

ANALISIS PERBANDINGAN ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE UNTUK PREDIKSI PENYAKIT DIABETES MELITUS

Mohammad Baharudin Yusuf^{1*}, Sofi Defiyanti², Rini Mayasari³

¹2210631170079@student.unsika.ac.id, ²sofi.defiyanti@unsika.ac.id,

³rini.mayasari@staff.unsika.ac.id. Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang; Jl. H.S. Ronggowaluyo, Telukjambe Timur, Karawang 41361; Telp. (0267) 641177

Keywords:

classification;
random forest;
SMOTE;
support vector machine;
type-2 diabetes mellitus

Correspondent Email:

2210631170079@student.unsika.ac.id

Abstrak. Prediksi dini Diabetes Melitus tipe-2 penting dilakukan untuk membantu proses diagnosis dan pengambilan keputusan klinis secara lebih cepat dan akurat. Penelitian ini bertujuan membandingkan kinerja algoritma Random Forest dan Support Vector Machine (SVM) dalam prediksi diabetes serta menganalisis pengaruh penerapan Synthetic Minority Over-sampling Technique (SMOTE) pada data tidak seimbang. Penelitian menggunakan dataset Diabetes Bangladesh 2025 dengan tahapan data selection, preprocessing, data transformation, pemodelan, dan evaluasi. Tahap preprocessing meliputi imputasi median dan penghapusan data duplikat, sedangkan standarisasi fitur dilakukan setelah pembagian data untuk mencegah data leakage. Dataset dibagi menggunakan rasio 80:20 dengan teknik stratifikasi. Penelitian menerapkan dua skenario eksperimen, yaitu tanpa SMOTE dan dengan SMOTE yang hanya diterapkan pada data latih. Evaluasi model dilakukan menggunakan metrik accuracy, precision, recall, dan F1-score berdasarkan data uji. Hasil penelitian menunjukkan bahwa Random Forest memberikan performa terbaik dibandingkan SVM. Pada skenario tanpa SMOTE, Random Forest memperoleh accuracy 94,8%, precision 96,4%, recall 97,0%, dan F1-score 96,7%, sedangkan SVM memperoleh accuracy 91,5% dan F1-score 94,6%. Setelah penerapan SMOTE, performa Random Forest sedikit meningkat dengan accuracy 95,3% dan F1-score 97,0%, sedangkan performa SVM menurun menjadi accuracy 90,6% dan F1-score 93,9%. Hasil penelitian menunjukkan bahwa Random Forest merupakan model terbaik untuk prediksi Diabetes Melitus tipe-2 pada dataset yang digunakan.



Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Abstract. Early prediction of type 2 diabetes mellitus is important to facilitate faster and more accurate diagnosis and clinical decision-making. This study aims to compare the performance of the Random Forest and Support Vector Machine (SVM) algorithms in predicting diabetes and to analyze the effect of applying the Synthetic Minority Over-sampling Technique (SMOTE) to imbalanced data. The study used the Bangladesh Diabetes 2025 dataset, following the stages of data selection, preprocessing, data transformation, modeling, and evaluation. The preprocessing stage included median imputation and the removal of duplicate data, while feature standardization was performed after data splitting to prevent data leakage. The dataset was split using an 80:20 ratio with a stratification technique. The study applied two experimental scenarios: one without SMOTE and one with SMOTE applied only to the training data. Model evaluation was conducted using the metrics accuracy, precision, recall, and F1-score based on the test data. The results show that Random Forest outperforms SVM. In the scenario without SMOTE, Random Forest achieved an accuracy of 94.8%, precision of 96.4%, recall of 97.0%, and an F1-score of 96.7%, while SVM achieved an accuracy

of 91.5% and an F1-score of 94.6%. After applying SMOTE, Random Forest's performance improved slightly to an accuracy of 95.3% and an F1-score of 97.0%, while SVM's performance declined to an accuracy of 90.6% and an F1-score of 93.9%. The results of the study show that Random Forest is the best model for predicting type 2 diabetes mellitus in the dataset used.

1. PENDAHULUAN

Diabetes melitus saat ini menjadi salah satu beban kesehatan global paling serius pada abad ke-21. Kondisi metabolik kronis ini muncul ketika tubuh tidak mampu memproduksi insulin dalam jumlah cukup atau tidak dapat menggunakan insulin secara optimal, yang mengakibatkan tingginya kadar glukosa dalam darah (hiperglikemia). Menurut laporan Federasi Diabetes Internasional (IDF) pada tahun 2025, sekitar 589 juta orang dewasa di dunia mengidap diabetes. Pada tahun 2024, penyakit ini berkontribusi pada 3,4 juta kematian, dan diproyeksikan angkanya akan melonjak hingga 853 juta pada tahun 2050. Lonjakan prevalensi yang masif ini menuntut adanya sistem deteksi dini yang cepat, murah, dan akurat guna meminimalkan risiko komplikasi kronis seperti penyakit kardiovaskular, gagal ginjal, dan kerusakan saraf.

Gambar 1. Peningkatan jumlah pengidap dan kematian penyakit diabetes



(sumber: [1])

Dalam beberapa tahun terakhir, implementasi *machine learning* berkembang pesat untuk mengeksplorasi pola-pola tersembunyi dari data medis pasien demi membantu diagnosis dini. Dua algoritma yang sering diadu kekuatannya dalam tugas klasifikasi biner data klinis adalah *Random Forest* (RF) dan *Support Vector Machine* (SVM). RF dikenal sangat stabil karena mengandalkan pendekatan *ensemble learning* berbasis *bagging* untuk mereduksi *overfitting*. Di sisi lain, SVM memiliki reputasi unggul dalam memetakan batas keputusan (*decision*

boundary) yang optimal pada ruang berdimensi tinggi melalui pemanfaatan fungsi *kernel*.

Sejumlah penelitian terdahulu telah membandingkan efektivitas kedua algoritma ini dalam prediksi diabetes, namun mencatat hasil yang saling berkontradiksi. Studi yang dilakukan oleh [2], [3], [4] secara konsisten melaporkan keunggulan performa algoritma *Random Forest* dibandingkan SVM. Sebaliknya, studi oleh [5] justru menemukan bahwa SVM menyajikan akurasi yang lebih superior (98,33%) daripada *Random Forest* (97,5%). Perbedaan temuan ini (juga dikenal sebagai *research gap*) terjadi karena eksperimen-eksperimen tersebut menggunakan karakteristik *dataset* yang berbeda, tahapan pra-pemrosesan yang tidak seragam, serta ketidakjelasan kontrol atas penanganan ketidakseimbangan kelas (*class imbalance*).

Oleh karena itu, keterbaruan (*novelty*) dari penelitian ini diposisikan sebagai *controlled comparative study* sekaligus *replication study* untuk menjernihkan inkonsistensi temuan literatur tersebut. Penelitian ini secara ketat menyamakan seluruh tahapan pra-pemrosesan, menggunakan dataset klinis Asia Selatan yang mutakhir [6] (*Type-2 Diabetes Dataset Bangladesh 2025*), serta mengendalikan penyeimbangan data lewat teknik *Synthetic Minority Over-sampling Technique* (SMOTE) yang hanya diisolasi pada data latih demi menghindari *data leakage*. Tujuan utama dari penelitian ini adalah memberikan bukti empiris yang objektif dan *apple-to-apple* mengenai performa relatif RF dan SVM dalam memprediksi status diabetes melitus tipe-2.

2. TINJAUAN PUSTAKA

2.1 Diabetes Melitus

Diabetes Melitus (DM) adalah sekelompok penyakit metabolik kronis yang ditandai dengan tingginya kadar glukosa darah (hiperglikemia). Kondisi ini terjadi akibat

kelainan pada sekresi insulin, kerja insulin, atau keduanya. Insulin adalah hormon yang diproduksi oleh pankreas yang berfungsi membantu sel-sel tubuh menyerap glukosa untuk diubah menjadi energi. Jika tidak dikelola dengan baik, hiperglikemia kronis dapat menyebabkan kerusakan jangka panjang pada berbagai organ, terutama mata, ginjal, saraf, jantung, dan pembuluh darah. Deteksi dini menjadi sangat krusial untuk mencegah komplikasi tersebut.

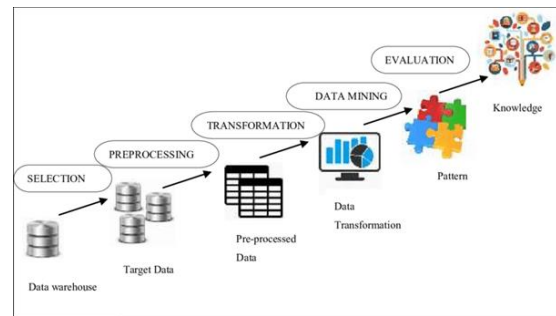
2.2 Machine Learning (Pembelajaran Mesin)

Salah satu tugas utama dalam ML adalah klasifikasi. Klasifikasi merupakan teknik *supervised learning* (pembelajaran terarah) di mana algoritma belajar untuk memetakan data input ke dalam kategori atau kelas diskrit yang telah ditentukan sebelumnya. Dalam konteks penelitian ini, klasifikasi digunakan untuk memprediksi apakah seorang pasien termasuk dalam kelas "Diabetes" atau "Tidak Diabetes" berdasarkan atribut-atribut (fitur) medis yang relevan.

2.3 Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases (KDD) merupakan suatu proses sistematis dan berurutan yang bertujuan untuk mengekstraksi pengetahuan (*knowledge*) yang bernilai, valid, dan dapat dipahami dari kumpulan data berukuran besar. KDD tidak hanya berfokus pada penerapan algoritma *data mining*, tetapi mencakup keseluruhan tahapan mulai dari pemilihan data, pra-pemrosesan, transformasi, hingga evaluasi hasil, sehingga pengetahuan yang dihasilkan dapat digunakan secara efektif dalam pengambilan Keputusan. Dalam bidang kesehatan, KDD berperan penting karena data medis umumnya bersifat besar, heterogen, dan kompleks, sehingga memerlukan proses analisis yang terstruktur dan terkontrol.

Berdasarkan kerangka kerja KDD, proses ini terdiri dari beberapa tahapan utama, yaitu *selection*, *preprocessing*, *transformation*, *data mining*, dan *evaluation*, yang secara keseluruhan bertujuan menghasilkan pengetahuan baru dari data seperti yang terlihat pada gambar berikut.



Gambar 2. Tahapan KDD (*Knowledge Discovery in Databases*)

(sumber: (Novitasari et al., 2023))

2.4 Random Forest

Random Forest (RF) merupakan metode klasifikasi *ensemble learning* yang bekerja dengan membangun sekumpulan besar pohon keputusan (*decision trees*) secara independen selama fase pelatihan. Algoritma ini mengombinasikan konsep *bootstrap aggregating* (*bagging*) dan pemilihan fitur acak pada setiap simpul pohon. Keputusan akhir klasifikasi ditentukan berdasarkan mekanisme *voting* terbanyak (*majority voting*) dari seluruh pohon yang terbentuk, sehingga menghasilkan model yang stabil dan tangguh terhadap masalah *overfitting*.

2.4.1 Tahapan Kerja Random Forest

1. Pembentukan *Bootstrap Samples*. Dari dataset awal dengan N sampel, algoritma mengambil sampel acak sebanyak N (dengan penggantian) untuk membentuk *bootstrap samples*. Setiap pohon dilatih pada *subset* data yang berbeda.
2. Pembentukan Pohon Keputusan. Untuk setiap node, hanya sebagian fitur yang dipilih secara acak untuk menentukan pemisahan terbaik. Misalnya, jika dataset memiliki m fitur, maka hanya $\sqrt{m}$ fitur yang diuji.
3. Prediksi dengan *Voting* atau *Averaging*. Untuk klasifikasi, hasil akhir diperoleh melalui *voting*:

$$\hat{y} = \text{mode}(h_1(x), h_2(x), \dots, h_k(x)) \quad (1)$$
 di mana $h_i(x)$ adalah hasil prediksi dari pohon ke- i .
4. Evaluasi Kinerja. Hasil prediksi dibandingkan dengan label aktual untuk menghitung metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

2.5 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma pembelajaran terarah (*supervised learning*) yang bertujuan mencari sebuah *hyperplane* (bidang pembatas) optimal yang memisahkan dua kelas objek dengan jarak margin maksimal. Data terdekat yang membatasi posisi *hyperplane* disebut sebagai *support vectors*. Dalam menghadapi sebaran data klinis yang bersifat *non-linear*, SVM memanfaatkan fungsi *kernel Radial Basis Function* (RBF) untuk memetakan data asli ke ruang vektor berdimensi lebih tinggi agar dapat dipisahkan secara *linear*.

2.6 Prinsip Kerja dan Formulasi Matematis SVM

SVM bertujuan untuk menemukan *hyperplane* optimal yang dapat memisahkan dua kelas data dengan margin maksimum. Secara matematis, fungsi keputusan SVM dapat dituliskan sebagai:

$$f(x) = w^T x + b \quad (2)$$

dengan w adalah vektor bobot dan b merupakan bias. Model berusaha memaksimalkan jarak antar kelas (margin) yang dirumuskan sebagai:

$$\text{Margin} = \frac{2}{\|w\|} \quad (3)$$

Agar pemisahan berhasil, setiap titik data x_i dengan label kelas $y_i \in \{-1, +1\}$ harus memenuhi:

$$y_i(w^T x_i + b) \geq 1 \quad (4)$$

Untuk data yang tidak terpisahkan secara *linear*, SVM menggunakan fungsi *kernel*, seperti *kernel RBF*, yang didefinisikan sebagai:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|) \quad (5)$$

Parameter γ mengontrol jarak pengaruh tiap sampel pelatihan terhadap model. Kombinasi parameter C (regularisasi) dan γ (*kernel*) berperan penting dalam mengontrol keseimbangan antara margin yang besar dan kesalahan klasifikasi minimum. Nilai C yang terlalu besar dapat menyebabkan model terlalu fokus pada data pelatihan dan berpotensi *overfitting*, sementara nilai γ mengatur seberapa jauh pengaruh satu titik data pelatihan dalam membentuk batas keputusan pada fungsi *kernel RBF*. Penelitian sebelumnya menunjukkan bahwa pemilihan kombinasi parameter C dan γ yang optimal mampu meningkatkan akurasi model klasifikasi secara signifikan [8].

2.7 Ketidakseimbangan Data (Class Imbalance)

Masalah *class imbalance* muncul ketika jumlah sampel pada satu kelas jauh lebih sedikit dibandingkan kelas lain, yang umum terjadi pada *dataset* kesehatan seperti diabetes, di mana jumlah kasus positif seringkali jauh lebih kecil dari jumlah non-diabetes. Ketidakseimbangan ini dapat menyebabkan model *machine learning* bias terhadap kelas mayoritas, sehingga tingkat deteksi kasus minoritas (misalnya penderita diabetes) menjadi rendah dan memengaruhi metrik evaluasi seperti *recall* dan *F1-score* secara negatif. Teknik *resampling* seperti *oversampling* dan *undersampling* sering digunakan untuk menangani permasalahan ini dalam penelitian klasifikasi berbasis *machine learning*.

2.8 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE merupakan teknik penyeimbangan data berbasis *oversampling* yang dirancang khusus untuk mengatasi kendala *class imbalance*. Berbeda dari *random oversampling* biasa yang menduplikasi baris secara mentah, SMOTE bekerja dengan cara memproduksi sampel-sampel sintetis baru untuk kelas minoritas melalui teknik interpolasi *linear* di antara data minoritas asli dan tetangga terdekatnya (*k-nearest neighbors*) dalam ruang fitur.

2.9 Confusion Matrix

		Actual value	
		Positive	Negative
Predicted value	Positive	True Positive	False Negative
	Negativ	False Positive	True Negative

Gambar 3. Confusion matrix
(sumber: [9])

Confusion Matrix adalah tabel dua dimensi yang memetakan performa riil sistem klasifikasi dengan membandingkan nilai aktual kelas target terhadap nilai prediksi model. Matriks ini terdiri atas empat parameter

esensial: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Parameter-parameter ini kemudian digunakan untuk menghitung nilai metrik evaluasi ilmiah yang meliputi tingkat Akurasi (*Accuracy*), Presisi (*Precision*), *Recall* (Sensitivitas), serta *F1-score* sebagai rata-rata harmonik.

2.10 Metrik Evaluasi Klasifikasi

1. *Accuracy* (Akurasi): Mengukur proporsi total prediksi yang benar (positif dan negatif) terhadap jumlah total data.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

2. *Precision* (Presisi): Mengukur proporsi prediksi positif yang benar (*True Positive*) dari total prediksi yang dikategorikan sebagai positif. Presisi penting ketika biaya *False Positive* tinggi.

$$Precision = \frac{TP}{(TP+FP)} \quad (7)$$

3. *Recall* (Perolehan) / Sensitivitas: Mengukur proporsi prediksi positif yang benar (*True Positive*) dari total data yang sebenarnya positif. *Recall* sangat penting ketika biaya *False Negative* tinggi (misalnya, gagal mendeteksi pasien yang sakit).

$$Recall = \frac{TP}{(TP+FN)} \quad (8)$$

4. *F1-Score*: Rata-rata harmonik (*harmonic mean*) dari *Precision* dan *Recall*. Metrik ini memberikan keseimbangan antara kedua metrik tersebut, terutama berguna jika dataset tidak seimbang (*imbalanced*).

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (9)$$

Metrik-metrik ini penting untuk memastikan bahwa model tidak hanya akurat secara keseluruhan, tetapi juga seimbang dalam mendeteksi kasus positif diabetes. Evaluasi dengan menggunakan *precision*, *recall*, dan *F1-score* sangat diperlukan untuk menilai kemampuan model dalam mengenali kasus positif tanpa mengorbankan akurasi umum, terutama ketika data yang digunakan bersifat tidak seimbang [10], [11].

3. METODE PENELITIAN

3.1 Dataset

Penelitian ini mengeksplorasi data sekunder terstruktur berformat CSV dengan tajuk "*Type-2 Diabetes Dataset Bangladesh*" tahun 2025 yang diperoleh dari repositori terbuka Mendeley Data. *Dataset* ini mencakup rekaman rekam medis dari 1.065 pasien. Struktur data tersusun atas 9 atribut klinis selaku fitur independen (*No. of Pregnancy*, *Age*, *BMI*, *BP Systolic*, *BP Diastolic*, *Diabetes Pedigree Function*, *Insulin*, *Skin Thickness*, dan *Glucose*) serta 1 atribut target biner (*Type-2 Diabetic*). Distribusi awal target menunjukkan kondisi tidak seimbang (*imbalanced code*) dengan rincian 840 pasien positif diabetes (kelas 1) dan 225 pasien non-diabetes (kelas 0).

Tabel 1. Atribut dan Deskripsi Dataset Diabetes Prediction

No	Nama Atribut	Tipe Data	Deskripsi
1	<i>No. of Pregnancy</i>	Numerik (<i>Integer</i>)	Jumlah kehamilan yang pernah dialami pasien.
2	<i>Age</i>	Numerik (<i>Integer</i>)	Usia pasien dalam tahun.
3	<i>BMI</i>	Numerik (<i>Decimal</i>)	Indeks Massa Tubuh (kg/m ²).
4	<i>BP (Systolic)</i>	Numerik (<i>Integer</i>)	Tekanan darah sistolik (mmHg).
5	<i>BP (Diastolic)</i>	Numerik (<i>Integer</i>)	Tekanan darah diastolik (mmHg).
6	<i>Diabetes Pedigree Function</i>	Numerik (<i>Decimal</i>)	Fungsi silsilah diabetes (indeks risiko genetik).
7	<i>Insulin</i>	Numerik (<i>Integer</i>)	Konsentrasi insulin serum saat puasa (μU/mL).
8	<i>Skin Thickness (mm)</i>	Numerik (<i>Integer</i>)	Ketebalan lipatan kulit trisep (mm).
9	<i>Glucose</i>	Numerik (<i>Integer</i>)	Konsentrasi glukosa plasma setelah tes toleransi

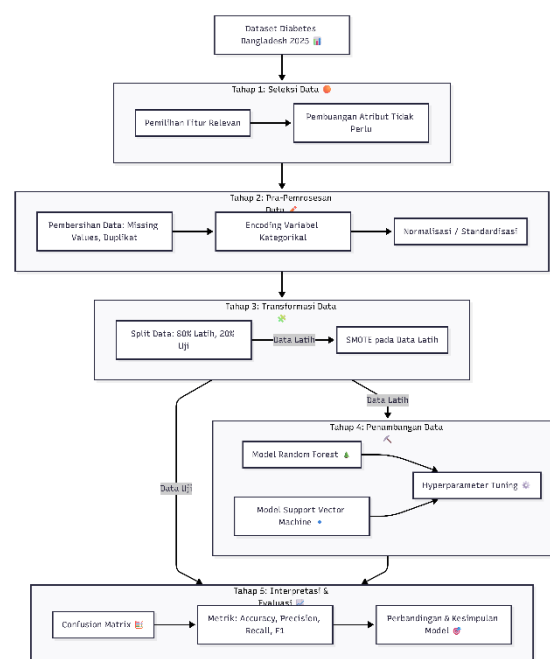
			glukosa oral (mg/dL).
10	Type-2 Diabetic (Target)	Kategorik (0/1)	Status diabetes: 0 = Non-Diabetes, 1 = Diabetes.

3.2 Tahapan Penelitian

Alur pengerjaan penelitian ini mengadopsi metodologi *Knowledge Discovery in Databases* (KDD) yang mencakup lima fase terstruktur:

1. *Data Selection*: Pemuatan dataset *Type-2 Diabetes Dataset Bangladesh 2025* ke dalam repositori kerja Google Colab.
2. *Data Preprocessing*: Identifikasi data kosong (*missing values*) yang dilanjutkan dengan eksekusi median *imputation*, serta analisis data duplikat.
3. *Data Transformation*: Pembagian porsi data (*train-test split*) dengan rasio stratifikasi 80:20 (852 data latih dan 213 data uji). Penskalaan nilai fitur menggunakan *StandardScaler* diaplikasikan tepat setelah pemisahan data dilakukan. Teknik *SMOTE* kemudian disuntikkan secara eksklusif hanya pada *subset* data latih untuk menyeimbangkan kelas tanpa memicu kebocoran data (*data leakage*).
4. *Data Mining*: Proses penginstansian, pelatihan, dan pengujian model klasifikasi menggunakan bahasa Python.
5. *Evaluation*: Rekapitulasi performa model melalui pengisian tabel kontingensi *Confusion Matrix* dan perhitungan skor metrik uji.

Berikut diagram alir yang mengilustrasikan tahapan penelitian secara visual, mulai dari awal hingga penarikan kesimpulan.



Gambar 4. Diagram Alir Penelitian

3.3 Model dan Hyperparameter

Untuk mewujudkan perbandingan yang adil (*baseline evaluation*), eksperimen ini sengaja menggunakan parameter bawaan (*default*) dari pustaka *Scikit-Learn* tanpa adanya modifikasi manual. Spesifikasi nilai parameter esensial yang berjalan adalah sebagai berikut:

1. *Random Forest*: Menggunakan modul *RandomForestClassifier()*, dengan jumlah pohon (*n_estimators*) = 100, kedalaman maksimal (*max_depth*) = *None*, dan ambang bagi daun (*min_samples_split*) = 2.
2. *Support Vector Machine*: Menggunakan modul *SVC()*, bersandar pada fungsi *kernel* = 'rbf', parameter penalti (*C*) = 1.0, dan koefisien *kernel* (*gamma*) = 'scale'.

3.4 Evaluasi

Kualitas luaran dari klasifikasi dinilai secara kuantitatif berdasarkan pementasan empat metrik evaluasi utama: *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, yang seluruhnya dihitung secara matematis bersumber dari keluaran tabel *Confusion Matrix* data uji.

4. HASIL DAN PEMBAHASAN

4.1 Data Selection

Tahap data *selection* merupakan proses awal dalam penelitian ini yang bertujuan untuk mengidentifikasi, memahami, dan memastikan kelayakan *dataset* sebelum dilakukan proses analisis lebih lanjut.

Berikut adalah statistik ringkasan *dataset* yang ditampilkan pada Tabel 2.

Tabel 2. Statistik deskriptif

Atribut	Rat a- rat a	Stan dar Devi asi	Mini mum	Maksi mum
<i>No. of Pregnancy</i>	3,11	1,88	0	12
<i>Age</i>	44,47	12,69	21	86
BMI	25,25	3,80	17,18	37,44
BP (<i>Systolic</i>)	119,35	14,29	80	180
BP (<i>Diastolic</i>)	79,83	10,77	30	120
DiabetesPedigre eFunction	1,49	1,56	0	8
Insulin	5,41	8,88	0	44
<i>Skin Thickness</i> (mm)	307,24	35,37	29,21	406,40
<i>Glucose</i>	138,45	64,83	50	450

Berdasarkan Tabel 2, rata-rata usia pasien berada pada kisaran 44 tahun, dengan rentang usia 21 hingga 86 tahun. Nilai rata-rata BMI berada pada kategori *overweight* ringan, sementara kadar glukosa menunjukkan variasi yang sangat lebar (50–450 mg/dL), yang mengindikasikan adanya pasien dengan kondisi normoglikemia hingga hiperglikemia berat.

4.2 Hasil Preprocessing

Pada fase data *preprocessing*, pelacakan awal menemukan adanya 4 nilai kosong pada atribut BP(*Diastolic*) dan 7 nilai kosong pada *Skin Thickness*(mm). Nilai kosong tersebut berhasil ditangani secara penuh menggunakan teknik imputasi statistik berbasis median. Sebanyak 569 baris data terdeteksi memiliki nilai yang identik (duplikat), namun diputuskan

untuk dipertahankan karena secara klinis merepresentasikan karakteristik riil kelompok pasien dalam populasi diabetes. Ringkasan akhir dari pengelolaan data ini dirangkum pada Tabel 3.

Tabel 3. Ringkasan Hasil Pra-pemrosesan Data

Tahap <i>Preprocessing</i>	Strategi / Metode Penanganan	Hasil Akhir Kondisi Data
<i>Missing Value</i>	Median Imputation (BP=80.0; Skin=304.8)	Teratasi penuh (0 missing values)
<i>Data Duplicate</i>	Dipertahankan demi validitas klinis distribusi	569 rekaman duplikat dipertahankan
<i>Encoding</i>	Tidak diterapkan (Fitur sudah bertipe angka)	Struktur data asli numerik konsisten
<i>Standardization</i>	Transformasi skala lewat StandardScaler	Rentang nilai fitur seragam

4.3 Data Transformation

Pada tahap transformasi, dilakukan pembagian data latih dan uji, standardisasi fitur, serta penerapan SMOTE pada skenario tertentu. Seluruh proses dilakukan secara terkontrol untuk menghindari data *leakage*.

1. Train-Test Split (80:20)

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    stratify=y,
    random_state=RANDOM_STATE
)

print("Jumlah data latih:", X_train.shape)
print("Jumlah data uji:", X_test.shape)
```

```
Jumlah data latih: (852, 9)
Jumlah data uji: (213, 9)
```

Gambar 5. Kode proses pembagian data dan hasilnya

Data dibagi menjadi data latih (80%) dan data uji (20%) menggunakan `train_test_split` dengan parameter `stratify=y` untuk menjaga proporsi kelas.

Hasilnya diperoleh 852 data latih dan 213 data uji.

2. Standardisasi Fitur (*Feature Scaling*)

```
scaler = StandardScaler()

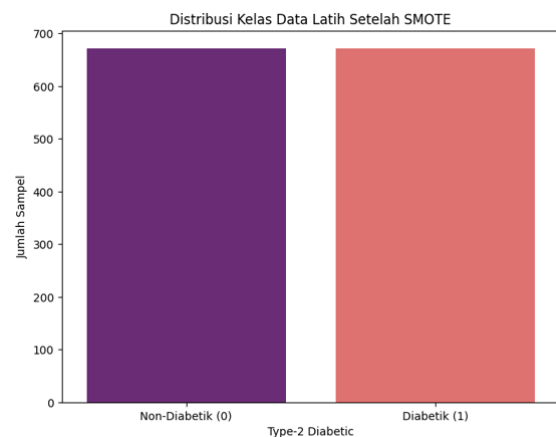
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Gambar 6. Kode standardisasi fitur

Standardisasi fitur menggunakan `StandardScaler` diterapkan pada data latih (`fit_transform`) dan data uji (`transform`). Proses `fit_transform()` hanya dilakukan pada data latih, sedangkan data uji hanya melalui `transform()`. Hal ini penting untuk mencegah data *leakage*, karena informasi statistik dari data uji tidak boleh memengaruhi proses pelatihan. Standardisasi sangat penting khususnya untuk SVM, karena algoritma ini berbasis jarak (*distance-based*) sehingga sensitif terhadap skala fitur. Tanpa *scaling*, fitur dengan rentang besar seperti *Glucose* dapat mendominasi fitur lain.

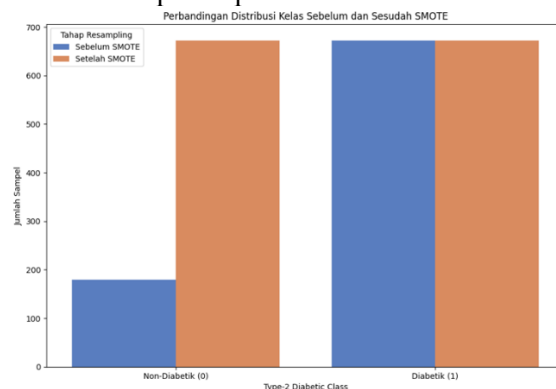
3. Penerapan SMOTE pada Data Latih
Untuk mengatasi ketidakseimbangan kelas, pada skenario ketiga dan keempat diterapkan teknik *Synthetic Minority Oversampling Technique* (SMOTE). Penerapan SMOTE hanya dilakukan pada data latih setelah proses *train-test split* untuk menghindari data *leakage*. Data uji tetap dipertahankan dalam kondisi aslinya.

Penerapan SMOTE bertujuan untuk menyeimbangkan jumlah sampel antar kelas pada data latih dengan menambah sampel sintetis pada kelas minoritas hingga menyamai jumlah kelas mayoritas, bukan didasarkan pada label kelas tertentu secara spesifik. Berdasarkan hasil eksekusi, sistem mendeteksi bahwa kelas mayoritas pada data latih adalah sebanyak 672 sampel. Oleh karena itu, SMOTE melakukan *resampling* pada kelas minoritas (dari 180 sampel) dengan cara interpolasi hingga jumlahnya mencapai 672 sampel. Hasilnya, distribusi data latih menjadi seimbang secara sempurna (672 vs 672), sehingga model dapat mempelajari karakteristik kedua kelas secara adil tanpa mengalami bias terhadap salah satu kelompok.



Gambar 7. Distribusi Kelas Data Latih Setelah SMOTE

Untuk memperkuat analisis dan menunjukkan efektivitas SMOTE secara lebih komprehensif, dibuat visualisasi perbandingan distribusi kelas sebelum dan sesudah penerapan SMOTE. Grafik perbandingan yang dihasilkan pada Gambar 8 menyajikan informasi yang sangat informatif tentang perubahan distribusi kelas akibat penerapan SMOTE.



Gambar 8. Grafik perbandingan distribusi kelas sebelum dan sesudah SMOTE

4.4 Data Mining

Pada penelitian ini, tahap *data mining* difokuskan pada pembangunan model klasifikasi untuk memprediksi keberadaan penyakit Diabetes Melitus berdasarkan variabel independen yang telah dipilih dan ditransformasi. Keempat skenario ini akan dievaluasi secara terpisah agar dapat dilihat pengaruh penanganan ketidakseimbangan data terhadap kinerja model. Pada subbab ini, pembahasan dimulai dari skenario data asli sebagai *baseline*.

1. Skenario Pemodelan Data Asli

Skenario ini bertujuan untuk mengetahui performa awal (*baseline*) dari algoritma *Random Forest* dan SVM pada kondisi data yang tidak seimbang. Hasil dari skenario ini akan menjadi pembandingan untuk melihat efektivitas SMOTE. Pendekatan ini dipilih sebagai *baseline* untuk mengetahui performa algoritma dalam kondisi data dunia nyata yang umumnya tidak seimbang, di mana ketidakseimbangan kelas sering kali menjadi tantangan karena model cenderung bias terhadap kelas mayoritas [12].

```
rf = RandomForestClassifier(random_state=RANDOM_STATE)

# Latih langsung dengan parameter default (baseline)
rf.fit(X_train_scaled_imputed, y_train)

print("Random Forest berhasil dilatih dengan parameter default.")
print("Parameter yang digunakan:", rf.get_params())
```

Random Forest berhasil dilatih dengan parameter default.
Parameter yang digunakan: {'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None,

Gambar 9. Pelatihan *Random Forest* Data Asli

Berdasarkan *output* yang dihasilkan, model *Random Forest* dilatih menggunakan *RandomForestClassifier* dengan parameter default (*n_estimators*=100, *max_depth*=None, *min_samples_split*=2, dan parameter lainnya sesuai nilai *default scikit-learn*). Pemilihan parameter *default* bertujuan untuk menjaga kesederhanaan dan reproduksibilitas, serta memberikan gambaran performa awal yang adil.

```
svm = SVC(kernel='rbf', probability=True, random_state=RANDOM_STATE)

# Latih langsung dengan parameter default (baseline)
svm.fit(X_train_scaled_imputed, y_train)

print("SVM berhasil dilatih dengan parameter default.")
print("Parameter yang digunakan:", svm.get_params())
```

SVM berhasil dilatih dengan parameter default.
Parameter yang digunakan: {'C': 1.0, 'break_ties': False, 'cache_size': 200,

Gambar 10. Pelatihan *Support Vector Machine* (SVM) Data Asli

Hasil pelatihan menunjukkan bahwa Model SVM dilatih menggunakan SVC dengan *kernel* RBF dan parameter *default* (*C*=1.0, *gamma*='scale'). Parameter *default* ini memberikan titik awal yang netral untuk perbandingan yang adil dengan *Random Forest*.

2. Skenario Pemodelan dengan SMOTE

```
rf = RandomForestClassifier(random_state=RANDOM_STATE)

# Latih langsung dengan parameter default (baseline) menggunakan data SMOTE
rf.fit(X_train_res, y_train_res)

print("Random Forest berhasil dilatih dengan parameter default.")
print("Parameter yang digunakan:", rf.get_params())
```

Random Forest berhasil dilatih dengan parameter default.
Parameter yang digunakan: {'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None,

Gambar 11. Pelatihan *Random Forest* pada Data Seimbang

Berdasarkan *output* yang dihasilkan, model *Random Forest* berhasil dilatih dengan konfigurasi parameter *default*, antara lain *bootstrap*=True, *max_depth*=None, *min_samples_split*=2, *n_estimators*=100, dan lain-lain. Penggunaan *default* ini cukup efektif karena SMOTE telah memperkaya kelas minoritas, sehingga model *ensemble* dapat mempelajari pola kedua kelas secara lebih seimbang tanpa perlu penyesuaian parameter tambahan.

```
svm = SVC(kernel='rbf', probability=True, random_state=RANDOM_STATE)

# Latih langsung dengan parameter default (baseline) menggunakan data SMOTE
svm.fit(X_train_res, y_train_res)

print("SVM berhasil dilatih dengan parameter default.")
print("Parameter yang digunakan:", svm.get_params())
```

SVM berhasil dilatih dengan parameter default.
Parameter yang digunakan: {'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None,

Gambar 12. Pelatihan *Support Vector Machine* (SVM) pada Data Seimbang

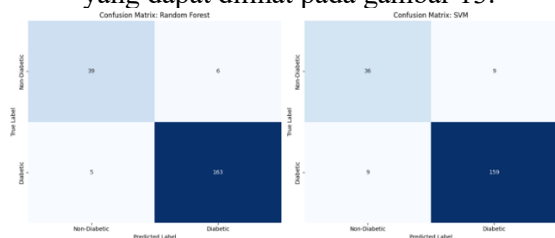
Hasil pelatihan menunjukkan bahwa SVM menggunakan konfigurasi default: *C*=1.0, *kernel*='rbf', *gamma*='scale', dan lain-lain. Penerapan SMOTE membantu SVM dalam memperjelas batas keputusan (*decision boundary*) antara kelas mayoritas dan minoritas, sehingga model dapat bekerja lebih optimal meskipun menggunakan parameter default.

4.5 Evaluation

Tahap ini merupakan tahap terakhir dari kerangka kerja KDD. Tahap evaluasi bertujuan untuk mengukur kinerja model yang telah dilatih sebelumnya, maka dari itu dilakukan evaluasi performa model terbaik yang telah di-*training* pada data pengujian (*test set*) yang belum pernah dilihat sebelumnya. Evaluasi mencakup metrik utama seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk mengukur kemampuan model dalam memprediksi kelas diabetes melitus (positif) secara akurat.

1. Skenario Pengujian Model Data Asli

Eksperimen ini mengukur sejauh mana model dapat menangani ketidakseimbangan kelas secara inheren tanpa bantuan teknik *oversampling*, dan mengidentifikasi potensi bias model terhadap kelas mayoritas yang dapat mengorbankan sensitivitas dalam mendeteksi kelas minoritas yang justru menjadi fokus utama prediksi [13]. Sebelum mengevaluasi performa model menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, dilakukan analisis menggunakan *Confusion Matrix*. Analisis *Confusion Matrix* memberikan gambaran yang lebih rinci mengenai kemampuan model dalam mengklasifikasikan pasien diabetes dan non-diabetes sebelum dilakukan interpretasi terhadap metrik evaluasi lainnya. Berikut ini adalah diagram visualisasi *Confusion Matrix* (*Heatmap*) yang dapat dilihat pada gambar 13.



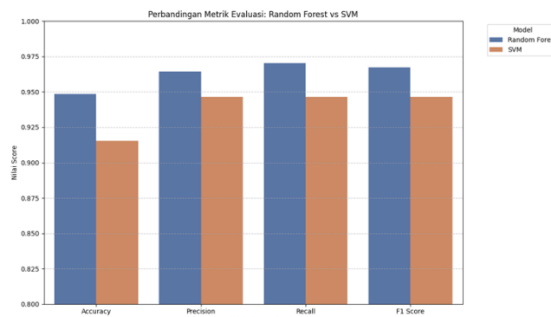
Gambar 13. Visualisasi *Confusion Matrix*

Perbedaan paling signifikan antara kedua model terletak pada jumlah *False Negative* (FN). *Random Forest* hanya menghasilkan 5 FN, sedangkan SVM menghasilkan 9 FN. Dalam konteks diagnosis diabetes, FN merupakan indikator paling kritis karena menunjukkan pasien diabetes yang tidak terdeteksi. Tingginya FN pada SVM disebabkan oleh sifat algoritmanya yang berbasis *support vectors*. Pada data tidak seimbang, jumlah *support vectors* dari kelas minoritas sangat sedikit, sehingga *hyperplane* yang terbentuk cenderung didominasi oleh kelas mayoritas dan mengorbankan sensitivitas terhadap kelas minoritas.

Model	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
<i>Random Forest</i> (Data asli)	0.948	0.964	0.970	0.967
SVM (Data asli)	0.915	0.946	0.946	0.946

Tabel 4. Hasil evaluasi kedua model dengan skenario data asli

Berdasarkan Tabel 4, terlihat bahwa algoritma *Random Forest* secara konsisten mengungguli *Support Vector Machine* (SVM) pada seluruh metrik evaluasi dalam skenario data asli. *Random Forest* mencapai akurasi 94,8%, sementara SVM hanya mencapai 91,5%. Perbedaan yang lebih signifikan terlihat pada metrik recall, di mana *Random Forest* mencatatkan nilai 97,0% dibandingkan SVM yang hanya 94,6%. Tingginya *recall* pada *Random Forest* menunjukkan bahwa model ini jauh lebih baik dalam mendeteksi kasus positif diabetes (meminimalkan *false negative*), yang merupakan prioritas utama dalam diagnosis medis. Demikian pula, nilai *precision* dan *F1-score* *Random Forest* (96,4% dan 96,7%) lebih tinggi dibandingkan SVM (94,6% dan 94,6%). Hasil ini memperkuat temuan bahwa *Random Forest* dengan parameter *default* lebih handal daripada SVM pada *dataset* yang tidak seimbang, tanpa memerlukan teknik penyeimbangan data seperti SMOTE.



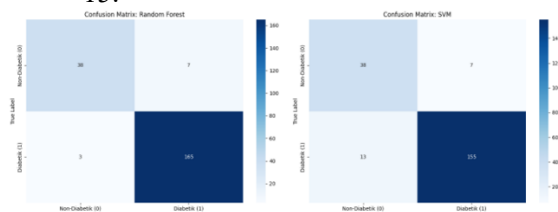
Gambar 14. Visualisasi Perbandingan Metrik Evaluasi Antar Model

Berdasarkan diagram pada Gambar 14, *Random Forest* mendominasi hampir seluruh metrik berbasis klasifikasi. Keunggulan konsisten ini menunjukkan bahwa arsitektur *ensemble learning* pada *Random Forest* mampu menangani sifat *non-linear* data medis lebih baik daripada model *vector* seperti SVM. Menurut [14], *Random Forest* seringkali memberikan hasil yang lebih stabil pada data medis karena kemampuannya merata-ratakan

prediksi dari berbagai *subset* fitur, sehingga mengurangi bias yang mungkin muncul akibat ketidakseimbangan kelas.

2. Skenario Pengujian Model dengan SMOTE

Pada skenario ini, evaluasi dilakukan menggunakan model terbaik yang telah dilatih dengan data yang diseimbangkan melalui teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Pendekatan ini bertujuan untuk mengukur kemampuan generalisasi model terhadap kondisi dunia nyata, di mana prevalensi penderita diabetes jauh lebih kecil dibandingkan populasi non-diabetes. Analisis diawali dengan melihat distribusi kesalahan prediksi secara konkret guna mengevaluasi risiko klinis dari setiap model. Berikut diagram *heatmap* yang dapat dilihat pada Gambar 15.



Gambar 15. Visualisasi Confusion Matrix (Heatmap)

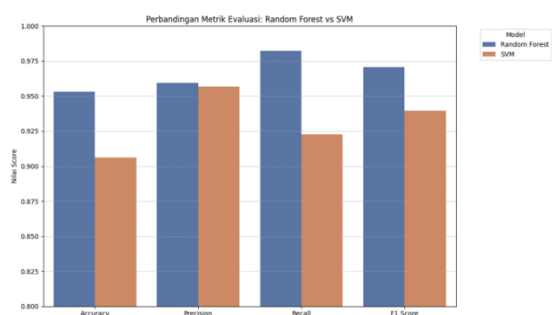
Pada *Random Forest*, SMOTE berhasil menurunkan FN dari 5 menjadi 3 (penurunan 40%). Hal ini menunjukkan bahwa sampel sintetis yang dihasilkan SMOTE efektif dalam memperkaya representasi kelas minoritas tanpa mengganggu kinerja model ensemble. Temuan ini sejalan dengan penelitian [12] bahwa penyeimbangan data pada model *ensemble* seperti *Random Forest* dapat meningkatkan reliabilitas prediksi. Pada SVM, SMOTE justru meningkatkan FN dari 9 menjadi 13 (kenaikan 44%). Hal ini disebabkan oleh cara kerja SMOTE yang menghasilkan sampel sintetis melalui interpolasi antar tetangga terdekat. Sampel sintetis ini dapat memperbesar *overlap* antar kelas atau menciptakan *noise* yang mempersulit SVM dalam menemukan *hyperplane* optimal. SVM yang bergantung pada *support vectors* menjadi

sensitif terhadap perubahan distribusi data akibat penambahan sampel sintetis.

Tabel 5. Hasil evaluasi kedua model dengan skenario dengan SMOTE

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.953	0.959	0.982	0.970
SVM	0.906	0.956	0.922	0.939

Berdasarkan Tabel 5, algoritma *Random Forest* secara konsisten mengungguli *Support Vector Machine* (SVM) pada seluruh metrik evaluasi dalam skenario dengan SMOTE. *Random Forest* mencapai akurasi 95,3%, sementara SVM hanya mencapai 90,6%. Perbedaan paling signifikan terlihat pada metrik *recall*, di mana *Random Forest* mencatatkan nilai 98,2% dibandingkan SVM yang hanya 92,2%. Tingginya *recall* pada *Random Forest* menunjukkan bahwa model ini sangat baik dalam mendeteksi kasus positif diabetes, dengan tingkat kegagalan deteksi (*missed diagnosis*) hanya sekitar 1,8%. Sementara itu, SVM memiliki tingkat kegagalan deteksi sekitar 7,8%, lebih dari empat kali lipat lebih tinggi. Nilai *F1-score Random Forest* (97,0%) juga jauh lebih tinggi dibandingkan SVM (93,9%), menegaskan keseimbangan yang lebih baik antara *precision* dan *recall*. Hasil ini memperkuat temuan bahwa *Random Forest* dengan parameter *default* tetap menjadi model yang lebih handal dibandingkan SVM, baik pada skenario data asli maupun dengan SMOTE. Visualisasi agregat disajikan untuk merangkum performa seluruh model setelah intervensi SMOTE. Hasil dari cuplikan kode tersebut berupa diagram yang dapat dilihat pada Gambar 16 berikut ini.



Gambar 16. Visualisasi Perbandingan Metrik Antar Model

Diagram batang pada Gambar 16 memperlihatkan dominasi *Random Forest* di seluruh metrik utama. Keunggulan ini membuktikan bahwa *Random Forest* adalah algoritma yang paling optimal untuk menangani data medis yang telah diseimbangkan. Hasil ini konsisten dengan temuan [15] bahwa metode *ensemble* cenderung memberikan performa yang lebih unggul dan seimbang dalam tugas klasifikasi penyakit kronis.

4.6 Pembahasan

Tabel 6. Tabel Hasil Pengujian Performa Model Klasifikasi

Model	Skenario	Accuracy	Precision	Recall	F1-Score
Random Forest	Data asli	0.948	0.964	0.970	0.967
Random Forest	Dengan SMOTE	0.953	0.959	0.982	0.970
SVM	Data asli	0.915	0.946	0.946	0.946
SVM	Dengan SMOTE	0.906	0.956	0.922	0.939

Analisis mendalam terhadap Tabel 6 mengungkap bahwa algoritma *Random Forest* (RF) menyajikan dominasi performa yang telak atas *Support Vector Machine* (SVM) di seluruh skenario eksperimen. Pada kondisi data asli tanpa SMOTE, RF mampu mendaki nilai *F1-score* hingga 96,7% dan Akurasi 94,8%. Keberhasilan ini membuktikan secara teoretis bahwa mekanisme *ensemble learning* berbasis

bagging (penggabungan keputusan dari ratusan pohon acak) sangat adaptif dan stabil dalam mengisolasi pola sebaran data tabular medis, meskipun dihadapkan pada ketidakseimbangan distribusi kelas awal[16].

Fenomena ilmiah yang paling menarik terlihat pada respon kedua algoritma terhadap suntikan sampel sintetis SMOTE. Penerapan SMOTE pada data latih sukses mendongkrak performa RF secara minor, di mana akurasi naik menjadi 95,3% dan skor F1 menyentuh angka 97,0%. Hal ini terjadi karena sampel sintetis baru hasil interpolasi SMOTE berhasil memperluas wilayah keputusan (*feature space*) bagi kelas minoritas, sehingga menuntun pohon keputusan untuk membangun percabangan simpul pembagi yang lebih presisi.

Sebaliknya, penerapan SMOTE justru memberikan efek destruktif bagi algoritma SVM, di mana tingkat akurasi merosot dari 91,5% menjadi 90,6%, didampangi penurunan *F1-score* menjadi 93,9%. Penurunan performa ini terjadi karena karakteristik matematika dari SMOTE yang menghasilkan data buatan di sepanjang garis antar tetangga terdekat, sehingga berisiko menciptakan titik data baru yang tumpang tindih (*overlapping classes*) di area perbatasan kelas. Bagi SVM yang sangat bergantung pada kejelasan batas luar objek (margin) untuk merumuskan posisi *hyperplane*, kemunculan data sintetis yang bias di area perbatasan ini justru mengaburkan posisi *support vectors*, mempersempit jarak margin optimal, dan memicu terjadinya kesalahan klasifikasi pada data uji.

Tabel 7. Persentase peningkatan data asli dan data SMOTE pada kedua model

Model	Metrik	Data Asli	Data SMOTE	Persentase Peningkatan (%)
Random Forest	Accuracy	0.948	0.953	0.495
Random Forest	Precision	0.964	0.959	-0.538
Random Forest	Recall	0.970	0.982	1.226
Random Forest	F1-score	0.967	0.970	0.333

SVM	<i>Accuracy</i>	0.915	0.906	-1.025
SVM	<i>Precision</i>	0.946	0.956	1.094
SVM	<i>Recall</i>	0.946	0.922	-2.515
SVM	<i>F1-score</i>	0.946	0.939	-0.743

Berdasarkan hasil perhitungan Tabel 7 di atas, dampak penerapan SMOTE dapat dianalisis secara lebih rinci. Pada *Random Forest*, SMOTE memberikan dampak positif secara keseluruhan. *Recall* mengalami peningkatan paling signifikan sebesar 1,23% (dari 0,970 menjadi 0,982), yang menunjukkan kemampuan model dalam mendeteksi kasus diabetes positif (kelas minoritas) menjadi lebih baik. *Accuracy* dan *F1-score* juga mengalami peningkatan meskipun kecil (masing-masing 0,50% dan 0,33%). *Precision* mengalami penurunan minor sebesar 0,54%, namun tetap berada pada level yang sangat tinggi.

Sebaliknya, pada SVM, penerapan SMOTE cenderung menurunkan performa secara keseluruhan. *Accuracy* turun sebesar 1,03%, *Recall* turun cukup signifikan sebesar 2,52%, dan *F1-score* turun sebesar 0,74%. Meskipun *Precision* mengalami peningkatan sebesar 1,09%, peningkatan ini tidak mampu mengimbangi penurunan pada metrik-metrik lain.

Jika dikomparasikan dengan studi terdahulu, temuan penelitian saat ini berhasil mereplikasi sekaligus memvalidasi kesimpulan dari penelitian Agusti dkk. (2025), Kurniawan & Megawaty (2025), dan Fallo & Nay (2024) yang menempatkan RF sebagai jawara prediksi diabetes. Penelitian ini sekaligus membantah klaim dari Alrasyid dkk. (2024) yang menyatakan SVM lebih unggul. Hasil ini membuktikan bahwa apabila tahapan pra-pemrosesan distandarkan dan teknik penyeimbangan kelas diisolasi secara ketat agar tidak terjadi kebocoran data, *Random Forest* terbukti memiliki ketahanan struktural yang lebih kokoh dibandingkan SVM berbasis *kernel* RBF standar untuk kasus diabetes.

5. KESIMPULAN

Berdasarkan hasil eksperimen *controlled comparative study* yang telah dilaksanakan,

dapat ditarik beberapa poin kesimpulan sebagai berikut:

- Algoritma *Random Forest* menyajikan performa terbaik dalam memprediksi penyakit Diabetes Melitus tipe-2 pada dataset Bangladesh 2025, dengan puncaknya menembus nilai *Accuracy* sebesar 95,3% dan *F1-score* mencapai 97,0%.
- Intervensi teknik penyeimbangan kelas menggunakan SMOTE pada data latih memberikan dampak yang bertolak belakang; mampu meningkatkan akurasi *Random Forest* secara minor, namun justru memperburuk performa *Support Vector Machine* akibat distorsi data sintesis yang mengaburkan batas keputusan *hyperplane*.
- Model *Random Forest*, baik yang dikombinasikan dengan SMOTE maupun tanpa SMOTE, sangat direkomendasikan untuk diadopsi menjadi mesin penggerak utama dalam perancangan aplikasi Sistem Pendukung Keputusan (SPK) deteksi dini risiko diabetes melitus di fasilitas kesehatan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Ibu Dr. Sofi Defiyanti, M.Kom. dan Ibu Rini Mayasari, M.Kom. selaku dosen pembimbing yang telah memberikan arahan dan bimbingan selama penelitian berlangsung. Terimakasih juga kepada pihak-pihak yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- International Diabetes Federation, "IDF Diabetes Atlas (11th ed.)," 2025. Accessed: Dec. 07, 2025. [Online]. Available: <https://diabetesatlas.org/>
- A. Renata Agusti, A. Fauzi, K. Ahmad Baihaqi, and T. Rohana, "Identifikasi Penyakit Diabetes Mellitus Menggunakan Algoritma Support Vector Machine dan Random Forest," *Jurnal Riset Komputer*, vol. 12, no. 4, pp. 2407–389, 2025, doi: 10.30865/jurikom.v12i4.8686.
- M. Fadli Kurniawan and D. Ayu Megawaty, "Comparison of Logistic Regression, Random Forest, Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) Algorithms in Diabetes Prediction," 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>

- [4] S. I. Fallo and A. Nay, "KLASIFIKASI SUPPORT VECTOR MACHINE DAN RANDOM FOREST PADA DATA BIOMEDIS : APLIKASI DALAM ANALISIS DATA PENYAKIT DIABETES CLASSIFICATION OF SUPPORT VECTOR MACHINE AND RANDOM FOREST ON BIOMEDICAL DATA: APPLICATIONS IN DIABETES DISEASE DATA ANALYSIS," 2024.
- [5] H. Alrasyid, A. Homaidi, M. Kom, and Z. Fatah, "Comparison Support Vector Machine and Random Forest Algorithms in Detect Diabetes," 2024.
- [6] M. Y. Bhuiyan, S. S. Ayon, Md. E. Hossain, M. S. U. Miah, and A. Sarower, "Type-2 Diabetes Dataset Bangladesh," Nov. 12, 2025, *Mendeley Data*.
- [7] N. Novitasari, N. D. Nuris, and R. Herdiana, "Penerapan Algoritma K-Means untuk Clustering Data Jumlah Penduduk Miskin Berdasarkan Kota/Kabupaten di Jawabarat menggunakan Rapidminer," *Jurnal Informatika Terpadu*, vol. 9, no. 1, pp. 68–73, Mar. 2023, doi: 10.54914/jit.v9i1.660.
- [8] M. Desiawan and A. Solichin, "SVM Optimization with Grid Search Cross Validation for Improving Accuracy of Schizophrenia Classification Based on EEG Signal," *JURNAL TEKNIK INFORMATIKA*, vol. 17, no. 1, pp. 10–20, May 2024, doi: 10.15408/jti.v17i1.37422.
- [9] F. S. Pratiwi, M. Agung Barata, and A. D. Ardianti, "IMPLEMENTASI METODE SMOTE DAN RANDOM OVER-SAMPLING PADA ALGORITMA MACHINE LEARNING UNTUK PREDIKSI CUSTOMER CHURN DI SEKTOR PERBANKAN," *Jurnal Sistem Informasi dan Informatika (Simika)*, vol. 8, no. 1, 2025, [Online]. Available: <https://www.kaggle.com/datasets/gauravtopre/bank-customer-churn-dataset/data>
- [10] A. R. MOHAMMED, "Enhancing Diabetes Mellitus Onset Prediction through Advanced Ensemble Learning Techniques," *Journal of Statistical Modelling and Analytics*, vol. 6, no. 2, pp. 1–18, Dec. 2024, doi: 10.22452/josma.vol6no2.2.
- [11] M. Talebi Moghaddam *et al.*, "Predicting diabetes in adults: identifying important features in unbalanced data over a 5-year cohort study using machine learning algorithm," *BMC Med. Res. Methodol.*, vol. 24, no. 1, p. 220, Sep. 2024, doi: 10.1186/s12874-024-02341-z.
- [12] Y. Xu and Y. Nie, "Diabetes Prediction Based on Support Vector Machine Model," *Highlights in Science, Engineering and Technology*, vol. 102, pp. 311–320, Jul. 2024, doi: 10.54097/ygdasy74.
- [13] M. Imani, M. Joudaki, A. Bagheri, and H. R. Arabnia, "Why ROC-AUC Is Misleading for Highly Imbalanced Data: In-Depth Evaluation of MCC, F2-Score, H-Measure, and AUC-Based Metrics Across Diverse Classifiers," *Technologies (Basel)*, vol. 14, no. 1, p. 54, Jan. 2026, doi: 10.3390/technologies14010054.
- [14] M. A. Widyananda and I. Palupi, "Implementation of the Spiral Optimization Algorithm in the Support Vector Machine (SVM) Classification Method (Case Study: Diabetes Prediction)," in *2021 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ICADEIS52521.2021.9701953.
- [15] O. Toyin, J.-J. Timothy, S. E. Obamiyi, B. A. O, O. M. Aweh, and A. A. James, "Comparative Analysis of Supervised Machine Learning Algorithms for Diabetes Prediction," *INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH*, vol. 12, no. 08, Aug. 2024, doi: 10.47191/ijmcr/v12i8.05.
- [16] B. R. Prasetyo, E. D. Wahyuni, and P. M. Kusumantara, "KOMPARASI PERFORMA MODEL BERBASIS ALGORITMA RANDOM FOREST DAN LIGHTGBM DALAM MELAKUKAN KLASIFIKASI DIABETES MELITUS GESTASIONAL," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4817.