Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

Penerapan Metode Machine Learning Dan Teknik SMOTE untuk Prediksi Diabetes

Alrayssa Davinka Sembiring Depari, Ken Ditha Tania*, Putri Eka Sevtiyuni

Sistem Informasi, Fakultas Ilmu Komputer, Universitas Sriwijaya, Palembang, Indonesia
Email: [1]rayssadepari@gmail.com, [2]kenya.tania@gmail.com,[3]putrieka@unsri.ac.id
Email Penulis Korespondensi: kenya.tania@gmail.com*
Submitted: **19/08/2025**; Accepted: **01/11/2025**; Published: **31/12/2025**

Abstrak– Diabetes merupakan salah satu penyakit tidak menular yang prevalensinya terus meningkat secara global maupun nasional. Kondisi ini menimbulkan risiko komplikasi serius seperti penyakit jantung, stroke, hingga gagal ginjal apabila tidak terdeteksi sejak dini. Oleh karena itu, dibutuhkan metode prediksi berbasis data yang mampu membantu proses deteksi awal secara cepat, akurat, dan efisien. Penelitian ini bertujuan membandingkan kinerja empat algoritma pembelajaran mesin, yaitu Random Forest, XGBoost, Support Vector Machine (SVM), dan K-Nearest Neighbor (KNN) dalam memprediksi penyakit diabetes menggunakan dataset publik dari Kaggle. Penelitian dilakukan dengan mengacu pada kerangka Knowledge Discovery in Databases (KDD) yang terdiri dari tahapan seleksi data, pra-pemrosesan (data cleaning, transformasi, dan normalisasi), penyeimbangan kelas menggunakan Synthetic Minority Over-sampling Technique (SMOTE), pembagian data latih dan data uji dengan rasio 80:20, implementasi algoritma, serta evaluasi performa model. Evaluasi dilakukan menggunakan metrik Accuracy, Precision, Recall, dan F1-Score untuk memastikan kualitas prediksi secara menyeluruh. Hasil penelitian menunjukkan bahwa Random Forest dan XGBoost memberikan performa terbaik dengan nilai Accuracy, Precision, Recall, dan F1-Score sebesar 0,97. Model KNN menunjukkan performa cukup baik dengan skor 0,94, sementara SVM memperoleh nilai terendah sebesar 0,89. Temuan ini menegaskan bahwa penerapan kerangka KDD dengan teknik SMOTE mampu menghasilkan model prediksi yang optimal. Random Forest dan XGBoost direkomendasikan sebagai algoritma unggulan pada penelitian serupa, terutama pada dataset dengan karakteristik kelas yang tidak seimbang.

Kata Kunci: Knowledge Discovery; Prediksi diabetes; Pembelajaran Mesin; Random Forest; XGBoost; SMOTE

Abstract– Diabetes is one of the non-communicable diseases whose prevalence continues to increase both globally and nationally. This condition poses a risk of serious complications such as heart disease, stroke, and kidney failure if not detected early. Therefore, data-driven prediction methods are needed to support early detection that is fast, accurate, and efficient. This study aims to compare the performance of four machine learning algorithms, namely Random Forest, XGBoost, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN), in predicting diabetes using a public dataset from Kaggle. The research was conducted by adopting the Knowledge Discovery in Databases (KDD) framework, which consists of data selection, preprocessing (data cleaning, transformation, and normalization), class balancing using the Synthetic Minority Over-sampling Technique (SMOTE), splitting the dataset into training and testing data with a ratio of 80:20, algorithm implementation, and model performance evaluation. The evaluation was carried out using Accuracy, Precision, Recall, and F1-Score metrics to ensure comprehensive prediction quality. The results show that Random Forest and XGBoost achieved the best performance, with Accuracy, Precision, Recall, and F1-Score values of 0.97. The KNN model demonstrated fairly good performance with a score of 0.94, while SVM obtained the lowest score of 0.89. These findings confirm that applying the KDD framework with the SMOTE technique can produce an optimal prediction model. Random Forest and XGBoost are recommended as leading algorithms for similar studies, particularly for datasets with imbalanced class characteristics.

Keywords: Knowledge Discovery; Diabetese prediction; Pembelajaran Mesin; Random Forest; XGBoost; SMOTE

1. PENDAHULUAN

Penyakit Tidak Menular (PTM) merupakan kelompok penyakit yang tidak dapat ditularkan dari satu manusia ke manusia lainnya. Diabetes merupakan salah satu dari 5 jenis Penyakit Tidak Menular (PTM). Diabetes adalah penyakit dimana metabolisme pada manusia terganggu sehingga menyebabkan hormon insulin tidak dapat digunakan secara efektif [1]. Menurut *International Diabetes Federation* (IDF), dilaporkan bahwa pada tahun 2025 populasi dari umur 20 – 79 tahun hidup dengan diabetes, dan lebih dari 4 – 10 orang tidak menyadari bahwa mereka menderita diabetes.[2]. Banyak faktor penyebab keterlambatan deteksi penyakit diabetes, seperti rendahnya kesadaran mereka terkait diabetes sehingga keterbatasan pada fasilitas kesehatan di daerah tertentu.
Diabetes tergolong penyakit serius yang menimbulkan bahaya apabila tidak memperoleh penanganan secara cepat. Kondisi yang tidak terkontrol dapat mengakibatkan dampak merugikan bagi tubuh. Gangguan tersebut dapat muncul dalam bentuk masalah metabolik akut, seperti *hipoglikemia* maupun krisis *hiperglikemia*. Selain itu, diabetes juga berpotensi menimbulkan komplikasi jangka panjang pada sistem kardiovaskular, antara lain penyakit jantung, *stroke*, *dislipidemia*, penyakit pembuluh darah perifer, hipertensi, serta kerusakan pembuluh darah kecil (*mikroangiopati*) maupun besar (*makroangiopati*) [3] Maka dari itu, diperlukan metode prediksi yang lebih cepat, akurat, dan efisien untuk menghindari terjadinya keterlambatan deteksi diabetes.
Penelitian ini menggunakan beberapa model *machine learning* untuk memprediksi penyakit diabetes berdasarkan data yang memiliki 7 atribut faktor risiko diabetes, dimana dapat membantu dalam prediksi diabetes. Metode yang gunakan adalah metode *Random Forest*, *XGBoost*, *Support Vector Machine* dan, *K-Nearest Neighbor*. Keempat

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

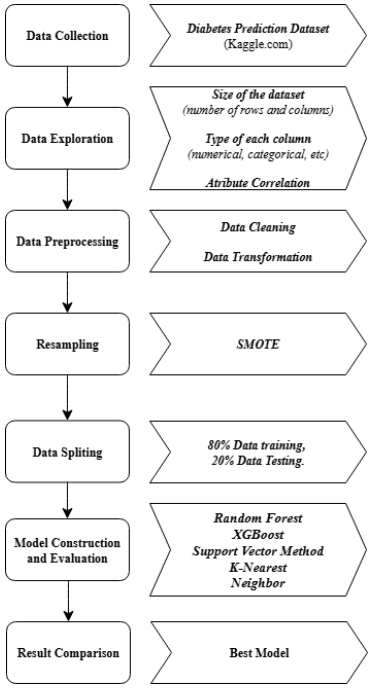
metode tersebut dipilih karena mereka merupakan metode prediksi yang sering digunakan dalam penelitian prediksi serta. Tiap metode juga memiliki keunikan nya masing-masing serta memiliki kemampuan dalam penanganan data yang tidak seimbang. *Random Forest* adalah metode yang menggabungkan banyak pohon keputusan (*decision tree*) dan menggunakan ukuran seperti entropi informasi untuk membedakan data, sehingga dapat meningkatkan akurasi dalam memprediksi hasil.[4] *XGBoost* adalah algoritma *machine learning* berbasis *gradient boosting* yang unggul dalam akurasi, efisiensi, dan pencegahan *overfitting* [5] *Support Vector Machine* (SVM) dipilih karena kemampuannya yang unggul dalam menangani dataset berdimensi tinggi dan tidak seimbang.[6] *K-Nearest Neighbor* (KNN) merupakan metode yang mengklasifikasi data berdasarkan kategori yang telah ditentukan dengan mempertimbangkan jarak antara data.[7]. Hasil dari keempat metode tersebut akan dibandingkan untuk melihat hasil prediksi yang lebih efektif. Dari hasil ini juga memberikan wawasan serta rekomendasi dalam pemilihan metode yang paling sesuai dalam melakukan prediksi penyakit diabetes.

Penelitian sejenis telah dilakukan oleh beberapa peneliti. Seperti yang dilakukan oleh Agus Ambarwari, Qadhil Jafar Adrian, dan Yeni Herdiyeni pada tahun 2020 dimana melakukan penelitian prediksi diabetes dengan membandingkan 6 metode machine learning, yaitu *Multi-Layer Perceptron, Decision Tree, AdaBoost, Random Forest, XGBoost,* dan *LightGBM* yang membuktikan bahwa metode *XGBoost* menghasilkan nilai paling besar sebesar 87.97% [8]. Pada Penelitian berikutnya dilakukan oleh Rendi Risqi Pradana dan Yani Parti Astuti pada tahun 2025, dimana mereka membandingkan kinerja metode *Naive Bayes* dan *Random Forest* dalam klasifikasi penyakit diabetes melitus dan menghasilkan bahwa hasil *Random Forest* lebih unggul dengan hasil sebesar 79,5% [9]. Berikutnya penelitian oleh Amri, Rodi, Wathani, Bagja, dan Zulkipli pada tahun 2025 menggunakan algoritmas *K-Nearest Neighbor* dengan teknik *SMOTE* yang menghasilkan akurasi sebesar 96% [10]. Penelitian berikutnya ialah oleh Hastono, Vitianingsih, Pamudi, Maukar, dan Wati pada tahun 2025, melakukan penelitian prediksi diabetes melitus dengan membandingkan metode yang dimana salah satunya ialah metode *Support Vector Machine (SVM)*, dimana metode *Support Vector Method* menghasilkan akurasi sebesar 77,24%. [11].Dari studi-studi tersebut, tujuan utama dari penelitian ini adalah menguji serta membandingkan kinerja dari keempat model machine learning seperti *Random Forest, XGBoost, Support Vector Method,* dan *K-Nearest Neighbor* untuk menghasilkan akurasi tertinggi yang dapat dijadikan bahan rekomendasi pada prediksi diabetes dengan penggunaan dataset publik dari Kaggle. Selain penerapan machine learning terhadap prediksi diabetes, diharapkan juga adanya penggunaan kerangka *Knowledge Discovery in Database* (KDD). KDD merupakan serangkaian tahapan sistematis untuk mengubah data mentah menjadi pengetahuan yang bermanfaat, meliputi seleksi data, pra-pemrosesan, transformasi, *data mining*, hingga interpretasi hasil. *Knowledge discovery* menjadi elemen penting untuk menemukan pola tersembunyi yang dapat mendukung pengambilan keputusan berbasis data. Dengan memanfaatkan konsep KDD, penelitian ini berfokus pada penggalian informasi dari data pasien yang memiliki atribut risiko diabetes, sehingga dihasilkan pola dan hubungan antar-atribut yang dapat digunakan sebagai dasar pengambilan keputusan [12].

2. METODOLOGI PENELITIAN

Terdapat beberapa tahap yang dilakukan untuk mencapai tujuan penelitian, di mana alurnya disusun dengan mengacu pada tahapan yang digunakan dalam penelitian oleh Novalia, Tania, Meiriza dan Wedhasmara pada tahun 2024 dan Sofiah, Tania, Meiriza dan Wedhasmara pada tahun 2024 [13]. Gambar 1 memperlihatkan visualisasi setiap tahapan tersebut, mulai dari proses awal hingga akhir penelitian ini.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032



Gambar 1. Tahap Penelitian. Diadaptasi dari [14][13]

Metodologi yang digunakan dalam penelitian ini mengacu pada tahapan *Knowledge Discovery in Databases (KDD)* sebagaimana diterapkan pula dalam penelitian sebelumnya yang dilakukan oleh Novalia, Tania, Meiriza dan Wedhasmara pada tahun 2024 dan Sofiyah, Tania, Meiriza dan Wedhasmara pada tahun 2024, di mana proses *knowledge discovery* diawali dengan pemilihan dataset relevan (*data selection*), dilanjutkan dengan pra-pemrosesan (*preprocessing*) seperti pembersihan data, normalisasi, dan transformasi [14]. Tahap pertama pada penelitian ini adalah Pengumpulan Data (*Data Collection*) dan Eksplorasi Data (*Data Exploration*) untuk memahami karakteristik dan korelasi atribut pada dataset. Setelah itu dilakukan Pra-pemrosesan Data (Preprocessing Data) yang mencakup Pembersihan Data (*Cleaning Data*) dan Pengubahan Data (*Data Transformation*), termasuk proses transformasi untuk mempersiapkan format input yang sesuai bagi algoritma *machine learning*. Selanjutnya, dilakukan *Resampling* menggunakan teknik SMOTE untuk mengatasi ketidakseimbangan kelas, kemudian pembagian data (*split data*) dengan rasio 80:20. Proses berikutnya adalah *Data Mining*, yaitu penerapan algoritma *Random Forest, XGBoost, Support Vector Machine (SVM), dan K-Nearest Neighbor (KNN)* untuk mengekstraksi pola. Tahap terakhir adalah Interpretasi dan Evaluasi, sebagaimana disarankan pada penelitian oleh Metti Detricia Pratiwi dan Ken Ditha Tania pada tahun 2025, yaitu menilai kinerja model berdasarkan metrik evaluasi dan melakukan Perbandingan Hasil (*Result Comparison*) untuk menentukan model terbaik yang dapat dijadikan rekomendasi prediksi diabetes [12].

2.1 Dataset

Pada penelitian ini, dataset yang digunakan ini berisi data pasien yang memiliki faktor – faktor diabetes yang digunakan sebagai atribut, yaitu *gender* (kelamin), *age* (umur), *hypertension* (hipertensi), *heart_disease* (Penyakit Jantung), *smoking_history* (Riwayat Merokok), *Body Mass Index* (Indeks Masa Tubuh), *HbA1c_level* (Level Hemoglobin A1c), *blood_glucose_level* (Level Gula Darah), dan diabetes.
Pada Penelitian ini, dataset yang digunakan memiliki data sebanyak 100.000 data dari https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data. Namun, peneliti hanya mengambil data dengan ketentuan rentang usia dari 17 – 80 tahun, sehingga totalnya adalah 80.437 data. Tabel 1 menunjukkan dataset yang digunakan dalam penelitian ini.

Tabel 1. Tabel Dataset

	gender	age	*hypertension*	*heart_ disease*	*smoking_ history*	bmi	*HbA1c_ level*	*blood_glucose_ level*	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
…	…	…	…	…	…	…	…	…	…
99998	Female	24.0	0	0	never	35.42	4.0	100	0
99999	Female	29.0	0	0	current	22.43	6.6	90	0

2.2 Pra-Pemrosesan Data

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

Pra-Pemprosesan Data merupakan sekumpulan tahap atau teknik yang dilakukan pada sebuah data sebelum diolah sebelum dilakukannya analisis atau pemodelan [15]. Terdapat 2 tahap yang dilalui pada Pra – Pemrosesan Data ini, yaitu *Data Cleaning* dan Data *Transformation.*

2.2.1 *Data Cleaning*

Data Cleaning merupakan tahap pertama yang berfungsi untuk pemeriksaan terhadap nilai null atau kosong pada seluruh kolom data. Hal ini penting dilakukan untuk menjaga keakuratan model dan menghindari bias selama analisis. Proses ini dilakukan agar data yang digunakan bersih dan representatif saat dianalisis. [16]

2.2.2 Data Transformation

Tahap kedua adalah Transformasi Data, dimana kita mengubah format data yaang disesuaikan dengan kebutuhan analisis. Variabel kategorikal seperti *gender* dan *smoking_history* disesuaikan menjadi bentuk numerikal. Berikutnya, variabel *smoking_history* dilakukan normalisasi menggunakan metode *Min-Max Scalling* agar mendapatkan jarak nilai yang seragam antara 0 hingga 1 [17]. Untuk menghitung *Min – Max Scalling* dapat menggunakan persamaan (1)

$$\boldsymbol{X\_normalisasi} = \frac{(\mathrm{X}-\mathrm{Xmin})}{(Xmax-Xmin)} \quad (1)$$

Ada Ada pula penambahan variabel baru yaitu kategori umur berdasarkan rentang usia yang dikelompokkan menjadi remaja, dewasa, dan lanjut usia. Transformasi data ini penting dilakukan agar proses pengelompokan saat pemodelan nanti berjalan dengan lancar.[18]

2.3 *Resampling Data*

Adanya ketidakseimbangan pada jumlah data di tiap variabel seperti pada variabel diabetes dan *smoking_history* bisa memengaruhi hasil prediksi pada model. Hal ini juga akan membuat model cenderung bias terhadap kelas mayoritas dan me nurunkan akurasi terhadap kelas minoritas. Maka dari itu, peneliti memutuskan menggunakan teknik *Resampling* dalam mengatasi masalah tersebut. Peneliti memilih untuk menggunakan teknik *Resampling* dengan metode *Synthetic Minority Over-sampling Technique* (SMOTE).

2.3.1 *SMOTE*

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu algoritma preprocessing yang paling sering dimanfaatkan dalam penanganan permasalahan data tidak seimbang [19]. Metode *SMOTE* memodifikasi data yang tidak seimbang dengan menghasilkan sampel sintetis dari kelompok minoritas, sehingga mampu meningkatkan efektivitas proses klasifikasi[20].

2.4 *Splitting Data*

Di tahap ini, variabel atribut (X) dipisahkan dengan variabel target (Y). Variabel atribut terdiri dari semua kolom dalam *dataset* kecuali kolom diabetes yang digunakan pada variabel target (Y). Kolom diabetes digunakan sebagai variabel target untuk memberikan kategori apakah orang tersebut mengidap diabetes (1) atau tidak (0).
Kemudian dataset dibagi menjadi dua kelompok, yaitu data latih (*data testing*) dan data uji (*data trial*). Peneliti memutuskan untuk menggunakan rasio 80:20. Rasio tersebut sudah digunakan oleh beberapa penelitian prediksi diabetes menggunakan metode pemodelan *machine learning*. Seperti penelitian Brahmandjati, Rahim, dan Asharudin pada tahun 2024 yang menggunakan rasio 80:20 pada prediksi Diabetes dengan menggunakan algoritma *XGBoost* [5]. Ada pula penelitian yang dilakukan oleh Farisi dan Homaidi pada tahun 2025, yang menggunakan rasio 80:20 di penelitian prediksi Diabetes dengan metode *Support Vector Machine (SVM)* [6]. Oleh karena itu, penulis memutuskan untuk menggunakan rasio tersebut sebagai pendekatan yang optimal.

2.5 Pembangunan Model

Tahap berikut ini merupakan tahap pembangunan model deteksi pengidap diabetes dengan memnggunakan metode algoritma machine leanring. Algoritma yang digunakan adalah *Random Forest, XGBoost, Support Vector Machine* dan, *K-Nearest Neighbor*. Metode-metode tersebut memiliki keunikannya masing-masing dalam menangani masalah klasifikas dengan data yang tidak seimbang, dimana sangat cocok dengan penelitian ini.

2.5.1 Random Forest

Random Forest adalah salah satu algoritma *machine learning* yang termasuk dalam kelompok *ensemble learning*, yang bekerja dengan membentuk sekumpulan pohon keputusan (*decision tree*) dan menggabungkan hasil prediksinya untuk meningkatkan akurasi dan kestabilan model. Model ini cukup unggul dikarenakan pada penelitian Setiawan pada tahun 2025, *Random Forest* mampu menangani data dengan *noise* lebih baik dibanding algoritma lainnya yang digunakannya [21].

2.5.2 XGBoost

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

XGBoost adalah algoritma *ensemble learning* berbasis *gradient boosting* yang dirancang untuk efisiensi dan kinerja tinggi. Keunggulan dari metode ini ialah membentuk model prediktif secara bertahap dari sejumlah pohon keputusan dan mengoptimalkan kesalahan model sebelumnya. Model ini juga dapat menghasilkan akurasi yang tinggi dan pelatihan yang lebih efisien terutama dalam menangani fituer yang kompleks dan jumlah data yang besar[22]

2.5.3 Support Vector Method (SVM)

Support Vector Method (SVM) adalah algoritma klasifikasi yang bekerja dengan mencari *hyperplane* terbaik yang memisahkan kelas-kelas data. Dalam memprediksi suatu kelas pada data, *Support Vector Method* akan memberikan nilai berdasarkan daerah kelas mana yang merupakan tempat dari data tersebut.[23]

2.5.4 K-Nearest Neighbor

K-Nearest Neighbor adalah algoritma klasifikasi yang menentukan kelas suatu data berdasarkan mayoritas label dari sejumlah tetangga terdekatnya. *K-Nearest Neighbor* bersifat non-parametrik dan sederhana, namun efektif untuk dataset kecil dan tidak terlalu kompleks.[21]

2.6 Evaluasi Model

Tahap berikutnya adalah mengevaluasi hasil dari seluruh model dalam deteksi diabetes dengan menggunakan dataset tersebut. Hasil ini dapat memberikan pemahaman yang lebih akurat terkait kemampuan deteksi pada tiap model. Berikut beberapa evaluasi yang digunakan dalam penelitian ini meliputi :

a. *Confusion Matrix*: merupakan tabel yang berfungsi untuk mendeskripsikan kinerja model klasifikasi dengan membandingkan prediksi model terhadap label sebenarnya. Tabel ini terdiri dari empat komponen utama yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN)[24] Bentuk dari *Confusion Matrix* dapat dilihat pada Tabel 2.

Tabel 2. *Confussion Matrix*

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

b. *Accuracy* : merupakan tolak ukur proporsi prediksi yang benar terhadap total prediksi. Akurasi mengukur sejauh mana model dapat mengklasifikasikan data dengan benar secara keseluruhan[25]. Untuk menghitung *accuracy*, dapat menggunakan formula pada persamaan (2).

$$\boldsymbol{Accuracy} = \frac{\text{TP+TN}}{TP+TN+FP+FN} \quad (2)$$

c. *Precision* : merupakan tolak ukur sejauh mana prediksi positif yang benar[25]. Untuk menghitung precision, dapat menggunakan formula pada persamaan (3)

$$\boldsymbol{Precision} = \frac{\text{TP}}{TP+FP} \quad (3)$$

d. *Recall* : digunakan untuk mengukur kemampuan hasil analisis terhadap model saat mendeteksi seluruh kasus positif sebenarnya[24]. Untuk menghitung recall, dapat menggunakan formula pada persamaan (4).

$$\boldsymbol{Recall} = \frac{\text{TP}}{TP+FN} \quad (4)$$

e. F1-Score : merupakan rata – rata yang seimbang antara precision dan recall. Metrik ini berguna untuk mengevaluasi model ketika terdapat ketidakseimbangan antara kelas positif dan negatif [24]. Untuk menghitung F1-Score, dapat menggunakan formula pada persamaan (5)

$$\boldsymbol{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision \times Recall} \quad (5)$$

Setelah menghitung semua metrik – metrik tersebut. Hasil evaluasi pada tiap model akan menentukan yang mana yang paling efektif dalam memprediksi diabetes. Diharapkan dengan metode ini dapat mengidentifikasi model yang memiliki kinerja terbaik yang dapat diandalkan dalam prediksi deteksi diabetes.

3. HASIL DAN PEMBAHASAN

Penelitian ini tentu melalui beberapa tahap sebelum memulai mengolah dataset. Tiap tahapan pengolahan serta analisis pada data akan didokumentasikan dengan menggunakan visualisasi yang menunjukkan perubahan pada data di setiap prosesnya.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

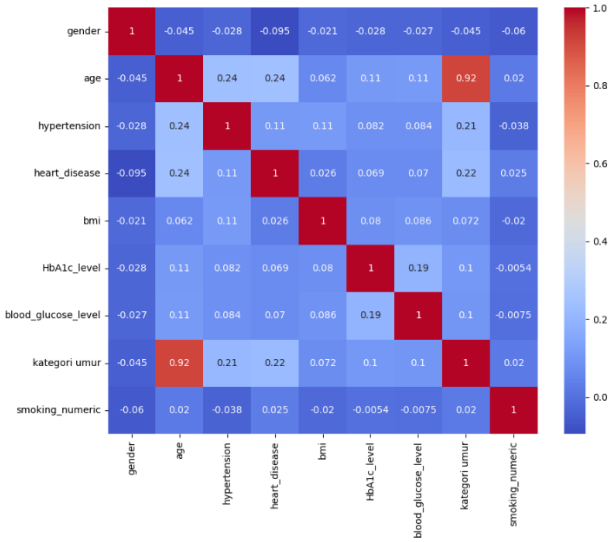
3.1 Pemahaman Data Dan Analisis Korelasi

Dataset yang digunakan berisi variabel *gender*, *hypertension*, *heart_disease*, *bmi*, *HbA1c_level*, *blood_glucose_level*, kategori umur, dan *smoking_numeric* yang digunakan untuk memprediksi kondisi target. Untuk mengetahui hubungan antar variabel, dilakukan analisis korelasi menggunakan metode *Pearson* menggunakan kode seperti pada Gambar 2 dan divisualisasikan dengan *heatmap* pada Gambar 3.

```
#visualisasi korelasi
fig, ax = plt.subplots(figsize=(10, 8))
sns.heatmap(X.corr(), annot=True, cmap='coolwarm', ax=ax)
plt.show()
```

Gambar 2. Kode Hasil *Heatmap*

Visualisasi korelasi pada Gambar 3 dilakukan untuk menganalisis hubungan antaratribut pada dataset. Perhitungan nilai korelasi dilakukan menggunakan fungsi .corr() dari pandas, yang menghasilkan matriks korelasi antarfitur numerik. Matriks tersebut kemudian divisualisasikan dalam bentuk heatmap dengan bantuan seaborn, dimana nilai korelasi ditampilkan langsung pada setiap sel melalui parameter *annot=True.* Skema warna *coolwarm* digunakan untuk memudahkan pembacaan intensitas hubungan, dengan warna merah menunjukkan korelasi negatif dan biru menunjukkan korelasi positif. Visualisasi ini bertujuan untuk membantu dalam mengidentifikasi atribut-atribut yang memiliki hubungan kuat, baik positif maupun negatif, sehingga dapat dipertimbangkan dalam pemilihan fitur pada tahap pemodelan.



Gambar 3. *Heatmap* Analisis Korelasi

Berdasarkan Gambar 2, sebagian besar pasangan variabel memiliki korelasi yang rendah (< 0,3), menunjukkan bahwa fitur-fitur bersifat relatif independen dan tidak memiliki multikolinearitas tinggi. Korelasi tertinggi ditemukan antara kategori umur dan *hypertension* (0,21) serta kategori umur dan *heart_disease* (0,22), yang logis mengingat prevalensi kedua kondisi tersebut meningkat seiring bertambahnya usia. Sementara itu, variabel bmi, HbA1c_level, dan *blood_glucose_level* memiliki korelasi rendah satu sama lain, yang berarti masing-masing memberikan kontribusi informasi yang berbeda pada model.

3.2 Hasil Pra-Pemrosesan data

Pada tahap ini, data melakukan 2 jenis tahap, yaitu *Data Cleaning* serta *Data Transformation*. *Data Cleaning* ialah tahap dimana adanya pemastian dimana data tidak memiliki data yang kosong (*null*) dan itidak memiliki duplikat. Pada data ini tidak memiliki data dengan nilai yang hilang. Saat dilakukan pengecekan terkait duplikasi data, terdeteksi dataset memiliki sebesar 3854 data duplikat, menyisahkan jumlah data sebesar 96.146 data. Kemudian setelah mengfilter dimana data yang digunakan ialah data yang memiliki umur dengan rentang 17 – 80 tahun, tersisalah data sebanyak 80.437 data.

Berikutnya adalah tahap *Data Transformation*, dimana pada proses ini dilakukannya pengubahan nilai pada beberapa data menjadi data numerik. Adapun penggunaan normalisasi *Minimum – Maximum Scale* pada atribut smoking_history untuk mendapatkan jarak nilai yang sama pada data. Hasil dari normalisasi pada data dapat dilihat pada Gambar 3 dan Tabel 3.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

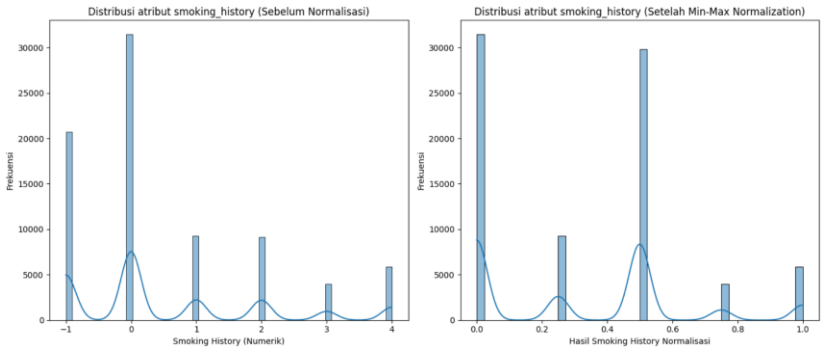
Tabel 3. Hasil *Minimum – Maximum Scale* Pada Atribut *Smoking_History*

	Smoking_history	***Smoking_numeric***	***Smoking_normalized***
1	*Never*	0	0.0
2	*No info*	2	0.5
…	…	…	…
99998	*Never*	0	0.0
99999	*Current*	2	0.5

```
import matplotlib.pyplot as plt
import seaborn as sns

# Create a temporary dataframe to visualize the 'smoking_numeric' before replacing -1
temp_df = df.copy()
temp_df['smoking_numeric_original'] = temp_df['smoking_history'].map(mapping)

# Visualisasi hasil sebelum dan sesudah normalisasi Min-Max pada 'smoking_history'
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plot sebelum normalisasi (using the 'smoking_numeric_original' with -1 for 'No Info')
sns.histplot(temp_df['smoking_numeric_original'].dropna(), kde=True, ax=axes[0])
axes[0].set_title('Distribusi atribut smoking_history (Sebelum Normalisasi)')
axes[0].set_xlabel('Smoking History (Numerik)')
axes[0].set_ylabel('Frekuensi')

# Plot sesudah normalisasi
sns.histplot(df1['smoking_normalized'], kde=True, ax=axes[1])
axes[1].set_title('Distribusi atribut smoking_history (Setelah Min-Max Normalization)')
axes[1].set_xlabel('Hasil Smoking History Normalisasi')
axes[1].set_ylabel('Frekuensi')

plt.tight_layout()
plt.show()
```

Gambar 4. Coding Menghasilkan Diagram Batang Perbandingan

Pada Gambar 4, bertujuan untuk memvisualisasikan distribusi data atribut *smoking_history* sebelum dan sesudah dilakukan normalisasi menggunakan *Min-Max Scaling*. Pertama, dibuat salinan dari *DataFrame* df ke dalam *temp_df*, lalu kolom *smoking_history* diubah menjadi nilai numerik menggunakan mapping, dan disimpan dalam kolom baru *smoking_numeric_original*. Nilai -1 kemungkinan merepresentasikan kategori "*No Info*". Selanjutnya, digunakan matplotlib.pyplot dan seaborn untuk membuat dua subplot yang sejajar secara horizontal. Plot pertama menampilkan histogram distribusi dari *smoking_numeric_original* sebelum normalisasi, sementara plot kedua menampilkan distribusi dari smoking_normalized pada *DataFrame* df1, yang sudah melalui proses normalisasi Min-Max. Fungsi sns.histplot() digunakan untuk menggambarkan histogram dengan kurva KDE (*Kernel Density Estimation*) guna memberikan gambaran distribusi data yang lebih halus. Akhirnya, plt.tight_layout() memastikan layout antar plot tidak saling tumpang tindih, dan plt.show() menampilkan visualisasi pada Gambar 5.



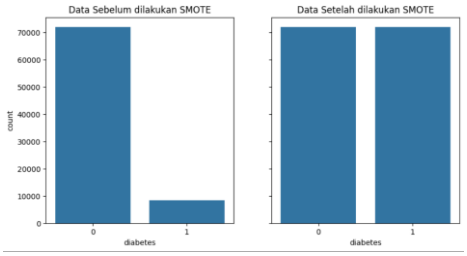
Gambar 5. Hasil *Running Codding* Perbandingan Normalisasi

Berdasarkan grafik dan tabel dapat diketahui bahwa distribusi setelah normalisasi belum seimbang dan dibutuhkannya proses *resampling* untuk mengatasi ketidakseimbangan pada data.

3.3 Hasil *Resampling Data*

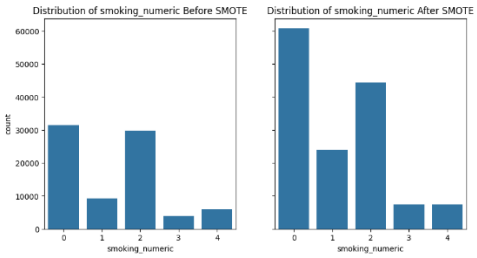
Pada penerapan SMOTE, pada atribut diabetes menghasilkan 72.015 data pengidap diabetes (1) dan 72.015 yang tidak mengidap diabetes (0), yang dimana kedua data tersebut seimbang . Grafik perbandingan data sebelum dan sesudah Resampling dapat dilihat pada Gambar 6.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032



Gambar 6. Hasil SMOTE pada atribut Diabetes.

Sementara pada atribut *smoking_history* atau *smoking_numeric*, terdapat perbedaan dimana distribusi data setelah dilakukannya SMOTE, terlihat hasil seperti pada kelas 1, 3, dan 4, dimana memiliki jumlah data relatif sedikit mengalami peningkatan jumlah sampel yang lumayan signifikan. Meskipun jumlah masing-masing kelas tidak sepenuhnya disamakan, distribusi data secara keseluruhan menjadi lebih seimbang dibandingkan sebelum dilakukan SMOTE. Hal ini menunjukkan bahwa SMOTE mampu mengurangi dominasi kelas mayoritas, sehingga diharapkan dapat meningkatkan kinerja model klasifikasi dengan mengurangi bias terhadap kelas yang lebih besar. grafik perbandingan dapat dilihat melalui Gambar 7



Gambar 7 Hasil SMOTE pada atribut *Smoking_numeric*.

Dimana pada ke atribut smoking_numeric menghasilkan total data pada kelas 0 ialah 60.801, kelas 1 menghasilkan data sebesar 24.035, kelas 2 menghasilkan 44.316, kelas 3 sebesar 7.444, dan kelas 4 menghasilkan 7.434.

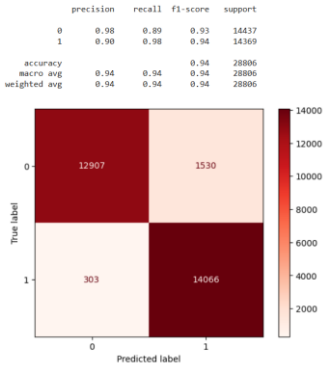
3.4 Hasil pengujian antar model

Setelah dilakukannya tahap *Resampling*, data kemudian dipisah dengan menggunakan rasio 80:20. Dimana setelah itu diimplementasikan kedalam model – model yang sudah ditentukan.

a. *K-Nearest Neighbor (KNN)*

Hasil pada gambar 8, membuktikan bahwa KNN dibangun dengan parameter *n_neighbors* = 3. Hasil pengujian menunjukkan *accuracy* sebesar 94%, *precision* kelas 0 sebesar 0,98 dan kelas 1 sebesar 0,90, *recall* masing-masing 0,89 dan 0,98.

Confusion matrix menunjukkan 1.547 *false positive* dan 271 *false negative*. KNN efektif dalam mengenali pola lokal, tetapi sensitif terhadap *outlier*.

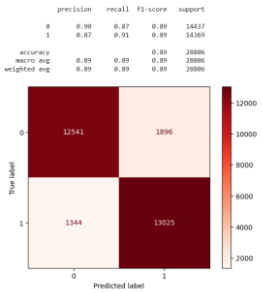


Gambar 8. Hasil Analisis *K-Nearest Neighbor*

b. *Support Vector Method (SVM*)

SVM menghasilkan *accuracy* 89%, *precision* kelas 0 sebesar 0,90 dan kelas 1 sebesar 0,87, *recall* masing-masing 0,87 dan 0,91. Jumlah *false positive* sebesar 1.885 dan *false negative* sebesar 1.354, menunjukkan bahwa model ini cenderung kurang optimal pada distribusi fitur yang kompleks, dimana dapat dilihat berdasarkan Gambar 9.

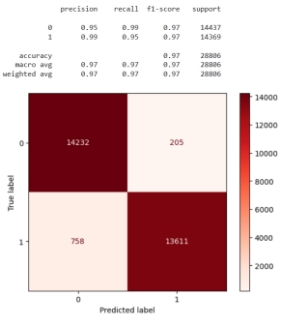
Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032



Gambar 9. Hasil Analisis *Support Vector Method*

c. XGBoost

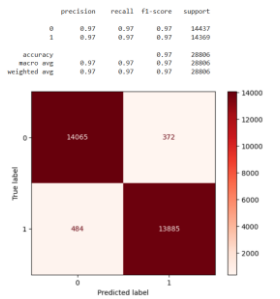
Pada Gambar 10, XGBoost memberikan performa terbaik dengan *accuracy* 97%, *precision* kelas 0 sebesar 0,95 dan kelas 1 sebesar 0,99, *recall* masing-masing 0,99 dan 0,95. *False positive* yang dihasilkan hanya 195, terendah dibandingkan model lainnya. XGBoost unggul berkat proses *boosting* iteratif yang meminimalkan kesalahan prediksi.



Gambar 10. Hasil Analisis XGBoost

d. *Random Forest*

Pada Gambar 11, *Random Forest* juga menunjukkan performa tinggi dengan *accuracy* 97%, *precision* dan *recall* pada kedua kelas sama-sama 0,97. Jumlah *false positive* sebesar 393 dan *false negative* sebesar 483. Model ini mampu mengurangi *overfitting* melalui agregasi banyak pohon keputusan.



Gambar 11. Hasil Analisis *Random Forest*

Perbandingan hasil tersebut dapat dilihat melalui Tabel 4, Tabel 5 dan Gambar 12. Tabel 4 menunjukkan hasil perhitungan manual klasifikasi tiap model dengan empat metrik evaluasi, yaitu Accuracy, Precision, Recall, dan F1-Score. Dari hasil tersebut terlihat bahwa *Random Forest* dan *XGBoost* memberikan performa terbaik dengan nilai akurasi yang sama yaitu 0,975, serta F1-Score sebesar 0,968. Model *Support Vector Machine* menunjukkan performa lebih rendah dengan akurasi 0,89 dan F1-Score 0,861, sementara *K–Nearest Neighbor* menghasilkan akurasi 0,94 dengan F1-Score 0,923, sehingga dapat disimpulkan bahwa *Random Forest* dan *XGBoost* unggul dibandingkan model lainnya dalam perhitungan manual.

Tabel 4. Hasil Perhitungan Manual Klasifikasi Tiap Model

Model	Accuracy	Precision	Recall	F1 - Score
Random Forest	**0.975**	**0.974**	**0.962**	**0.968**
XGBoost	**0.975**	**0.961**	**0.974**	**0.968**

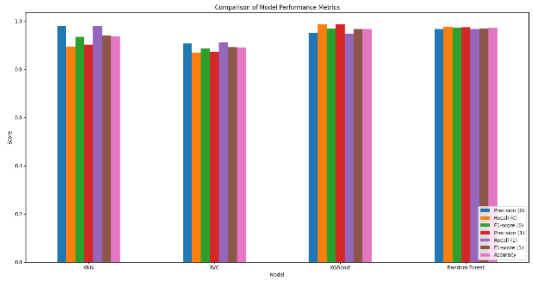
Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

Support Vector Machine	0.89	0.89	0.872	0.861
K – Nearest Neighbor	0.94	0.923	0.923	0.923

Tabel 5 menyajikan hasil klasifikasi otomatis dari tiap model dengan nilai evaluasi yang lebih sederhana namun konsisten. Hasilnya memperlihatkan bahwa *Random Forest* dan *XGBoost* tetap menjadi model dengan kinerja terbaik, masing-masing memperoleh nilai akurasi, precision, recall, dan F1-Score sebesar 0,97. Sementara itu, *Support Vector Machine* hanya mencapai skor 0,89 pada semua metrik, dan *K–Nearest Neighbor* memperoleh nilai 0,94 di setiap metrik. Hal ini memperkuat hasil pada Tabel 4 bahwa *Random Forest* dan *XGBoost* merupakan algoritma paling optimal untuk digunakan pada kasus klasifikasi ini.

Tabel 5. Hasil Klasifikasi Tiap Model

Model	Accuracy	Precision	Recall	F1 - Score
Random Forest	**0.97**	**0.97**	**0.97**	**0.97**
XGBoost	**0.97**	**0.97**	**0.97**	**0.97**
Support Vector Machine	0.89	0.89	.89	0.89
K – Nearest Neighbor	0.94	0.94	0.94	0.94



Gambar 12. Grafik perbandingan hasil tiap model

Dari hasil pada Tabel 4 dan Gambar 4, dapat dilihat bahwa model Random Forest dan XGBoost adalah yang terbaik berdasarkan lengkap metric yaitu Accuracy, Precision, Recall sampai dengan F1-Score. Dari empat model pembanding yaitu K-Nearest Neighbor dan KNN adalah model yang cukup baik dengan score 0,94 di semua aspek. Dan yang paling buruk adalah model Support Vector Machine dan diperoleh dengan nilai 0,89. Grafik pada Gambar 10 memperkuat hasil tabel karena memperlihatkan seberapa cara tiap metrik mempertahankan score antar masing-masing model secara visual, di mana batang berwarna memaparkan seberapa tinggi konsistensi score-nya pada Random Forest dan XGBoost, KNN sedikit lebih rendah, dan pada klasifikasi yang menurun dua minggu latihan, SVM menurun secara signifikan. Hasil penelitian ini merupakan hasil dari tahap evaluasi KDD. Hasil akhir dari penerapan algoritma data mining menunjukkan bahwa Random Forest dan XGBoost menghasilkan hasil yang paling baik. Temuan ini selaras dengan hasil penelitian sebelumnya dari Sofiah, Tania, Meiriza dan Wedhasmara pada tahun 2024, mengklaim bahwa hasil data pertumbuhan metode penambahan memastikan model diskriminatif memberikan hasil yang akurat dan stabil yang 7. Demikian pula, penelitian oleh Metti Detricia Pratiwi dan Ken Ditha Tania pada tahun 2025 menegaskan kebenaran pertanyaan terakhir, menyatakan bahwa proses evaluasi dalam KDD akan memungkinkan 8 melakukan pembandingan performa dari beberapa metode sehingga pola yang dihasilkan memiliki relevansi yang lebih kuat terhadap tujuan penelitian

4. KESIMPULAN

Berdasarkan hasil penelitian, semua model yang diuji mampu memprediksi diabetes dengan tingkat akurasi yang beragam. Prosedur pengembangan sistem prediksi dengan menerapkan kerangka Knowledge Discovery in Databases secara lebih berurutan, mulai dari seleksi data, pra-pemrosesan, penyeimbangan kelas, hingga evaluasi model, sehingga perbandingan kinerja antar model dapat dibuat. Berdasarkan penemuan, tidak diragukan lagi bahwa Random Forest dan XGBoost adalah model dengan performa tertinggi. Random Forest memperoleh akurasi, presisi, recall, dan F1-score 0,97, Guna pengoptimalan, KDD sangat diunggulkan, terutama jika dilihat nilai rata-rata pada prediksi terbalik. Nilai paling baik kemudian didapat KNN sebagai metode prediksi diabetes teroptimum; yaitu 0,94 pada setiap metrik, sementara SVM memiliki rata-rata paling rendah yaitu 0,89. Oleh sebab itu, KDD sangat memungkinkan penggunaan metode prediksi diabetes yang optimum dalam kasus tersebut, termasuk penggunaan Random Forest dan XGBoost sebagai metode pertama diatas lainnya, terutama ketika menjadikan kriteria dataset yang sama.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

REFERENCES

[1] R. P. Febrinasari, T. A. Sholikah, D. N. Pakha, and S. E. Putra, *BUKU SAKU DIABETES MELITUS UNTUK AWAM*, 1st ed., vol. 1. Surakarta, Jawa Tengah : Penerbitan dan Pencetakan UNS (UNS Press) , 2020.

[2] International Diabetes Federation, "Fakta & angka," https://journal.mediapublikasi.id/index.php/logic/article/view/4963.

[3] H. Aulianah and H. Meylina, "Babul Ilmi_Jurnal Ilmiah Multi Science Kesehatan," vol. 14, no. 2, pp. 161–171, 2022, [Online]. Available: https://jurnal.stikes-aisyiyah-palembang.ac.id/index.php/Kep/article/view/

[4] R. Fan, N. Zhang, L. Yang, J. Ke, D. Zhao, and Q. Cui, "AI-based prediction for the risk of coronary heart disease among patients with type 2 diabetes mellitus," *Sci Rep*, vol. 10, no. 1, Dec. 2020, doi: 10.1038/s41598-020-71321-2.

[5] A. Brahmandjati, A. Mizwar, A. Rahim, and F. Asharudin, "Optimasi Prediksi Diabetes Dengan Algoritma XGBoost Dan Teknik Preprocessing Data," Yogyakarta, Dec. 2024. doi: https://doi.org/10.47065/bits.v6i3.6110.

[6] A. Farisi and A. Homaidi, "Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Teknologi dan Manajemen Industri Terapan (JTMIT)*, vol. 4, no. 3, pp. 612–621, 2025.

[7] N. Devian *et al.*, "PREDIKSI PENYAKIT DIABETES DENGAN METODE K-NEAREST NEIGHBOR (KNN) DAN SELEKSI FITUR INFORMATION GAIN," 2024.

[8] R. Rastogi and M. Bansal, "Diabetes prediction model using data mining techniques," *Measurement: Sensors*, vol. 25, Feb. 2023, doi: 10.1016/j.measen.2022.100605.

[9] R. R. Pradana and Y. P. Astuti, "Perbandingan Kinerja Metode Naïve Bayes dan Random Forest untuk Klasifikasi Penyakit Diabetes Berdasarkan Data Medis," *Technology and Science (BITS)*, vol. 7, no. 1, 2025, doi: 10.47065/bits.v7i1.7446.

[10] Z. Amri, Muhammad Rodi, M. Nurul Wathani, Amir Bagja, and Zulkipli, "Prediksi Diabetes Menggunakan Algoritma K-Nearest (KNN) Teknik SMOTE-ENN," *Infotek: Jurnal Informatika dan Teknologi*, vol. 8, no. 1, pp. 193–204, Jan. 2025, doi: 10.29408/jit.v8i1.27975.

[11] Akbar Febrian Dwi Hastono, Anik Vega Vitianingsih, Pamudi Pamudi, Anastasia Lidya Maukar, and Seftin Fitri Ana Wati, "Diabetes Mellitus Disease Prediction Using Logistic Regression (LR) and Support Vector Machine (SVM) Methods," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 5, no. 1, pp. 54–64, Mar. 2025, doi: 10.51454/decode.v5i1.1039.

[12] M. D. Pratiwi and K. D. Tania, "Knowledge Discovery Through Topic Modeling on GoPartner User Reviews Using BERTopic, LDA, and NMF," *Journal of Applied Informatics and Computing*, vol. 9, no. 1, pp. 1–7, Jan. 2025, doi: 10.30871/jaic.v9i1.8782.

[13] N. A. Sofiah, K. D. Tania, A. Meiriza, and A. Wedhasmara, "A Comparative Assessment SARIMA and LSTM Models for the Gurugram Air Quality Index's Knowledge Discovery," in *2024 International Conference on Electrical Engineering and Computer Science (ICECOS)*, IEEE, Sep. 2024, pp. 26–31. doi: 10.1109/ICECOS63900.2024.10791243.

[14] V. Novalia, K. Ditha Tania, A. Meiriza, and A. Wedhasmara, "Knowledge Discovery of Application Review Using Word Embedding's Comparison with CNN-LSTM Model on Sentiment Analysis," in *2024 International Conference on Electrical Engineering and Computer Science (ICECOS)*, IEEE, Sep. 2024, pp. 234–238. doi: 10.1109/ICECOS63900.2024.10791113.

[15] S. Eka, A. Buananta, and A. Chowanda, "BI DASHBOARD TO SUPPORT DECISION MAKING ON PRODUCT PROMOTION FOR PAYMENT/PURCHASE TRANSACTIONS ON E-BANKING," *J Theor Appl Inf Technol*, vol. 15, no. 15, 2021, [Online]. Available: www.jatit.org

[16] N. Putu, A. Widiari, M. Agus, D. Suarjaya, and D. Putra Githa, "Teknik Data Cleaning Menggunakan Snowflake untuk Studi Kasus Objek Pariwisata di Bali."

[17] S. K. Dirjen *et al.*, "Terakreditasi SINTA Peringkat 2 Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman," *masa berlaku mulai*, vol. 1, no. 3, pp. 117–122, 2020.

[18] M. Ihksan, H. Susilo, N. Abdillah, and S. S. Saintika, "PENERAPAN DATA MINING K-MEANS CLUSTERING KEBUTUHAN OBAT DI KLINIK MEDIKA SAINTIKA," *Jurnal Kesehatan Medika Saintika Juni 2023 |Vol 14 Nomor*, vol. 14, no. 1, p. 394, 2023, doi: 10.30633/jkms.v14i1.2581.

[19] N. G. Ramadhan and F. D. Adhinata, "TEKNIK SMOTE DAN GINI SCORE DALAM KLASIFIKASI KANKER PAYUDARA," *RADIAL : Jurnal Peradaban Sains, Rekayasa dan Teknologi*, vol. 9, no. 2, pp. 125–134, Dec. 2021, doi: 10.37971/radial.v9i2.229.

[20] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiasari, "Analisis sentimen pada rating aplikasi Shopee menggunakan metode Decision Tree berbasis SMOTE," *AITI: Jurnal Teknologi Informasi*, vol. 18, no. Agustus, pp. 173–184, 2021.

[21] I. Setiawan, I. Fatah Yasin, Y. Tri Desianti, P. Studi Sistem Dan Teknologi Informasi, F. Sains Dan Teknologi, and A. Surakarta, "Komparasi Kinerja Algoritma Random Forest, Decision Tree, Naïve Bayes, dan KNN dalam Prediksi Tingkat Depresi Mahasiswa Menggunakan Student Depression Dataset," 2025. [Online]. Available: http://creativecommons.org/licences/by/4.0/

[22] L. R. Sitompul, A. A. Nababan, M. L. Manihuruk, W. A. Ponsen, and S. Supriyandi, "Comparison of Xgboost, Random Forest and Logistic Regression Algorithms in Stroke Disease Classification," *Sinkron*, vol. 9, no. 2, pp. 957–968, Jun. 2025, doi: 10.33395/sinkron.v9i2.14794.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9032

[23] P. Sidik, I. Made, G. Sunarya, I. Gede, and A. Gunadi, "Comparison of Random Forest and Support Vector Machine Methods in Sentiment Analysis of Student Satisfaction Questionnaire Comments at ITB STIKOM Bali," 2025. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[24] X. Deng, H. Shao, L. Shi, X. Wang, and T. Xie, "A classification–detection approach of COVID-19 based on chest X-ray and CT by using keras pre-trained deep learning models," *CMES - Computer Modeling in Engineering and Sciences*, vol. 125, no. 2, pp. 579–596, 2020, doi: 10.32604/cmes.2020.011920.

[25] M. Fadli and R. A. Saputra, "KLASIFIKASI DAN EVALUASI PERFORMA MODEL RANDOM FOREST UNTUK PREDIKSI STROKE Classification And Evaluation Of Performance Models Random Forest For Stroke Prediction," vol. 12, [Online]. Available: http://jurnal.umt.ac.id/index.php/jt/index