Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

Perbandingan Kinerja Algoritma K-Nearest Neighbors dan Decision Tree untuk Klasifikasi Diabetes

Amar Haris Yunianto[1,*], Egia Rosi Subhiyakto[2]

[1] Fakultas Ilmu Komputer, Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia
[2] Research Center for Intelligent Distributed Surveillance and Security (IDSS), Universitas Dian Nuswantoro, Semarang, Indonesia
Email: [1,*]111202113813@mhs.dinus.ac.id, [2]egia@dsn.dinus.ac.id
Email Penulis Korespondensi: 111202113813@mhs.dinus.ac.id
Submitted: **27/12/2024**; Accepted: **15/03/2025**; Published**: 16/03/2025**

Abstrak−Diabetes adalah penyakit metabolik kronis yang menjadi perhatian utama di bidang kesehatan global karena prevalensinya yang terus meningkat, termasuk di Indonesia, dengan dampak signifikan pada kesehatan individu dan sistem kesehatan. Penelitian ini bertujuan untuk membandingkan kinerja algoritma *K-Nearest Neighbors* (KNN) dan *Decision Tree* (DT) dalam klasifikasi diabetes menggunakan dataset *Pima Indians Diabetes Database* (PIDD). Metode penelitian meliputi pengumpulan data, pra-pemrosesan, penanganan missing value, deteksi dan penanganan outlier, serta teknik balancing data menggunakan *Synthetic Minority Oversampling Technique* (SMOTE) untuk mengatasi ketidakseimbangan kelas dalam dataset. Implementasi model dilakukan dengan mengoptimalkan parameter menggunakan GridSearchCV, sedangkan evaluasi kinerja dilakukan berdasarkan matrik akurasi, presisi, recall, dan F1 score. Hasil penelitian menunjukkan bahwa algoritma DT memiliki performa lebih unggul dibandingkan KNN, baik tanpa SMOTE maupun dengan SMOTE. Pada model tanpa SMOTE, DT mencapai akurasi 85,71%, sementara KNN hanya mencapai 83,12%. Setelah penerapan SMOTE, kinerja kedua algoritma meningkat signifikan, dengan DT mencapai akurasi 92%, presisi 94%, recall 90,38%, dan F1 score 92,16%, sedangkan KNN mencapai akurasi 91%, recall 96,59%, dan F1 score 90,43%. Penelitian ini mengungkapkan bahwa penggunaan SMOTE secara efektif meningkatkan kinerja model dalam menangani ketidakseimbangan data, sementara algoritma DT menunjukkan stabilitas performa yang lebih baik. Temuan ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan model prediksi yang lebih akurat untuk diagnosis diabetes, sekaligus memperkaya wawasan tentang penerapan machine learning di bidang kesehatan.

Kata Kunci: Diabetes; Klasifikasi; K-Nearest Neighbors; Decision Tree; SMOTE

Abstract−Diabetes is a chronic metabolic disease that is a major concern in global health due to its increasing prevalence, including in Indonesia, with significant impacts on individual health and health systems. This study aims to compare the performance of K-Nearest Neighbors (KNN) and Decision Tree (DT) algorithms in diabetes classification using the Pima Indians Diabetes Database (PIDD) dataset. Research methods include data collection, pre-processing, missing value handling, outlier detection and handling, and data balancing techniques using Synthetic Minority Oversampling Technique (SMOTE) to overcome class imbalance in the dataset. Model implementation is done by optimizing parameters using GridSearchCV, while performance evaluation is done based on accuracy, precision, recall, and F1 score matrices. The results show that the DT algorithm has superior performance compared to KNN, both without SMOTE and with SMOTE. In the model without SMOTE, DT achieved 85.71% accuracy, while KNN only reached 83.12%. After applying SMOTE, the performance of both algorithms improved significantly, with DT achieving 92% accuracy, 94% precision, 90.38% recall, and 92.16% F1 score, while KNN achieved 91% accuracy, 96.59% recall, and 90.43% F1 score. This study revealed that the use of SMOTE effectively improved the model's performance in handling data imbalance, while the DT algorithm showed better performance stability. These findings are expected to make a significant contribution to the development of more accurate prediction models for diabetes diagnosis, while enriching insights into the application of machine learning in the healthcare field.

Keywords: Diabetes; Classification; K-Nearest Neighbors; Decision Tree; SMOTE

1. PENDAHULUAN

Kesehatan bagian penting dari kehidupan setiap orang, yang menentukan kualitas hidup dan produktivitas individu. Namun, seiring dengan perubahan gaya hidup modern, berbagai penyakit kronis mula menjadi ancaman serius bagi masyarakat di dunia. Salah satu penyakit yang terus menjadi perhatian utama di bidang kesehatan adalah diabetes, yang prevalensinya semakin meningkat setiap tahun dan memberikan dampak signifikan terhadap kesehatan individu maupun sistem kesehatan secara keseluruhan. Diabetes merupakan penyakit metabolik yang dicirikan dengan tubuh tidak responsif terhadap insulin atau produksi insulin yang tidak tercukupi, sehingga mengakibatkan hiperglikemia atau peningkatan glukosa darah [1]. Diabetes dianggap sebagai penyakit pembunuh diam-diam karena orang sering kali baru menyadari bahwa mereka mengidapnya ketika terjadi komplikasi serius. Komplikasi serius, misalnya gangguan pada jantung, saraf, mata, ginjal, dan pembuluh darah, dapat timbul akibat penyakit ini, yang seringkali berlangsung seumur hidup [2].

Menurut *International Diabetes Federation* (IDF), sejak penerbitan pertama *IDF Diabetes Atlas* pada tahun 2000, estimasi prevalensi diabetes pada orang dewasa berusia 20–79 tahun telah meningkat lebih dari tiga kali lipat, dari 151 juta (4,6% dari populasi dunia pada saat itu) menjadi 537 juta (10,5%) pada tahun 2021. Jika tidak ada langkah yang tepat, diperkirakan jumlah ini akan terus naik menjadi 643 juta (11,3% dari populasi di dunia) pada tahun 2030 dan melonjak menjadi 783 juta (12,2%) pada tahun 2045. Di Indonesia, prevalensi diabetes juga menunjukkan peningkatan yang signifikan, dengan sekitar 19,5 juta orang dewasa terdiagnosis diabetes pada tahun 2021, dan diprediksi akan mencapai 28,6 juta pada tahun 2045 [3]. Angka-angka ini menunjukkan pentingnya

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

peningkatan kesadaran, pencegahan, dan pengelolaan diabetes baik secara global maupun nasional untuk mengatasi situasi yang semakin mendesak ini.

Melihat ancaman serius dari peningkatan kasus diabetes, penting untuk tidak hanya memahami faktor-faktor yang memicu dan memperburuk kondisi ini, tetapi juga mengeksplorasi pendekatan baru dalam upaya pencegahannya. Dengan demikian, diperlukan pendekatan klasifikasi yang tepat dan efektif guna membantu diagnosis dan penanganan diabetes. Algoritma *machine learning* menjadi salah satu pendekatan yang bisa diterapkan untuk mengidentifikasi pola pada data, serta membantu dalam pengambilan keputusan klinis. Dua algoritma yang banyak digunakan dalam klasifikasi adalah *K-Nearest Neighbors* (KNN) dan *Decision Tree* (DT), yang masing-masing memiliki keunggulan dan tantangan tersendiri dalam penerapannya.

Beberapa penelitian sebelumnya telah menggunakan algoritma KNN dan DT untuk klasifikasi diabetes. Pertama, penelitian oleh Qonitah Alia Puteri dkk. (2023) membandingkan algoritma Naive Bayes (NB) dan KNN. Hasil penelitian tersebut menunjukkan KNN memperoleh akurasi sebesar 71% [4]. Kedua, penelitian oleh Aziz Perdana dkk. (2023) menggunakan algoritma KNN dengan hasil akurasi tertinggi sebesar 83,12% [5]. Ketiga, penelitian oleh Nining Nur Habibah dkk. (2023) menggunakan algoritma DT dengan hasil akurasi tertinggi sebesar 76,67% [6]. Keempat, penelitian oleh Nitisha Aggarwal dkk. (2023) membandingkan berbagai algoritma *machine learning*. Dari algoritma yang diuji, KNN mencatat akurasi sebesar 69,48%, sementara algoritma DT menghasilkan akurasi sebesar 74% [7]. Kelima, penelitian oleh Prosanjeet Sarkar dkk. (2023) juga membandingkan berbagai algoritma *machine learning*. Dari algoritma yang diuji, KNN mencatat akurasi sebesar 84,07%, sementara algoritma DT menghasilkan akurasi sebesar 80,12% [8]. Keenam, penelitian oleh Jobeda Jamal Khanam dkk. (2021) menunjukkan bahwa algoritma KNN mencatat akurasi sebesar 79,42%, sedangkan algoritma DT menghasilkan akurasi sebesar 73,14% [9]. Ketujuh, penelitian oleh Muhammad Exell Febrian dkk. (2023) membandingkan algoritma KNN dan NB. Hasil penelitian menunjukkan bahwa KNN memiliki akurasi sebesar 77,92% [10]. Kedelapan, penelitian oleh Egi Safitri dkk. (2024) mencatat bahwa algoritma DT memiliki akurasi sebesar 73% [11]. Kesembilan, penelitian oleh Bhuvaneswari Amma N.G. (2024) menunjukkan bahwa algoritma KNN memiliki akurasi sebesar 85,56%, sedangkan algoritma DT mencatat akurasi sebesar 81,11% [12]. Kesepuluh, penelitian oleh P. Venkata Kishan Rao dkk. (2024) membandingkan beberapa algoritma machine learning, termasuk KNN dan DT. Dalam penelitian ini KNN mencatat akurasi sebesar 70%, sedangkan algoritma DT mencapai akurasi 72,66% [13].

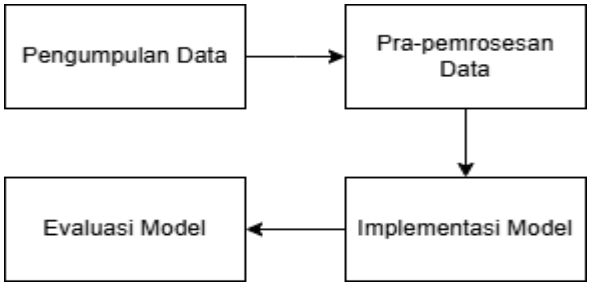
Algoritma KNN dan DT telah banyak digunakan sebagai model untuk mengklasifikasikan diabetes. Namun, sebagian besar nilai evaluasi kinerja untuk klasifikasi diabetes diatas belum optimal, artinya masih dapat ditingkatkan dengan pendekatan yang sesuai. Salah satu metode yang dapat meningkatkan nilai evaluasi performa klasifikasi adalah *Synthetic Minority Over-sampling Technique* (SMOTE). Tantangan dalam klasifikasi seringkali muncul ketika menghadapi ketidakseimbangan kelas dalam dataset, dimana kelas minoritas kurang terwakili dibandingkan kelas mayoritas. Ketidakseimbangan ini dapat menyebabkan model cenderung mengabaikan kelas minoritas, sehingga akurasi klasifikasi menurun. Untuk meningkatkan performa model, oversampling dengan SMOTE menjadi sangat penting, karena mampu menyeimbangkan distribusi kelas dengan menghasilkan sampel sintetis dari kelas minoritas [14][15].

SMOTE merupakan teknik oversampling yang menciptakan sampel sintetis dari kelas minoritas untuk menyeimbangkan distribusi kelas. Metode ini melibatkan penggunaan algoritma KNN untuk menemukan beberapa tetangga terdekat untuk setiap sampel kelas minoritas, kemudian melakukan interpolasi nilai fitur antara sampel asli dan salah satu tetangga terdekatnya untuk menciptakan sampel sintetis, yang kemudian ditambahkan ke kumpulan data hingga distribusi kelas menjadi lebih seimbang [16]. Penerapan SMOTE telah terbukti efektif dalam meningkatkan performa model machine learning di berbagai bidang penelitian. Sebagai contoh, penelitian oleh Agung Nugroho dkk. (2023) menggunakan algoritma Random Forest (RF) berbasis SMOTE untuk prediksi kebangkrutan perusahaan [17]. Andreyestha dkk. (2022) menggunakan algoritma NB dan RF berbasis SMOTE dalam analisis sentimen kicauan Twitter tentang Tokopedia [18]. Eka Rahmawati dkk. (2023) menerapkan algoritma KNN dan Deep Learning berbasis SMOTE pada ulasan pengguna aplikasi ChatGPT di Google Play Store [19]. Mutiara Persada Pulungan dkk. (2024) menggunakan algoritma NB berbasis SMOTE untuk klasifikasi kepribadian MBTI [20]. Sementara itu, Andi Surya Firmansyah dkk. (2023) menerapkan algoritma KNN untuk klasifikasi analisis sentimen [21]. Penelitian ini bertujuan untuk melakukan perbandingan kinerja algoritma KNN dan DT dalam klasifikasi diabetes. Evaluasi dilakukan berdasarkan matrik performa utama, yaitu akurasi, presisi, recall dan f1 score, untuk memberikan gambaran yang komprehensif mengenai keunggulan masing-masing algoritma. Selain itu, penelitian ini juga menganalisis pengaruh penggunaan teknik oversampling, khususnya SMOTE, dalam meningkatkan performa klasifikasi kedua algoritma tersebut. Dengan demikian, penelitian ini diharapkan dapat memberikan wawasan yang mendalam tentang algoritma yang lebih efektif dalam memprediksi diabetes, sekaligus mengidentifikasi bagaimana penerapan SMOTE dapat mengatasi ketidakseimbangan data untuk mendukung hasil klasifikasi yang lebih optimal.

2. METODOLOGI PENELITIAN

Metode penelitian ini terdiri dari beberapa tahapan utama yang dilakukan secara sistematis untuk membandingkan kinerja algoritma KNN dan DT dalam klasifikasi diabetes, dimana alur tahapan tersebut dapat dilihat secara visual pada Gambar 1 alur penelitian.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550



Gambar 1. Alur Penelitian

Gambar 1 menunjukkan tahapan utama dalam penelitian ini, yang dilakukan secara sistematis untuk membandingkan kinerja algoritma KNN dan DT dalam klasifikasi diabetes. Proses dimulai dengan pengumpulan dataset, diikuti oleh tahap pra-pemrosesan data untuk memastikan kualitas data yang optimal. Selanjutnya, model diterapkan dengan memilih algoritma yang sesuai, seperti KNN dan DT. Setelah itu, dilakukan evaluasi model untuk mengukur performa dan keakuratan prediksi. Hasil evaluasi ini kemudian dianalisis menggunakan *confusion matrix* guna memperoleh kesimpulan serta memberikan rekomendasi berdasarkan temuan penelitian.

2.1 Pengumpulan Data

Penelitian ini menggunakan *Pima Indians Diabetes Database* (PIDD) yang diperoleh dari *platform* Kaggle. Dataset ini berasal dari *National Institute of Diabetes and Digestive and Kidney Diseases*. Dataset ini memiliki sejumlah batasan, di antaranya semua pasien yang terdaftar adalah perempuan berusia minimal 21 tahun dengan latar belakang etnis Pima Indian. Dataset ini mencakup beberapa fitur, antara lain *Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, *BodyMassIndex* , *DiabetesPedigreeFunction*, *Age*, dan *Outcome*, yang digunakan untuk memprediksi kemungkinan terjadinya penyakit diabetes pada individu.

2.2 Pra-pemrosesan Data

Pada tahap pra-pemrosesan data, dilakukan serangkaian langkah untuk memastikan kualittas dataset sebelum digunakan dalam proses pelatihan model. Langkah pertama adalah membersihkan data dengan memeriksa keberadaan nilai kosong pada dataset. Jika ditemukan nilai kosong, atribut tersebut diisi menggunakan nilai median, yang dihitung secara terpisah berdasarkan setiap kelas target. Selanjutnya, dilakukan deteksi dan penanganan outlier. Sebuah data dikategorikan sebagai apabila nilainya berada di luar jangkauan yang ditentukan oleh area utama dan garis whisker pada boxplot, yaitu lebih besar dari $Q3 + 1{,}5 \times IQR$ atau lebih kecil dari $Q1 - 1{,}5 \times IQR$. Dalam hal ini, Q1 merepresentasikan kuartil pertama, Q3 adalah kuartil ketiga, dan IQR (*interquartile range*) dihitung sebagai selisih antara Q3 dan Q1 [22].

Setelah itu, langkah berikutnya adalah menangani ketidakseimbangan pada dataset. Ketidaksemibangan dalam dataset diatasi menggunkan teknik SMOTE. SMOTE menciptakan sampel sintetis untuk kelas minoritas dengan melakukan interpolasi antara sampel yang ada, sehingga jumlah data di setiap kelas menjadi lebih seimbang. Langkah terakhir adalah *feature scaling*, Untuk menyetarakan skala antar fitur, digunakan teknik *StandardScaler*, yang mentransformasikan data sehingga memiliki rata-rata (μ) sebesar 0 dan standar deviasi (σ) sebesar 1, sesuai dengan rumus:

$$z = \frac{x-\mu}{\sigma} \tag{1}$$

di mana z adalah nilai data baru yang telah di-scaling, x merupakan nilai asli data, μ adalah rata-rata nilai dalam atribut, dan σ adalah standar deviasi dari atribut tersebut. Proses ini bertujuan untuk memastikan bahwa semua fitur memiliki skala yang seragam, sehingga model dapat bekerja lebih optimal tanpa bias terhadap fitur dengan rentang nilai yang lebih besar.

2.3 Implementasi Model

Pada tahap ini, dilakukan eksperimen dengan berbagai konfigurasi model untuk membandingkan performa klasifikasi menggunakan algoritma KNN dan DT. Eksperimen ini mencakup dua kondisi utama, yaitu model yang dikembangkan tanpa penerapan metode penyeimbangan data dan model yang menggunakan metode SMOTE untuk menangani ketidakseimbangan kelas pada dataset. Untuk setiap kondisi, dilakukan pencarian parameter optimal dengan menggunakan teknik optimasi grid search (*GridSearchCV*), yang secara sistematis menguji berbagai kombinasi parameter model. GridSearchCV digunakan untuk menentukan kombinasi model dan hyperparameter dengan menguji serta memvalidasi setiap kemungkinan kombinasi secara otomatis. Proses ini membantu mengoptimalkan waktu pemrosesan. Selain itu, kombinasi hyperparameter yang memberikan akurasi tertinggi dan error terendah dianggap sebagai pilihan yang paling optimal [23].

Pada KNN, parameter yang dioptimalkan meliputi jumlah tetangga terdekat (*n_neighbors*), metode pembobotan (*weights*), dan matrik jarak (*metric*). Sedangkan pada Decision Tree, parameter yang diuji meliputi kriteria pemisahan (*criterion*), kedalaman maksimum pohon (*max_depth*), jumlah minimum sampel untuk pembagian

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

simpul (*min_samples_split*), jumlah minimum sampel pada daun (*min_samples_leaf*), dan jumlah fitur maksimum yang digunakan untuk pemisahan (*max_features*).

2.4 Evaluasi Model

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

Tabel 1. Confusion Matrix

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

Berdasarkan confusion matrix pada Tabel 1, matrik untuk mengevaluasi kinerja model dipresentasikan melalui beberapa skor, termasuk akurasi, presisi, recall, dan f1 score. Rumus untuk matrik-matrik tersebut disajikan pada persamaan (2) hingga (5).

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

$$Presisi = \frac{TP}{TP+FP} \quad (3)$$

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

$$F1\ Score = 2 \times \frac{Presisi \times Recall}{Presisi+Recall} \quad (5)$$

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

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

Tabel 2. Cuplikan dari dataset

No	Preg	Glu	BP	Skin	Ins	BMI	DPF	Age	Out
1	6	148	72	35	0	33,6	0,627	50	1
2	1	85	66	29	0	26,6	0,351	31	0
3	8	183	64	0	0	23,3	0,672	32	1
4	1	89	66	23	94	28,1	0,167	21	0
5	0	137	40	35	168	43,1	2,288	33	1
…	…	…	…	…	…	…	…	…	…
764	10	101	76	48	180	32,9	0,171	63	0
765	2	122	70	27	0	36,8	0,340	27	0
766	5	121	72	23	112	26,2	0,245	30	0
767	1	126	60	0	0	30,1	0,349	47	1
768	1	93	70	31	0	30,4	0,315	23	0

Pima Indians Diabetes Database terdiri dari 768 baris data dan 9 kolom, yang mencakup 8 kolom fitur, yaitu *Pregnancies* (Preg), *Glucose* (Glu), *BloodPressure* (BP), *SkinThickness* (Skin), *Insulin* (Ins), *BodyMassIndex* (BMI), *DiabetesPedigreeFunction* (DPF), dan *Age*, serta 1 kolom target yang berisi *Outcome* (Out). Penjelasan mengenai setiap kolom dalam dataset ini dapat dilihat pada Tabel 3.

Tabel 3. Deskripsi Atribut Dataset

Atribut	Keterangan/Deskripsi
Preg	Jumlah kehamilan
Glu	Kadar glukosa plasma setelah tes toleransi glukosa 2 jam (mg/dl)
BP	Tekanan darah diastolik (mm Hg)
Skin	Ketebalan lipatan kulit triceps (mm)
Ins	Kadar insulin serum setelah toleransi glukosa 2 jam (mu U/ml)
BMI	Indeks masa tubuh (kg/m^2)
DPF	Riwayat diabetes dalam keluarga
Age	Usia

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

Atribut	Keterangan/Deskripsi
Out	Positif diabetes (1) dan negative diabetes (0)

3.2 Pra-pemrosesan Data

Hasil analisis terhadap dataset menunjukkan ada beberapa atribut tidak memiliki nilai yang lengkap. Ditemukan adanya kejanggalan pada atribut Glu, BP, Skin, Ins, dan BMI, karena nilai minimum untuk atribut-atribut tersebut seharusnya tidak bernilai 0. Sementara itu, pada atribut Preg, nilai 0 dianggap sebagai indikator bahwa seseorang belum pernah melahirkan, yang sesuai dengan kondisi sebenarnya. Oleh karena itu, nilai 0 pada atribut Glu, BP, Skin, Ins, dan BMI diganti dengan NaN. Setelah proses perhitungan, terdeteksi adanya *missing value* pada atribut Glu sebanyak 5, atribut BP sebanyak 35, atribut Skin sebanyak 227, atribut Ins sebanyak 374, dan atribut BMI sebanyak 11. Untuk menangani *missing value* pada penelitian ini menggunakan median berdasarkan klasifikasi pada atribut Out. Pada proses imputasi, nilai yang hilang pada atribut diisi dengan median yang dihitung secara terpisah untuk setiap kelas Out (0 dan 1). Hasil dari perhitungan nilai median untuk masing-masing kelas dapat dilihat pada Tabel 4 berikut.

Tabel 4. Hasil nilai median

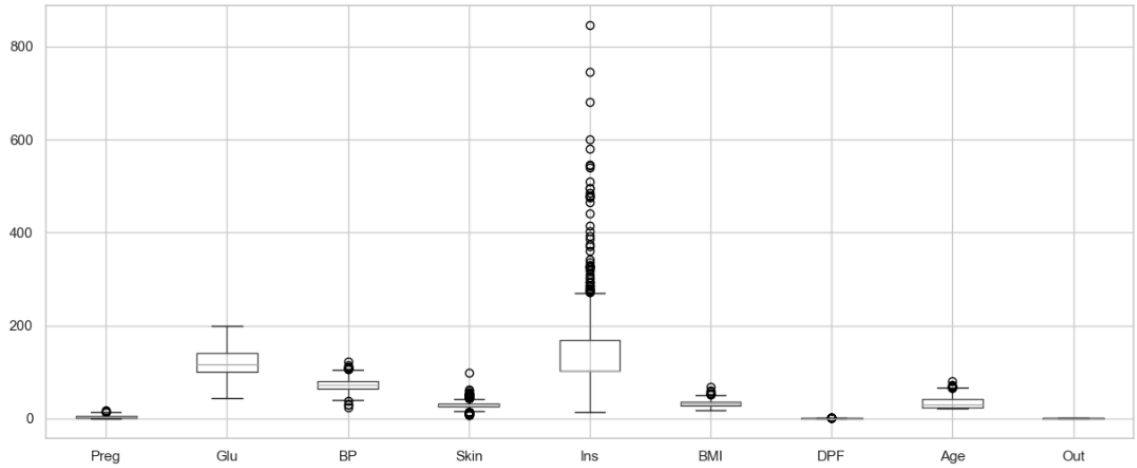
Out	Glu	BP	Skin	Ins	BMI
0	107	70	27	102,5	30,1
1	140	74,5	32	169,5	34,3

Setelah proses imputasi dilakukan, dataset yang telah terisi nilai median ditunjukkan pada Tabel 5. Cuplikan dari dataset setelah proses inputasi nilai median dapat dilihat sebagai berikut.

Tabel 5. Cuplikan dataset setelah inputasi nilai median

No	Preg	Glu	BP	Skin	Ins	BMI	DPF	Age	Out
1	6	148	72	35	169,5	33,6	0,627	50	1
2	1	85	66	29	102,5	26,6	0,351	31	0

Dengan melihat Tabel 4, proses imputasi dilakukan berdasarkan nilai median untuk masing-masing kelas. Tabel 5 menunjukkan bagaimana data yang hilang telah terisi menggunakan nilai median. Setelah menangani *missing value*, langkah selanjutnya adalah penanganan outlier untuk memastikan distribusi data menjadi lebih representatif dan tidak bias terhadap model. Visualisasi data sebelum penanganan outlier dapat dilihat pada Gambar 2.

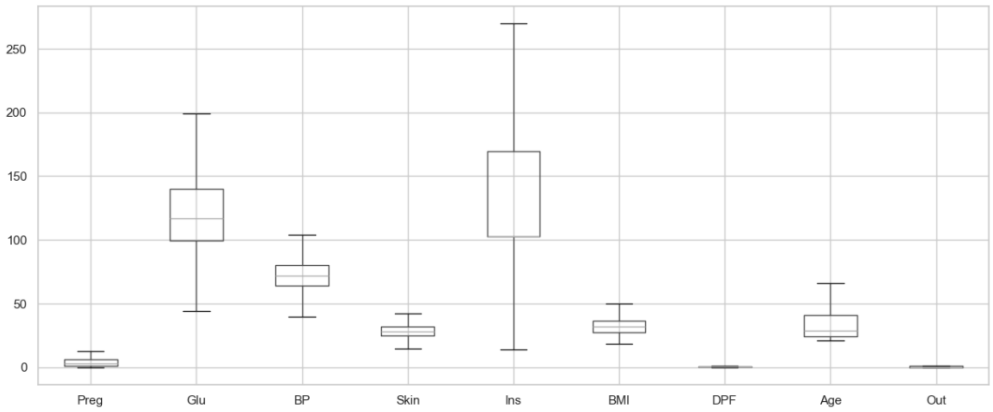


Gambar 2. Outlier

Pada Gambar 2, terlihat adanya outlier pada beberapa atribut, yaitu Preg, BP, Skin, Ins, BMI, DPF, dan Age. Outlier diidentifikasi menggunakan metode *Interquartile Range* (IQR), di mana data yang berada di luar rentang $[Q1 - 1{,}5 \times IQR, Q3 + 1{,}5 \times IQR]$ dianggap sebagai outlier.

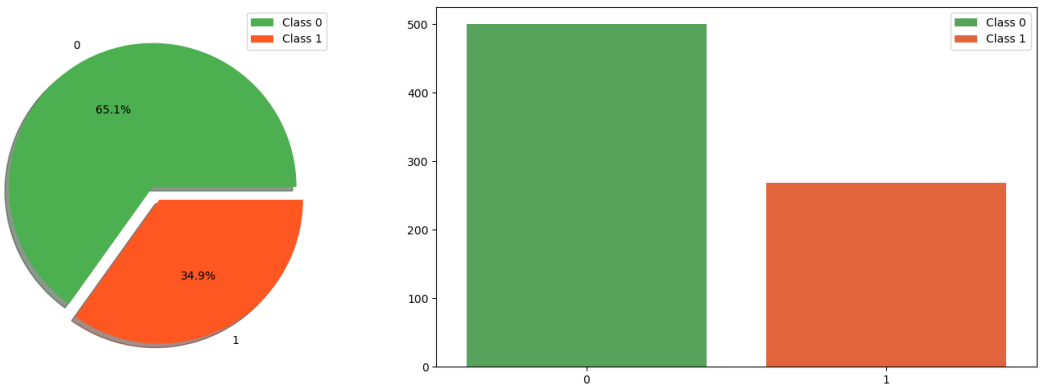
Pada penelitian ini, outlier tidak dihapus, melainkan ditangani dengan metode capping. Nilai outlier yang melebihi batas atas (*upper bound*) diganti dengan nilai maksimum yang masih berada dalam batas wajar, sedangkan nilai di bawah batas bawah (*lower bound*) diganti dengan nilai minimum yang valid. Hasil dari penanganan outlier ini bisa diamati pada Gambar 3.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550



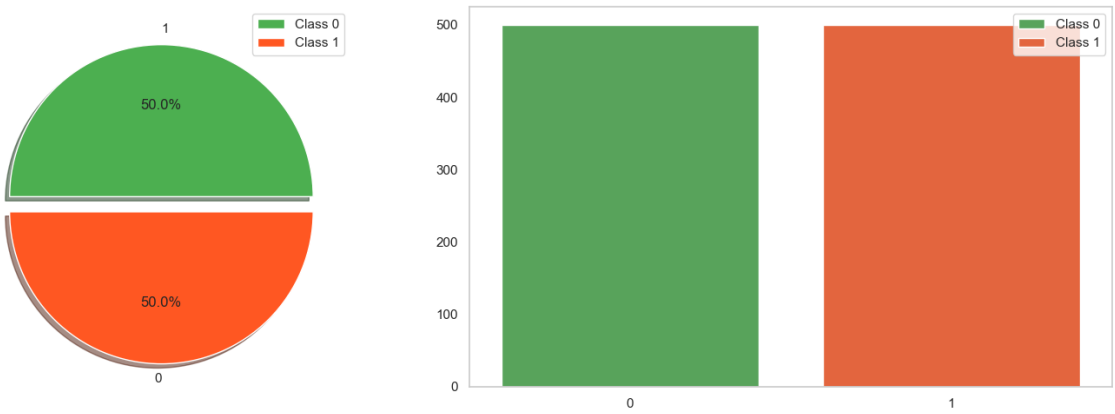
Gambar 3. Hasil penanganan outlier

Tahap selanjutnya, dilakukan pemeriksaan untuk memastikan apakah distribusi data sampel yang digunakan seimbang atau tidak. Hasil pemeriksaan, seperti yang terlihat pada Gambar 3, dapat disimpulkan bahwa dataset memiliki distribusi kelas yang tidak seimbang, dengan jumlah data sebagai berikut: kelas 0 sebanyak 500 sampel dan kelas 1 sebanyak 268 sampel.



Gambar 4. Distribusi data sebelum penyeimbangan

Untuk mengatasi masalah ketidakseimbangan kelas dalam dataset, dilakukan teknik oversampling, yaitu metode dalam pemrosesan data yang bertujuan untuk menyeimbangkan jumlah sampel di setiap kelas. Dalam penelitian ini, pendekatan yang diterapkan adalah SMOTE, yang menghasilkan sampel buatan berdasarkan data kelas 1. Setelah penerapan SMOTE, seperti yang ditunjukkan pada Gambar 5, distribusi data menjadi seimbang, dengan jumlah data di setiap kelas sebagai berikut: kelas 0 sebanyak 500 sampel dan kelas 1 sebanyak 500 sampel.



Gambar 5. Distribusi data setelah peyeimbangan dengan SMOTE

Selanjutnya, dataset dipisahkan menjadi variabel X (fitur) dan y (target). Setelah itu, data dibagi menggunakan metode splitting untuk memperoleh data latih dan data uji. Dalam penelitian ini, dataset dibagi dengan rasio 80:20, di mana 80% dipakai untuk melatih model dan 20% sisanya untuk pengujian. Perbandingan jumlah sampel sebelum dan sesudah penerapan SMOTE disajikan pada Tabel 6 di bawah ini.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

Tabel 6. Perbandingan jumlah sampel sebelum dan sesudah SMOTE

Kondisi	Data Training (80%)	Data Testing (20%)	Total Sampel
Sebelum SMOTE	614 (kelas 0: 400, Kelas 1: 214)	154 (Kelas 0: 100, Kelas 1: 54)	768
Setelah SMOTE	800 (Kelas 0: 400, Kelas 1:400)	200 (Kelas 0: 100, Kelas 1: 100)	1000

Tahapan terakhir adalah melakukan *feature scaling* menggunakan *StandardScaler* untuk menyetarakan skala atau rentang nilai dari data. Proses ini bertujuan untuk memastikan bahwa model tidak terpengaruh oleh perbedaan skala antar fitur, sehingga kinerja model menjadi lebih optimal. Tabel 7 menunjukkan perbandingan antara nilai asli dan nilai setelah dilakukan *feature scaling* pada beberapa fitur dataset.

Tabel 7. Nilai StandardScaler

Atribut	Nilai Asli	Nilai StandardScaler
Preg	6	-0,85584214
Glu	148	-1,0568694
BP	72	-0,85732472
Skin	35	-1,87940405
Ins	169,5	-1,55379191
BMI	33,6	-0,78065792
DPF	0,627	0,4005788
Age	50	-0,7989446

Pada Tabel 7, terlihat bahwa setelah penerapan *feature scaling* menggunakan StandardScaler, nilai dari fitur-fitur tersebut disesuaikan menjadi lebih seragam.

3.3 Implementasi Model

Penelitian Penelitian ini membandingkan performa model dengan empat kombinasi eksperimen, yaitu KNN tanpa SMOTE, KNN dengan SMOTE, DT tanpa SMOTE, dan DT dengan SMOTE. Untuk setiap eksperimen, model klasifikasi dikembangkan menggunakan algoritma KNN dan DT berdasarkan dataset yang telah melalui tahap preprocessing. Pada algoritma KNN, parameter yang dioptimalkan meliputi *n_neighbors* yang memiliki rentang nilai 3, 5, 7, dan 9; *weights* yang dapat berupa *uniform* atau *distance*; serta *metric* yang menggunakan *Euclidean*. Sementara itu, pada algoritma DT, parameter yang disesuaikan meliputi *max_depth* dengan opsi *None*, 10, 20, dan 30; *min_samples_split* dengan nilai 2, 5, dan 10; *min_samples_leaf* dengan nilai 1, 2, dan 4; *max_features* dengan opsi *None*, *sqrt*, dan *log2*; serta *criterion* dengan pilihan *gini* dan *entropy*.

Untuk mendapatkan kombinasi parameter terbaik, penelitian ini menggunakan metode pencarian hyperparameter melalui GridSearchCV dari library sklearn. GridSearchCV bekerja dengan menguji berbagai kombinasi hyperparameter yang telah ditentukan sebelumnya. Proses ini dilakukan dengan menggunakan cross-validation, yaitu metode yang membagi dataset menjadi beberapa subset (folds) untuk memastikan bahwa model diuji pada data yang berbeda setiap iterasi, sehingga hasil evaluasi menjadi lebih stabil dan mengurangi risiko overfitting. Setiap kombinasi parameter diuji berdasarkan metrik evaluasi yang telah ditentukan, seperti akurasi, presisi, recall, dan F1 score. Kombinasi parameter yang menghasilkan nilai evaluasi terbaik akan dipilih dan digunakan dalam implementasi model akhir. Hasil pencarian parameter terbaik disajikan dalam Tabel 8 dan Tabel 9. Tabel 8 menampilkan parameter terbaik untuk algoritma KNN, baik dengan maupun tanpa SMOTE, sedangkan Tabel 9 memuat parameter terbaik untuk algoritma DT, baik dengan maupun tanpa SMOTE.

Tabel 8. Parameter terbaik KNN

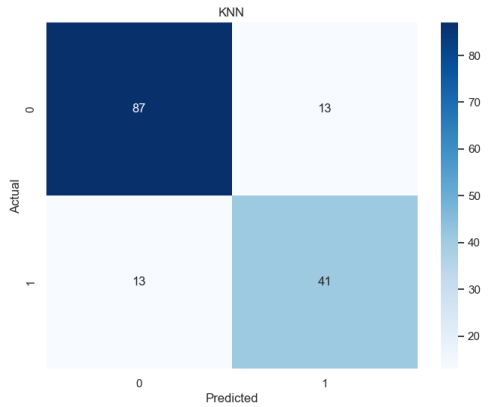
Penerapan	metric	n_neighbors	weights
KNN	Euclidean	7	Distance
KNN + SMOTE	Euclidean	5	Distance

Tabel 9. Parameter terbaik DT

Penerapan	criterion	max_depth	max_features	min_sample_leaf	min_sample_split
DT	entropy	none	none	1	2
DT + SMOTE	entropy	10	log2	4	10

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

3.4 Evaluasi Model

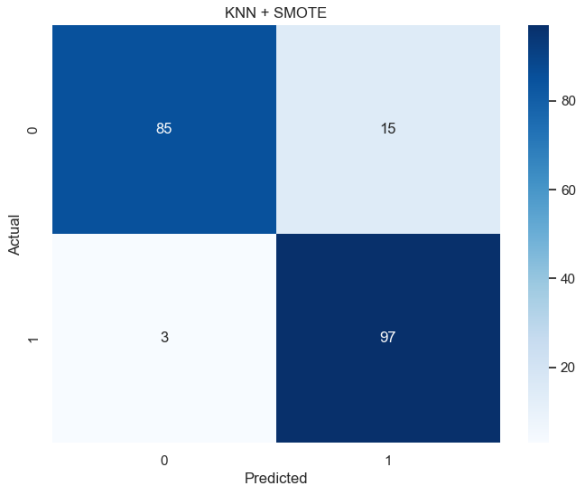


Gambar 6. Confusion Matrix KNN

Pada Gambar 6, ditampilkan hasil evaluasi performa algoritma KNN melalui confusion matrix. Dengan TP sebesar 87, TN sebesar 41, FP sebesar 13, dan FN sebesar 13, kita dapat menghitung beberapa Matrik evaluasi. Matrik ini membantu menilai kualitas model dalam memprediksi data, yang hasilnya dirangkum pada Tabel 10. Dari tabel itu, dapat dilihat bahwa nilai akurasi mencapai 83,12%, sementara nilai presisi, recall, dan F1 score masing-masing tercatat sebesar 87%.

Tabel 10. Hasil Evaluasi KNN

Parameter	Nilai
Akurasi	83,12%
Presisi	87%
Recall	87%
F1 Score	87%



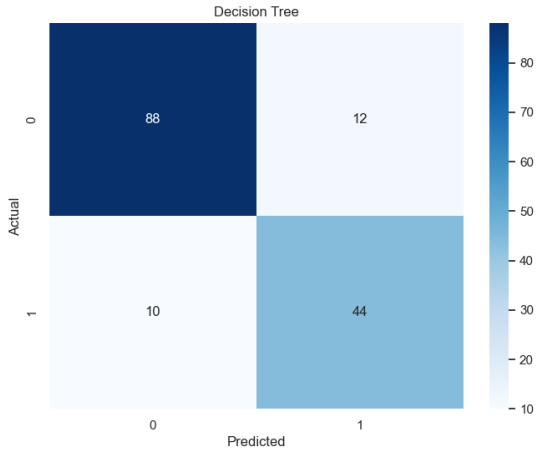
Gambar 7. Confusion Matrix KNN + SMOTE

Gambar 7 memperlihatkan analisis yang sama namun dengan penerapan teknik SMOTE pada algoritma KNN. Hasilnya terlihat cukup berbeda; TP sebesar 85 dan TN meningkat drastis menjadi 97, sementara FP dan FN menurun masing-masing menjadi 15 dan 3. Angka-angka ini mencerminkan dampak signifikan dari SMOTE dalam mengatasi ketidakseimbangan data. Matrik evaluasi yang dihitung, seperti yang ditampilkan pada Tabel 11, menunjukkan akurasi meningkat hingga 91%, dengan F1 score mencapai 90,43%, mencerminkan performa model yang lebih seimbang.

Tabel 11. Hasil Evaluasi KNN dengan SMOTE

Parameter	Nilai
Akurasi	91%
Presisi	85%
Recall	96,59%
F1 Score	90,43%

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

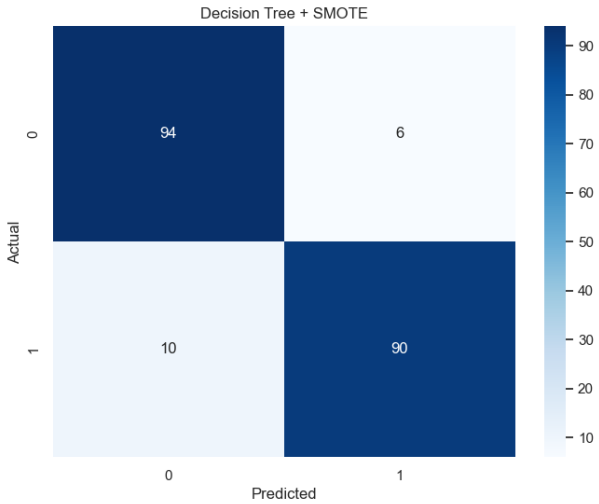


Gambar 8. Confusion Matrix DT

Kemudian, evaluasi pada algoritma DT disajikan melalui Gambar 8. Confusion matrix ini memberikan gambaran distribusi prediksi yang serupa dengan KNN. TP tercatat sebesar 88 dan TN sebesar 44, sementara P dan FN masing-masing sebesar 12 dan 10. Matrik yang dihasilkan, seperti yang tercantum pada Tabel 12, menunjukkan akurasi sebesar 85,71%, sementara nilai presisi, recall, dan F1 score yang relatif besar, berkisar di angka 88%-89%.

Tabel 12. Hasil Evaluasi KNN dengan SMOTE

Parameter	Nilai
Akurasi	85,71%
Presisi	88%
Recall	89,80%
F1 Score	88,89%



Gambar 9. Confusion Matrix DT + SMOTE

Terakhir, Gambar 9 menampilkan evaluasi DT setelah diterapkan teknik SMOTE. Hasilnya menunjukkan peningkatan performa model dengan TP sebesar 94 dan TN sebesar 90, sementara FP dan FN masing-masing turun menjadi 6 dan 10. Evaluasi matrik yang dirangkum dalam Tabel 13 mengungkapkan akurasi mencapai 92%, presisi 94%, serta F1 score yang meningkat menjadi 92,16%. Hal ini menggarisbawahi keefektifan SMOTE dalam meningkatkan performa model.

Tabel 13. Hasil Evaluasi KNN dengan SMOTE

Parameter	Nilai
Akurasi	92%
Presisi	94%
Recall	90,38%
F1 Score	92,16%

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

Secara keseluruhan, perbandingan hasil evaluasi KNN dan DT dengan dan tanpa SMOTE dapat dilihat dalam Tabel 14, yang memberikan gambaran performa masing-masing algoritma.

Tabel 14. Hasil Perbandingan Kinerja Algoritma KNN dan DT

Algoritma	Accuracy	Precision	Recall	F1 Score
KNN	83,12%	87%	87%	87%
KNN + SMOTE	91%	85%	96,59	90,43
DT	85,71%	88%	89,80%	88,89%
DT + SMOTE	92%	94%	90,38%	92,16%

4. KESIMPULAN

Berdasarkan hasil penelitian, implementasi algoritma KNN dan DT pada dataset PIDD menunjukkan perbedaan performa yang signifikan, terutama setelah penerapan SMOTE sebagai teknik penyeimbangan data. Algoritma DT secara konsisten menunjukkan performa yang lebih unggul dibandingkan KNN dalam hal akurasi dan stabilitas hasil. Pada KNN tanpa SMOTE, akurasi mencapai 83,12% dengan nilai presisi, recall, dan F1 score masing-masing sebesar 87%. Setelah SMOTE diterapkan, performa KNN meningkat drastis dengan akurasi 91%, recall 96,59%, dan F1 score 90,43%, yang mencerminkan dampak positif SMOTE dalam meningkatkan kualitas prediksi pada dataset tidak seimbang. Di sisi lain, algoritma DT tanpa SMOTE mencatat akurasi 85,71% dengan nilai presisi 88%, recall 89,80%, dan F1 score 88,89%. Penerapan SMOTE pada DT lebih lanjut meningkatkan akurasi menjadi 92%, dengan presisi 94%, recall 90,38%, dan F1 score 92,16%, menunjukkan keunggulan DT dalam mengolah dataset yang telah seimbang. Secara keseluruhan, hasil penelitian ini menunjukkan bahwa DT lebih superior dibandingkan KNN dalam semua matrik evaluasi, baik sebelum maupun sesudah penerapan SMOTE. Keunggulan DT dibandingkan KNN dalam penelitian ini dapat dijelaskan dari sifat Decision Tree yang lebih stabil dalam mengatasi data tidak seimbang. DT mampu secara otomatis menyesuaikan struktur pohonnya dengan dataset yang telah seimbang, sehingga dapat membuat pemisahan yang lebih jelas antara kelas mayoritas dan minoritas. Selain itu, DT tidak terlalu dipengaruhi oleh pemilihan parameter awal dibandingkan dengan KNN, yang sangat bergantung pada pemilihan jumlah tetangga terdekat (n_neighbors) dan metode pembobotan (weights). Penerapan SMOTE terbukti memberikan dampak signifikan pada peningkatan kinerja kedua algoritma, meskipun efeknya lebih konsisten pada DT. Temuan ini memberikan panduan penting bagi peneliti dan praktisi dalam memilih algoritma dan teknik balancing data yang tepat untuk mengoptimalkan hasil analisis pada dataset dengan distribusi kelas tidak seimbang.

REFERENCES

[1] R. Sianturi and A. Mustofa, "Aerobic Exercise Reduce Blood Glucose in Type 2 Diabetes Mellitus," *Media Keperawatan Indonesia*, vol. 5, no. 1, p. 73, Feb. 2022, doi: 10.26714/mki.5.1.2022.73-83.

[2] World Health Organization, "Diabetes." Access Date Sept 2024

[3] International Diabetes Federation, *IDF Diabetes Atlas*, 10th ed. Brussels, Belgium, 2021.

[4] Q. A. Puteri, T. Sagirani, and J. Lemantara, "Perbandingan Algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) untuk Mengetahui Keakuratan Diagnosa Penyakit Diabetes," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, no. 3, pp. 247–254, Dec. 2023, doi: 10.25077/teknosi.v9i3.2023.247-254.

[5] A. Perdana, A. Hermawan, and D. Avianto, "Analyze Important Features of PIMA Indian Database For Diabetes Prediction Using KNN," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 70–75, Mar. 2023, doi: 10.32736/sisfokom.v12i1.1598.

[6] N. N. Habibah, A. Nazir, I. Iskandar, F. Syafria, L. Oktavia, and I. Syurfi, "Pemodelan Klasifikasi Untuk Menentukan Penyakit Diabetes dengan Faktor Penyebab Menggunakan Decision Tree C4.5 Pada Wanita," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 4, p. 654, Jun. 2023, doi: 10.30865/json.v4i4.6202.

[7] N. Aggarwal, C. Bagath Basha, A. Arya, and N. Gupta, "A Comparative Analysis of Machine Leaming-Based Classifiers for Predicting Diabetes," in *Proceedings - 2023 International Conference on Advanced Computing and Communication Technologies, ICACCTech 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 615–621. doi: 10.1109/ICACCTech61146.2023.00105.

[8] P. Sarkar and S. Pawar, "Machine Learning based Early Predication and Detection of Diabetes Mellitus," in *International Conference on Artificial Intelligence for Innovations in Healthcare Industries, ICAIIHI 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICAIIHI57871.2023.10489259.

[9] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *ICT Express*, vol. 7, no. 4, pp. 432–439, Dec. 2021, doi: 10.1016/j.icte.2021.02.004.

[10] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, "Diabetes prediction using supervised machine learning," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 21–30. doi: 10.1016/j.procs.2022.12.107.

[11] E. Safitri, D. Rofianto, N. Purwati, H. Kurniawan, and S. Karnila, "Prediksi Penyakit Diabetes Melitus Menggunakan Algoritma Machine Learning," JUSTIN, Vol. 12, No. 4, 2024, doi: 10.26418/justin.v12i4.84620.

[12] B. Amma N.G., "En-RfRsK: An ensemble machine learning technique for prognostication of diabetes mellitus," *Egyptian Informatics Journal*, vol. 25, Mar. 2024, doi: 10.1016/j.eij.2024.100441.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2601−2611
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.6550

[13] P. V. K. Rao, Aarti, and A. S. Rao, "Machine Learning Approaches for Diabetes Prediction: Comparative Analysis and Pre-processing Insights," in *Proceedings - 2024 8th International Conference on Inventive Systems and Control, ICISC 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 39–46. doi: 10.1109/ICISC62624.2024.00014.

[14] C. Haryawan, Y. Muria Kusuma Ardhana, "Analisa Perbandingan Teknik Oversampling Smote Pada Imbalanced Data," JIRE, Vol 6, No 1, 2023. doi: 10.36595/jire.v6i1.834.

[15] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020, doi: 10.1109/ACCESS.2020.2986809.

[16] S. Sofyan and A. Prasetyo, "Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang Pada Tingkat Pendapatan Pekerja Informal Di Provinsi D.I. Yogyakarta Tahun 2019," Seminar Nasional Official Statistics, 2021. doi: https://doi.org/10.34123/semnasoffstat.v2021i1.1081.

[17] A. Nugroho and E. Rilvani, "Penerapan Metode Oversampling SMOTE Pada Algoritma Random Forest Untuk Prediksi Kebangkrutan Perusahaan Application of the SMOTE Oversampling Method to the Random Forest Algorithm for Predicting Company Bankruptcy." Jurnal Teknologi Informasi, Vol 11, No 1, 2023, doi: 10.33633/tc.v22i1.7527.

[18] A. Andreyestha and Q. N. Azizah, "Analisa Sentimen Kicauan Twitter Tokopedia Dengan Optimalisasi Data Tidak Seimbang Menggunakan Algoritma SMOTE," *Infotek : Jurnal Informatika dan Teknologi*, vol. 5, no. 1, pp. 108–116, Jan. 2022, doi: 10.29408/jit.v5i1.4581.

[19] E. Rahmawati and C. Agustina, "Optimasi Ulasan Pengguna Aplikasi ChatGPT di Google Play Store Menggunakan SMOTE," J-TIT, vol 11, no 1, 2024. [Online]. Available: https://doi.org/10/25047/jtit.v11i1.360

[20] M. Persada Pulungan, A. Purnomo, A. Kurniasih, "Penerapan Smote Untuk Mengatasi Imbalance Class Dalam Klasifikasi Kepribadian Mbti Menggunakan Naive Bayes Classifier Application Of Smote To Overcome Class Imbalance In The Mbti Personality Classification Using The Naïve Bayes Classifier" JTIK, Vol 11, No 5, 2024, doi: 10.25126/jtiik.2024117989.

[21] A. Surya Firmansyah, A. Aziz, and M. Ahsan, "Optimasi K-Nearest Neighbor Menggunakan Algoritma Smote Untuk Mengatasi Imbalance Class Pada Klasifikasi Analisis Sentimen," JATI, Vol 7, No 6, 2023. doi: https://doi.org/10.36040/jati.v7i6.7257.

[22] M. Syukron, R. Santoso, and T. Widiharih, "Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data," *Jurnal Gaussian*, vol. 9, no. 3, pp. 227–236, Aug. 2020, doi: 10.14710/j.gauss.9.3.227-236.

[23] A. W. Ishlah, S. Sudarno, and P. Kartikasari, "Implementasi Gridsearchcv Pada Support Vector Regression (Svr) Untuk Peramalan Harga Saham," *Jurnal Gaussian*, vol. 12, no. 2, pp. 276–286, Jul. 2023, doi: 10.14710/j.gauss.12.2.276-286.