,5%	68%	78%	79,5%	72%

Berdasarkan Tabel 10, dari ketiga skenario pembagian data yang diuji (70:30, 80:20, dan 90:10), diperoleh bahwa skenario pembagian data 80:20 memberikan hasil yang paling optimal. Hal ini terlihat dari akurasi model yang paling tinggi pada pembagian tersebut, baik sebelum maupun sesudah penerapan metode SMOTE, khususnya pada algoritma Random Forest yang mencapai akurasi sebesar 79,5%. Pembagian data 80:20 memberikan keseimbangan yang baik antara jumlah data untuk pelatihan dan pengujian, sehingga mampu meningkatkan kemampuan generalisasi model secara signifikan. Oleh karena itu, dalam penelitian ini digunakan skenario pembagian data 80:20 sebagai metode pembagian data yang diimplementasikan, karena terbukti menghasilkan performa prediksi yang paling optimal dibandingkan skenario lainnya

4. KESIMPULAN

Berdasarkan penelitian ini, dapat disimpulkan bahwa penerapan SMOTE (Synthetic Minority Over-sampling Technique) memberikan dampak yang signifikan terhadap kinerja model Naive Bayes dan Random Forest dalam mengklasifikasikan pasien diabetes. Sebelum SMOTE diterapkan, kedua model menunjukkan akurasi yang cukup baik, namun hasil tersebut terbatas oleh ketidakseimbangan kelas dalam dataset, dengan kelas 0 (tidak diabetes) yang mendominasi. Setelah SMOTE diterapkan untuk menyeimbangkan distribusi kelas, akurasi model meningkat secara signifikan, terutama pada model Random Forest, yang menunjukkan peningkatan dari 74% menjadi 79.5%, sementara model Naive Bayes meningkat dari 75.3% menjadi 76.5%. Penerapan SMOTE terbukti efektif dalam menangani masalah ketidakseimbangan kelas, yang sebelumnya menghambat model dalam mendeteksi kelas minoritas (diabetes). SMOTE membantu model untuk belajar lebih baik pada data yang kurang terwakili, yang pada akhirnya meningkatkan kemampuan model dalam mengklasifikasikan pasien dengan diabetes secara lebih akurat. Hasil evaluasi model menunjukkan bahwa baik Naive Bayes maupun Random Forest memiliki performa yang solid setelah penerapan SMOTE, dengan Random Forest menunjukkan hasil yang lebih optimal. Secara keseluruhan, penelitian ini memberikan gambaran yang jelas bahwa teknik SMOTE sangat berguna dalam meningkatkan performa model klasifikasi pada dataset yang tidak seimbang. Penerapan teknik ini dapat meningkatkan kinerja model dalam berbagai aplikasi medis dan prediksi lainnya yang memiliki distribusi kelas tidak seimbang, seperti dalam kasus prediksi diabetes ini.

REFERENCES

[1] BPS (Badan Pusat Statistik), “Statistik Kesehatan Indonesia: Diabetes Melitus,” Jakarta, Jun. 2021. Accessed: Jun. 05, 2025. [Online]. Available: https://www.bps.go.id/id/publication/2021/12/22/0f207323902633342a1f6b01/profil-statistik-kesehatan-2021.html

[2] Kementerian Kesehatan Republik Indonesia, “Profil Kesehatan Indonesia 2021,” Jakarta, Jun. 2021. Accessed: Jun. 05, 2025. [Online]. Available: https://kemkes.go.id/app_asset/file_content_download/Profil-Kesehatan-Indonesia-2021.pdf

[3] American Diabetes Association, “Diabetes Mellitus: Diagnosis and Management,” *Diabetes Care*, vol. 44, no. Supplement_1, pp. S151–S167, Jan. 2021, doi: 10.2337/dc21-S011.

[4] International Diabetes Federation (IDF), *IDF Diabetes Atlas, 10th Edition*, 10th Edition. Brussels, Belgium: International Diabetes Federation, 2021. Accessed: Jun. 05, 2025. [Online]. Available: https://diabetesatlas.org/atlas/tenth-edition/

[5] World Health Organization, “Global Report on Diabetes,” Geneva, Jun. 2020. Accessed: Jun. 05, 2025. [Online]. Available: https://www.who.int/publications/i/item/9789241565257

[6] D. Hermawan and A. Supriyadi, “Analisis Penggunaan Machine Learning dalam Diagnosis Diabetes,” *Jurnal Teknologi Kesehatan*, vol. 15, no. 4, pp. 233–240, 2022.

[7] M. A. Alam, A. Sohel, K. M. Hasan, and M. A. Islam, “Machine Learning And Artificial Intelligence in Diabetes Prediction And Management: A Comprehensive Review of Models,” *Innovatech Engineering Journal*, vol. 1, no. 01, pp. 107–124, Dec. 2024, doi: 10.70937/jnes.v1i01.41.

[8] D. ; S. H. ; S. R. A. Nugroho, “Implementasi Naive Bayes dan Random Forest dalam Klasifikasi Diabetes,” *Jurnal Informatika Medis*, vol. 12, no. 3, pp. 112–118, 2021, Accessed: Jun. 04, 2025. [Online]. Available: https://www.kaggle.com/datasets

[9] H. Mansourifar and W. Shi, “Deep Synthetic Minority Over-Sampling Technique,” *arXiv preprint arXiv:2003.09788*, Mar. 2020, doi: 10.48550/arXiv.2003.09788.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 431−443
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7446

[10] C. Y. Huang and H. L. Dai, "Learning from class-imbalanced data: review of data driven methods and algorithm driven methods," *Data Science in Finance and Economics*, vol. 1, no. 1, pp. 21–36, 2021, doi: 10.3934/DSFE.2021002.
[11] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artif Intell Rev*, vol. 57, no. 6, Jun. 2024, doi: 10.1007/s10462-024-10759-6.
[12] W. Hasanah and L. C. Munggaran, "Comparison of Naïve Bayes and Random Forest Methods for Diabetes Prediction," *Int J Comput Appl*, vol. 174, no. 26, pp. 1–6, Mar. 2021, Accessed: Jun. 05, 2025. [Online]. Available: https://www.ijcaonline.org/archives/volume174/number26/31837-2021921184/
[13] K. Huda and M. Ula, "Penerapan Naive Bayes, Regresi Logistik, Random Forest, Svm, Dan Knn Untuk Prediksi Diabetes," SENASTIKA Universitas Malikussaleh, Nov. 2024. Accessed: Jun. 05, 2025. [Online]. Available: https://proceedings.unimal.ac.id/senastika/article/view/853
[14] Y. , L. W. , Z. T. Chen, "Evaluating the impact of different features on the performance of machine learning models in predicting diabetes," *Journal of Data Science*, vol. 18, no. 4, pp. 489–499, 2020.
[15] I. H. Sarker, "Machine learning in healthcare: An introduction and review," *J Healthc Eng*, pp. 1–17, 2021, doi: 10.1155/2021/9952417.
[16] Scikit-learn Developers, "Naive Bayes classifier," Scikit-learn Documentation. Accessed: Jun. 05, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html
[17] GeeksforGeeks, "Naive Bayes Classifiers." Accessed: Jun. 05, 2025. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/
[18] Sunil Ray, "Naive Bayes Classifier Explained With Practical Problems." Accessed: Jun. 05, 2025. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
[19] M. Shahhosseini and G. Hu, "Improved weighted random forest for classification problems," *ArXiv*, 2020.
[20] T. Fulazzaky, A. Saefuddin, and A. M. Soleh, "Evaluating Ensemble Learning Techniques for Class Imbalance in Machine Learning: A Comparative Analysis of Balanced Random Forest, SMOTE-RF, SMOTEBoost, and RUSBoost," *Scientific Journal of Informatics*, vol. 11, no. 4, pp. 969–980, Dec. 2024, doi: 10.15294/sji.v11i4.15937.