Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

Optimasi Model Particle Swarm Optimization (PSO) Menggunakan SMOTE Untuk Menentukan Penyakit Diabetes Mellitus

Satrio Allam Putro Utomo, Defri Kurniawan*

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia
Email: [1]111202113616@mhs.dinus.ac.id, [2,*]defri.kurniawan@dsn.dinus.ac.id
Email Penulis Korespondensi: defri.kurniawan@dsn.dinus.ac.id
Submitted: **11/03/2025**; Accepted: **25/03/2025**; Published**: 26/03/2025**

Abstrak−Diabetes mellitus merupakan penyakit kronis yang terus mengalami peningkatan secara global dan dapat memengaruhi berbagai kelompok usia. Jika tidak dikelola dengan baik, penyakit ini dapat menyebabkan komplikasi serius. Dalam beberapa tahun terakhir, perkembangan teknologi, terutama dalam bidang pembelajaran mesin (machine learning), telah memberikan kontribusi besar dalam meningkatkan akurasi diagnosis dan prediksi penyakit diabetes. Penelitian ini memanfaatkan algoritma Decision Tree yang ditingkatkan dengan dua metode optimasi, yaitu Synthetic Minority Over-sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan data dan Particle Swarm Optimization (PSO) untuk mengoptimalkan hyperparameter model, sehingga meningkatkan akurasi klasifikasi. Data yang digunakan dalam penelitian ini diperoleh dari Diabetes Prediction Dataset yang tersedia di Kaggle, dengan jumlah entri mencapai 100.000 data. Berdasarkan hasil analisis, penerapan preprocessing data dan optimasi hyperparameter terbukti mampu meningkatkan akurasi model dari 95.21% menjadi 96.52%. Selain itu, evaluasi menggunakan confusion matrix menunjukkan adanya peningkatan precision dari 70.82% menjadi 86.19% serta peningkatan F1-score dari 72.49% menjadi 78.52%, meskipun terjadi sedikit penurunan pada nilai recall dari 74.24% menjadi 72.11%. Hasil ini membuktikan bahwa kombinasi antara preprocessing data, penyeimbangan data, dan optimasi hyperparameter dapat secara signifikan meningkatkan performa model klasifikasi dalam mendeteksi diabetes. Untuk pengembangan lebih lanjut, disarankan agar model diuji pada dataset lain guna meningkatkan kemampuan generalisasi. Selain itu, eksplorasi terhadap algoritma tambahan seperti Random Forest atau XGBoost dapat dilakukan untuk memperoleh hasil yang lebih optimal.

Kata Kunci: Diabetes Mellitus; Pohon Keputusan; Synthetic Minority Over-sampling Technique; Particle Swarm Optimization; Prediksi

Abstract−Diabetes mellitus is a chronic disease that continues to increase globally and can affect various age groups. If not properly managed, this disease can lead to serious complications. In recent years, technological advancements, particularly in the field of machine learning, have significantly contributed to improving the accuracy of diabetes diagnosis and prediction. This study utilizes the Decision Tree algorithm, enhanced by two optimization methods: the Synthetic Minority Over-sampling Technique (SMOTE) to address data imbalance and Particle Swarm Optimization (PSO) to optimize the model's hyperparameters, thereby improving classification accuracy. The dataset used in this study is the Diabetes Prediction Dataset available on Kaggle, consisting of 100,000 entries. Based on the analysis results, the implementation of data preprocessing and hyperparameter optimization has proven to increase the model's accuracy from 95.21% to 96.52%. Additionally, an evaluation using the confusion matrix shows an improvement in precision from 70.82% to 86.19% and an increase in the F1-score from 72.49% to 78.52%, although there is a slight decrease in recall from 74.24% to 72.11%. These findings demonstrate that a combination of data preprocessing, data balancing, and hyperparameter optimization can significantly enhance the performance of a classification model in detecting diabetes. For future development, it is recommended that the model be tested on other datasets to improve generalizability. Furthermore, exploring additional algorithms such as Random Forest or XGBoost could be beneficial in obtaining more optimal results.

Keywords: Diabetes Mellitus; Decision Tree; Synthetic Minority Over-sampling Technique; Particle Swarm Optimization; Prediction

1. PENDAHULUAN

Kesehatan merupakan salah satu aspek yang sangat mendasar dan memiliki peranan krusial dalam kehidupan manusia, karena tanpa kondisi tubuh yang sehat, seseorang tidak dapat menjalankan aktivitas sehari-hari secara optimal. Oleh sebab itu, berbagai upaya perlindungan serta tindakan pencegahan terhadap munculnya gangguan kesehatan dan penyakit yang berpotensi berbahaya terus menjadi perhatian utama dalam dunia medis dan kebijakan kesehatan masyarakat. Akan tetapi, dalam beberapa tahun terakhir, terjadi peningkatan signifikan dalam jumlah kasus penyakit kronis dan genetik yang berpengaruh terhadap kualitas kesehatan masyarakat secara keseluruhan. Salah satu penyakit kronis yang banyak ditemukan di berbagai belahan dunia adalah diabetes [1]. Diabetes mellitus, yang lebih dikenal oleh masyarakat luas sebagai penyakit kencing manis, merupakan kondisi medis yang dapat mengganggu sistem metabolisme tubuh manusia. Penyakit ini muncul akibat meningkatnya kadar glukosa dalam darah, yang disebabkan oleh ketidakmampuan organ pankreas dalam memproduksi insulin dalam jumlah yang cukup. Padahal, insulin sangat diperlukan oleh tubuh untuk membantu proses penyerapan gula dari makanan yang dikonsumsi sehari-hari agar dapat digunakan sebagai sumber energi[2]. Diabetes tidak hanya menjadi persoalan kesehatan yang dihadapi secara individu oleh penderitanya, tetapi juga berkembang menjadi tantangan kesehatan global yang memengaruhi berbagai kelompok usia, mulai dari anak-anak hingga orang dewasa. Berdasarkan data yang tersedia, diperkirakan sekitar 1,2 juta anak dan remaja di seluruh dunia telah didiagnosis menderita diabetes, sebuah angka yang menunjukkan adanya peningkatan signifikan dalam prevalensi penyakit ini dalam beberapa dekade terakhir [3]. Selain itu, diabetes telah menjadi salah satu krisis kesehatan terbesar yang dihadapi dunia saat ini. Dalam kurun waktu dua dekade terakhir,

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

jumlah penderita diabetes mengalami lonjakan drastis, yaitu meningkat lebih dari tiga kali lipat sejak tahun 2000. Pada tahun 2000, jumlah penderita diabetes tercatat sebanyak 151 juta orang, sedangkan pada tahun 2021 angka ini melonjak drastis menjadi 537 juta penderita. Bahkan, berdasarkan proyeksi terbaru, jumlah individu yang hidup dengan diabetes diperkirakan akan terus bertambah hingga mencapai 783 juta orang pada tahun 2045 [4].

Pada diabetes, tubuh tidak memproduksi hormon insulin dalam jumlah yang memadai, yang berperan untuk membantu, mengarahkan, dan memerintahkan sel-sel tubuh menyerap gula "glukosa" dari darah. Diabetes terbagi menjadi tiga jenis utama: Tipe I, Tipe II, dan gestasional. Tipe I lebih sering terjadi pada anak-anak berusia 4 hingga 7 tahun dan 10 hingga 14 tahun. Diabetes gestasional (Tipe III) muncul saat kehamilan dan umumnya hilang setelah melahirkan. Tipe II adalah jenis diabetes yang paling umum [5]. Oleh karena itu, seseorang yang menderita diabetes dan tidak mampu mengontrol kadar gula darahnya dengan baik akan berada dalam kondisi yang sangat rentan terhadap berbagai komplikasi serius. Komplikasi ini tidak hanya berdampak pada satu aspek kesehatan saja, tetapi juga dapat memengaruhi berbagai organ vital dalam tubuh, seperti jantung, ginjal, mata, dan sistem saraf. Jika tidak ditangani dengan tepat, risiko kesehatan yang ditimbulkan oleh diabetes yang tidak terkontrol dapat meningkat secara signifikan, bahkan berpotensi menyebabkan kematian dini dalam jumlah yang lebih tinggi dibandingkan dengan individu yang memiliki kadar gula darah terkelola dengan baik [6].

Seiring dengan meningkatnya jumlah penderita diabetes di seluruh dunia, pemanfaatan teknologi canggih, seperti pembelajaran mesin (*machine learning*), telah menjadi solusi inovatif yang semakin relevan untuk diterapkan dalam bidang kesehatan. Dalam beberapa tahun terakhir, perkembangan pesat dalam teknologi pembelajaran mesin telah menjadikannya sebagai alat yang andal dalam mendukung berbagai aplikasi medis. Teknologi ini berperan penting dalam meningkatkan akurasi diagnosis, efektivitas metode perawatan, serta kemampuan dalam memprediksi risiko kesehatan secara lebih presisi [7]. Pembelajaran mesin bekerja dengan menggunakan berbagai algoritma yang secara umum dapat diklasifikasikan ke dalam dua kategori utama, yaitu *supervised learning* dan *unsupervised learning*. Kedua pendekatan ini berfungsi untuk membantu proses analisis data, melakukan prediksi, serta mendukung pengambilan keputusan secara otomatis berdasarkan informasi yang tersedia [8]. Dalam *supervised learning*, algoritma dilatih menggunakan data berlabel, sehingga dapat mengenali pola tertentu yang nantinya diterapkan dalam proses prediksi atau klasifikasi. Sebaliknya, dalam *unsupervised learning*, algoritma digunakan untuk menganalisis dan mengidentifikasi pola tersembunyi atau struktur tertentu dalam data yang tidak memiliki label, sehingga memungkinkan eksplorasi lebih lanjut terhadap hubungan kompleks di dalam dataset [9].

Penelitian ini menggunakan algoritma pembelajaran *supervised learning* dalam pengembangan model, karena memanfaatkan dataset berlabel sebagai dasar analisis. Dataset yang digunakan dalam penelitian ini diperoleh dari platform *Kaggle*, yaitu *Diabetes Prediction Dataset*, yang berisi data medis serta informasi demografis pasien [10]. Dataset ini mencakup total 100.000 entri data dengan 8 variabel prediktor dan 1 variabel target yang digunakan untuk menentukan apakah seorang individu terindikasi mengidap diabetes atau tidak. Namun, dataset ini mengalami ketidakseimbangan kelas (*class imbalance*), yang dapat memengaruhi kinerja model klasifikasi. Dari total data yang tersedia, sebanyak 8.500 data memiliki variabel target bernilai (1), yang menunjukkan individu terindikasi menderita diabetes, sedangkan 91.500 data memiliki variabel target bernilai (2), menunjukkan individu yang tidak terindikasi memiliki diabetes. Ketimpangan dalam distribusi data ini berpotensi menyebabkan penurunan sensitivitas model serta nilai *Area Under Curve* (AUC), karena model cenderung lebih teroptimasi untuk kelas mayoritas dan kurang akurat dalam mengklasifikasikan kelas minoritas [11]. Untuk mengatasi permasalahan ketidakseimbangan ini, penelitian ini menerapkan metode *Synthetic Minority Over-sampling Technique* (SMOTE). Berdasarkan penelitian sebelumnya, metode SMOTE terbukti efektif dalam meningkatkan kinerja model klasifikasi secara keseluruhan dengan menghasilkan sampel sintetis untuk kelas minoritas. Teknik ini memperkaya informasi pada kategori yang kurang terwakili, sehingga model dapat mengidentifikasi pola dengan lebih baik dan meningkatkan keandalan serta akurasinya dalam memprediksi kasus diabetes [12].

Berbagai penelitian sebelumnya telah membahas penerapan metode *machine learning* dalam melakukan prediksi terhadap penyakit diabetes. Penelitian yang dilakukan oleh Wisti Dwi Septiani dan Marlina pada tahun 2021 mengungkapkan bahwa penggunaan algoritma *Decision Tree* memiliki kinerja yang unggul dalam memprediksi diabetes, dengan tingkat akurasi mencapai 95,96%. Angka ini lebih tinggi dibandingkan dengan algoritma *Naïve Bayes*, yang menghasilkan akurasi sebesar 87,69%, serta *Neural Network*, yang hanya mencapai akurasi 61,54% [13]. Selanjutnya, penelitian yang dilakukan oleh Dela Rista Damayanti dan Aji Purwinarko pada tahun 2024 menunjukkan bahwa pada awalnya, algoritma Decision Tree memiliki tingkat akurasi sebesar 75,97%. Namun, setelah diterapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dan *Particle Swarm Optimization* (PSO), akurasinya meningkat menjadi 82,5% [14]. Kemudian, penelitian yang dilakukan oleh Muningsih *et al.* pada tahun 2024, yang berfokus pada optimasi model *Decision Tree*, berhasil memperoleh akurasi yang lebih tinggi lagi, yakni sebesar 97% [15]. Selain itu, penelitian yang dilakukan oleh Mehrpout *et al.* pada tahun 2023 menunjukkan bahwa model *Decision Tree* tidak hanya memiliki akurasi yang tinggi, tetapi juga menunjukkan kinerja yang unggul dalam beberapa metrik evaluasi lainnya. Model ini memiliki *recall* sebesar 93,3%, *specificity* 92,8%, *precision* 93,4%, *F1-score* 93,3%, serta *accuracy* sebesar 93,3% [16]. Lebih lanjut, penelitian yang dilakukan oleh Baiq Andriska Candra Permana dan Intan Komala Dewi pada tahun 2021 kembali menegaskan keunggulan algoritma *Decision Tree* dalam memprediksi diabetes dengan akurasi sebesar 95,58%, yang masih lebih tinggi dibandingkan dengan algoritma *Naïve Bayes*, yang hanya mencapai akurasi 87,69% [17].

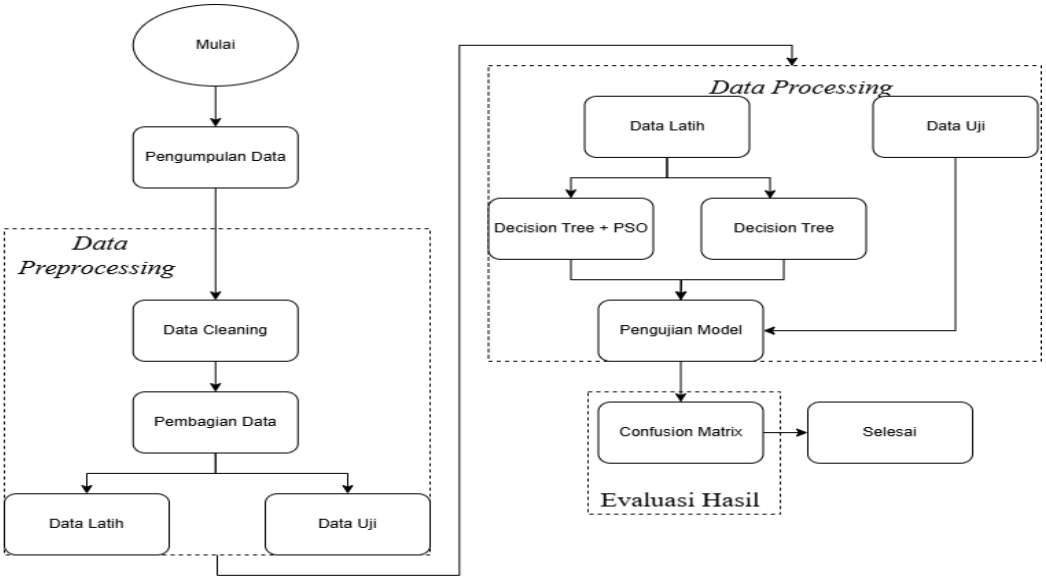
Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

Merujuk pada berbagai penelitian sebelumnya, keunggulan utama dari model *Decision Tree* terletak pada kemampuannya dalam menyajikan proses pengambilan keputusan yang intuitif dan mudah dipahami. Selain itu, model ini memiliki fleksibilitas yang tinggi dalam menangani berbagai jenis data, baik yang bersifat numerik maupun kategorikal. Berdasarkan karakteristik unggul tersebut, penelitian ini memilih untuk menerapkan algoritma *Decision Tree* sebagai metode utama dalam membangun model klasifikasi. Salah satu alasan utama pemilihan algoritma ini adalah kecepatan pemrosesannya yang tinggi, yang memungkinkan analisis data dilakukan secara efisien. Selain itu, kemudahan dalam menginterpretasikan hasil klasifikasi menjadikan *Decision Tree* sebagai algoritma yang dapat digunakan oleh berbagai kalangan, termasuk pengguna yang tidak memiliki latar belakang teknis yang mendalam [18]. Namun, di balik keunggulannya, *Decision Tree* juga memiliki beberapa kelemahan. Salah satu tantangan utama yang dihadapi algoritma ini adalah ketergantungannya pada struktur pohon yang terorganisir dengan baik. Jika atribut yang digunakan tidak sesuai atau desain pohon keputusan tidak optimal, kualitas model yang dihasilkan dapat menurun secara signifikan. Hal ini dapat menyebabkan kesalahan dalam akumulasi nilai di setiap tingkat pohon keputusan, terutama ketika pohon yang dibangun terlalu besar dan kompleks [19].

Sebagai solusi untuk mengatasi permasalahan yang muncul akibat ketidakseimbangan struktur pohon dalam *Decision Tree*, metode *Particle Swarm Optimization* (PSO) diterapkan sebagai upaya untuk mengoptimalkan hyperparameter model. Dengan menerapkan PSO, diharapkan akurasi serta stabilitas model klasifikasi dapat meningkat secara signifikan. Penelitian terdahulu menunjukkan bahwa algoritma *Naïve Bayes* tanpa optimasi menghasilkan tingkat akurasi sebesar 86,80%. Namun, setelah dilakukan optimasi menggunakan PSO, akurasi model meningkat menjadi 89,84%, yang membuktikan efektivitas metode ini dalam meningkatkan performa model klasifikasi [20]. Selain itu, penelitian lain mengungkapkan bahwa penerapan PSO juga berdampak positif pada berbagai algoritma *machine learning*. Setelah dioptimalkan dengan PSO, akurasi algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbors* (KNN) mengalami peningkatan signifikan hingga mencapai 98,3%. Sementara itu, model *Decision Tree* yang telah dioptimalkan menunjukkan akurasi sebesar 97,77%, sedangkan *Naïve Bayes* mencatat akurasi terendah, yaitu sebesar 69,30% [21]. Berdasarkan temuan tersebut, penerapan PSO dalam penelitian ini diharapkan dapat meningkatkan akurasi model klasifikasi diabetes secara signifikan. Dalam penelitian ini, PSO digunakan untuk mengoptimalkan *hyperparameter* pada model *Decision Tree* guna meningkatkan akurasi dan stabilitas prediksi. Proses optimasi dilakukan dengan mendefinisikan partikel dalam PSO sebagai sekumpulan kandidat nilai *hyperparameter*, seperti *max_depth*, *min_samples_split*, dan *min_samples_leaf*. Dengan metode ini, PSO memungkinkan pemilihan *hyperparameter* secara otomatis tanpa perlu melakukan pencarian manual, sehingga meningkatkan efisiensi dan efektivitas dalam membangun model klasifikasi diabetes yang lebih optimal dan akurat.

2. METODOLOGI PENELITIAN

Penelitian ini bertujuan untuk melakukan klasifikasi guna menentukan apakah seseorang memiliki indikasi penyakit diabetes atau tidak. Berdasarkan hasil tinjauan literatur yang telah dilakukan, metode *Synthetic Minority Over-sampling Technique* (SMOTE) akan diterapkan sebagai solusi dalam menangani ketidakseimbangan data pada dataset yang digunakan. Selain itu, algoritma *Decision Tree* dipilih sebagai metode utama dalam proses identifikasi, mengingat kemampuannya dalam menyajikan hasil klasifikasi yang mudah dipahami. Untuk meningkatkan akurasi serta stabilitas model, penelitian ini juga menerapkan *Particle Swarm Optimization* (PSO) dalam mengoptimalkan parameter algoritma, sehingga diharapkan dapat menghasilkan prediksi yang lebih akurat.



Gambar 1. Alur Penelitian

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

Sebagai gambaran keseluruhan mengenai tahapan penelitian, diagram alur penelitian ditampilkan pada Gambar 1. Diagram ini berfungsi untuk memberikan visualisasi yang lebih jelas terkait langkah-langkah yang dilakukan dalam menyelesaikan permasalahan serta mencapai tujuan penelitian.

Gambar 1 menggambarkan alur penelitian yang dilakukan dalam penelitian ini. Proses penelitian diawali dengan tahap pengumpulan data sebagai langkah awal dalam membangun model klasifikasi. Setelah data terkumpul, dilakukan tahap *preprocessing* yang mencakup beberapa prosedur penting, seperti *data cleaning* serta pembagian dataset menjadi *training set* dan *testing set* guna memastikan model dapat dilatih dan diuji secara optimal. Setelah tahap *preprocessing* selesai, data yang telah diproses digunakan dalam tahap pelatihan dengan melakukan penyeimbangan distribusi data (*balancing*) menerapkan metode *Synthetic Minority Over-sampling Technique* (SMOTE), dan pengujian model. Dalam penelitian ini, algoritma *Decision Tree* diterapkan sebagai metode utama dalam proses klasifikasi guna mengidentifikasi kemungkinan seseorang menderita diabetes. Pada tahap akhir, evaluasi model dilakukan menggunakan *confusion matrix* dan *classification report* sebagai metrik utama untuk mengukur kinerja algoritma yang diterapkan. Evaluasi ini mencakup pengukuran akurasi, presisi, sensitivitas (*recall*), dan nilai *F1-score*, yang bertujuan untuk menilai sejauh mana model mampu melakukan prediksi dengan tingkat kesalahan yang minimal.

2.1. Pengumpulan Data

Tahapan awal dalam penelitian ini dimulai dengan proses pengumpulan data, yang merupakan langkah fundamental dalam memperoleh informasi yang akurat dan relevan guna mendukung proses analisis serta pengembangan model penelitian[22]. Data yang digunakan dalam penelitian ini bersumber dari *Diabetes Prediction Dataset*, yang diperoleh dari platform Kaggle dan berisi informasi medis serta data demografis pasien [10]. Dataset ini mencakup total 100.000 entri data, yang terdiri dari 8 variabel prediktor dan 1 variabel target yang bertujuan untuk mengklasifikasikan apakah seorang individu memiliki indikasi diabetes atau tidak. Variabel target dalam dataset ini dikategorikan dengan nilai (1) untuk individu yang teridentifikasi sebagai penderita diabetes dan nilai (0) untuk individu yang tidak mengidap penyakit tersebut. Namun, dataset ini memiliki tantangan berupa ketidakseimbangan kelas yang cukup signifikan, di mana hanya 8,8% dari total data atau sekitar 8.800 individu yang diklasifikasikan sebagai penderita diabetes, sementara 91,2% atau sekitar 91.200 individu termasuk dalam kategori non-diabetes. Ketidakseimbangan distribusi data ini dapat berpengaruh terhadap kinerja model klasifikasi yang dibangun, sehingga diperlukan metode penyesuaian untuk meningkatkan akurasi dan keandalan hasil prediksi.

2.2 *Data Preprocessing*

Tahapan *Preprocessing Data* merupakan bagian krusial dalam proses *data mining* yang bertujuan untuk meningkatkan kualitas data sehingga hasil analisis dapat menjadi lebih akurat dan model klasifikasi dapat dilatih secara lebih optimal [23]. Proses ini mencakup serangkaian langkah sistematis yang melibatkan pembersihan, pengorganisasian, serta transformasi data mentah agar lebih sesuai untuk digunakan dalam analisis lebih lanjut. Dengan menerapkan *preprocessing*, data yang sebelumnya mungkin mengandung inkonsistensi, duplikasi, atau nilai yang hilang dapat diolah sedemikian rupa sehingga menghasilkan dataset yang lebih terstruktur dan representatif. Langkah ini sangat penting terutama dalam model prediktif, di mana kualitas data yang digunakan secara langsung memengaruhi performa dan akurasi hasil klasifikasi.

2.2.1 *Data Cleaning*

Tahapan *Data Cleaning* merupakan langkah awal yang bertujuan untuk memastikan bahwa dataset yang digunakan dalam penelitian ini bebas dari kesalahan atau ketidaksesuaian data, sehingga dapat meningkatkan akurasi analisis dan performa model yang akan digunakan [24]. Dalam penelitian ini, proses *data cleaning* dimulai dengan melakukan pengecekan terhadap adanya nilai yang hilang atau kosong (*missing values*) dalam dataset yang digunakan. Berdasarkan hasil pengecekan, tidak ditemukan adanya *missing values*, yang menunjukkan bahwa seluruh atribut dalam dataset telah terisi dengan lengkap. Namun, terdapat data duplikat yang berpotensi menyebabkan bias dalam analisis dan pelatihan model. Untuk mengatasi permasalahan ini, diterapkan metode penghapusan data duplikat agar dataset yang digunakan menjadi lebih bersih dan akurat. Langkah ini dilakukan guna memastikan bahwa setiap entri dalam dataset bersifat unik, sehingga model dapat belajar dari data yang benar-benar representatif tanpa adanya redundansi informasi yang dapat memengaruhi hasil klasifikasi.

2.2.2 Pembagian Data

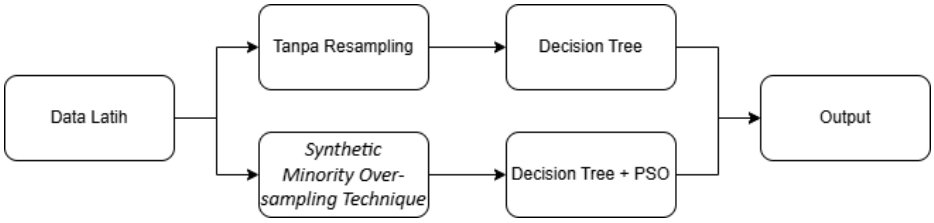
Data yang telah melewati serangkaian tahap *preprocessing* selanjutnya memasuki proses pembagian data, yang merupakan salah satu langkah krusial dalam pembangunan model pembelajaran mesin. Pembagian ini bertujuan untuk memastikan bahwa model yang dikembangkan dapat diuji menggunakan data yang belum pernah digunakan dalam proses pelatihan sebelumnya, sehingga kinerja model dapat dievaluasi secara objektif [25]. Dalam penelitian ini, data dibagi menjadi dua bagian dengan rasio 80:20.

2.3 *Data Processing*

Pada tahap ini, pengembangan model prediksi dilakukan dengan menerapkan algoritma Decision Tree melalui dua pendekatan yang berbeda. Setiap pendekatan dirancang untuk mengeksplorasi efektivitas dan kinerja model dalam

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

menghasilkan prediksi yang lebih akurat. Dalam hal ini, proses pemodelan melibatkan serangkaian langkah mulai dari pemrosesan data, penerapan algoritma, hingga evaluasi hasil guna menentukan metode yang paling optimal. Untuk memberikan gambaran yang lebih jelas mengenai perbedaan antara kedua pendekatan tersebut, Gambar 2 menyajikan ilustrasi perbandingan dari masing-masing metode yang digunakan dalam penelitian ini.



Gambar 2. *Data Processing*

Pada Gambar 2 pendekatan pertama menggunakan data latih yang belum melalui tahap preprocessing, di mana model dilatih dengan base model Decision Tree yang mengandalkan data mentah untuk mengevaluasi performanya terhadap data tanpa penyesuaian lebih lanjut. Pendekatan kedua, sebaliknya, melibatkan serangkaian tahap preprocessing data menggunakan teknik Synthetic Minority Over-sampling Technique (SMOTE) guna mengatasi ketidakseimbangan distribusi kelas dalam dataset. Setelah proses preprocessing selesai, model dilatih menggunakan Decision Tree dan selanjutnya dioptimalkan dengan algoritma Particle Swarm Optimization (PSO) untuk mencari kombinasi hyperparameter terbaik yang dapat meningkatkan akurasi dan stabilitas model. Kedua pendekatan ini kemudian dievaluasi menggunakan data uji yang sama, sehingga hasil kinerja dari masing-masing metode dapat dibandingkan secara objektif. Perbandingan hasil ini bertujuan untuk mengidentifikasi sejauh mana tahap preprocessing data dan teknik pengoptimalan lanjutan dapat berkontribusi dalam meningkatkan kinerja model, baik dari segi akurasi prediksi maupun kemampuannya dalam mengklasifikasikan data secara lebih efektif.

2.3.1 *Decision Tree*

Decision Tree merupakan salah satu model prediktif yang banyak digunakan dalam tugas klasifikasi karena memiliki struktur berbentuk pohon yang sistematis dan logis, sehingga memudahkan pengguna dalam menginterpretasikan hasilnya [26]. Pemilihan algoritma *Decision Tree* dalam penelitian ini didasarkan pada keunggulannya dalam menangani data yang kompleks dengan cara mengurai informasi menjadi serangkaian keputusan yang lebih sederhana dan mudah dipahami. Model *Decision Tree* bekerja dengan membangun struktur pohon keputusan, di mana setiap *node* mewakili suatu kondisi berdasarkan fitur dalam dataset, setiap *branch* menunjukkan hasil dari kondisi yang diuji, dan setiap *leaf node* menggambarkan hasil akhir dari proses klasifikasi [27]. Pendekatan ini memungkinkan *Decision Tree* untuk secara efektif membagi data ke dalam kategori yang berbeda berdasarkan aturan yang dapat dijelaskan dengan baik, sehingga model ini sangat berguna dalam memahami pola dalam data serta membuat keputusan berbasis logika yang dapat diinterpretasikan dengan jelas oleh pengguna.

$$Entropy(S) = -\sum_{i=1}^{n} Pi \log_2 Pi \quad (1)$$

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{Si}{s} * Entropy(Si) \quad (2)$$

Tahap pertama dalam algoritma Decision Tree dimulai dengan perhitungan entropy, yang digunakan untuk mengukur tingkat ketidakpastian dalam dataset. Entropy didefinisikan dalam persamaan (1), di mana S merupakan kumpulan data pada simpul yang sedang dianalisis, n menunjukkan jumlah kelas dalam dataset, dan Pi merepresentasikan probabilitas kemunculan kelas ke-i. Setelah entropy dihitung, langkah selanjutnya adalah menentukan information gain, yang bertujuan untuk mengukur seberapa besar pengurangan ketidakpastian dalam dataset setelah dilakukan pemisahan berdasarkan atribut tertentu. Perhitungan information gain mengikuti persamaan (2), di mana $Entropy(S)$ menggambarkan entropy awal dataset sebelum pemisahan terjadi, n merujuk pada jumlah subset yang terbentuk akibat pemisahan berdasarkan atribut A, $|Si|$ menunjukkan jumlah sampel dalam subset Si, $|S|$ adalah total jumlah sampel dalam dataset, sedangkan $Entropy(Si)$ merepresentasikan entropy dari subset yang telah terbentu.

2.3.2 *Resampling*

Resampling merupakan suatu metode yang diterapkan untuk mengatasi permasalahan ketidakseimbangan kelas dalam dataset dengan menerapkan dua pendekatan utama, yaitu *oversampling* dan *undersampling* [28]. *Oversampling* dilakukan dengan menambahkan jumlah sampel pada kelas minoritas sehingga distribusi kelas menjadi lebih seimbang, yang pada akhirnya dapat meningkatkan kemampuan model dalam mengenali pola dari kelas yang kurang terwakili dalam data. Sebaliknya, *undersampling* diterapkan dengan cara mengurangi jumlah sampel dari kelas mayoritas, sehingga distribusi data menjadi lebih proporsional dan model tidak cenderung berpihak pada kelas yang dominan. Dengan menerapkan metode *resampling* ini, diharapkan model yang digunakan dalam penelitian dapat menghasilkan prediksi yang lebih akurat dan adil terhadap semua kelas dalam dataset [29]. Karena dataset yang digunakan memiliki distribusi kelas yang tidak seimbang, diperlukan penyesuaian dengan menerapkan teknik

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

Synthetic Minority Over-sampling Technique (SMOTE). Dengan cara ini, jumlah sampel pada kelas minoritas bertambah, memungkinkan model untuk belajar lebih baik tanpa bias terhadap kelas mayoritas. Salah satu keunggulan utama dari SMOTE adalah kemampuannya dalam mencegah *overfitting*, yang sering terjadi pada metode oversampling konvensional. Hal ini dikarenakan data yang dihasilkan bukan sekadar salinan, melainkan kombinasi dari beberapa sampel terdekat dalam ruang fitur. Dengan demikian, SMOTE membantu model dalam membentuk representasi yang lebih umum terhadap kelas minoritas, sehingga meningkatkan akurasi dan performa keseluruhan dalam proses klasifikasi [30].

2.3.3 *Particle Swarm Optimization*

Dalam penelitian ini, *Particle Swarm Optimization* (PSO) digunakan untuk mengoptimalkan hyperparameter Decision Tree guna meningkatkan akurasi model klasifikasi diabetes. PSO merupakan algoritma berbasis populasi yang meniru perilaku kolektif kawanan burung atau ikan dalam mencari solusi optimal. Algoritma ini telah terbukti efektif dalam optimasi hyperparameter pada berbagai model pembelajaran mesin. PSO bekerja melalui tiga tahap utama. Pertama, setiap partikel diinisialisasi dengan posisi dan kecepatan awal dalam ruang pencarian, di mana setiap partikel merepresentasikan kombinasi *hyperparameter max_depth* (3–15), *min_samples_split* (2–10), dan *min_samples_leaf* (1–4). Kedua, kecepatan partikel diperbarui berdasarkan *personal best* (nilai terbaik yang pernah ditemukan oleh partikel itu sendiri) dan *global best* (nilai terbaik dari seluruh populasi partikel). Ketiga, posisi partikel diperbarui secara iteratif hingga mencapai solusi optimal[31]. Fungsi evaluasi yang digunakan adalah akurasi model pada data validasi, sehingga setiap kombinasi *hyperparameter* diuji dengan melatih *Decision Tree* dan mengukur akurasinya. Proses ini terus berlangsung hingga jumlah iterasi maksimum tercapai atau hingga konvergensi ditemukan. Dengan metode ini, PSO memungkinkan pemilihan *hyperparameter* secara otomatis, meningkatkan efisiensi pencarian, dan menghasilkan model yang lebih akurat.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{3}$$

$$v_i^{t+1} = w * v_i^t + c_1 * rand * \left(P_{best_i} - x_i^t\right) + c_2 * rand * (g_{best} - x_i^t) \tag{4}$$

Seluruh proses perhitungan dilakukan secara berurutan dengan menggunakan Persamaan (3) dan (4) [32]. Pada perhitungan ini, x_i^{t+1} merepresentasikan posisi terbaru dari suatu partikel, sedangkan v_i^{t+1} menggambarkan kecepatan partikel pada iterasi berikutnya. Faktor $w * v_i^t$ mencerminkan kemampuan eksplorasi dari algoritma *Particle Swarm Optimization* (PSO), di mana www berfungsi sebagai faktor pembobot, dan *rand* merupakan bilangan acak yang bernilai antara 0 hingga 1. Selain itu, P_{best_i} mengacu pada solusi terbaik yang telah ditemukan oleh agen ke-i selama proses iterasi, sedangkan g_{best} merepresentasikan solusi terbaik secara keseluruhan dalam populasi.

2.4 Evaluasi Hasil

Confusion matrix merupakan salah satu alat evaluasi yang digunakan untuk mengukur tingkat kinerja dari suatu algoritma klasifikasi dengan memberikan pemahaman yang lebih mendalam mengenai distribusi prediksi yang benar dan salah. Alat ini menyajikan informasi dalam bentuk matriks yang terdiri dari empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Melalui *confusion matrix*, performa model klasifikasi dapat dianalisis dengan lebih rinci, terutama dalam menilai seberapa baik model dalam mengidentifikasi kelas yang benar serta kesalahan yang terjadi dalam prediksi. Dengan kata lain, *confusion matrix* tidak hanya memberikan gambaran mengenai akurasi secara keseluruhan, tetapi juga membantu dalam mengevaluasi keseimbangan antara presisi dan sensitivitas model. Oleh karena itu, alat evaluasi ini sangat berguna dalam memahami kekuatan dan kelemahan model klasifikasi sebelum diterapkan dalam pengambilan keputusan yang lebih luas [33].

Tabel 1. *Confusion Matrix*

	Predicted Negative Class	**Predictef Positive Class**
Actual Negative Class	True Negative	False Positive
Actual Positive Class	False Negative	True Positive

Rincian hasil klasifikasi model yang ditampilkan dalam bentuk *confusion matrix* dapat dilihat pada Tabel 1. *Confusion matrix* merupakan alat yang sangat penting dalam evaluasi model pembelajaran mesin. Dengan menggunakan *confusion matrix*, berbagai metrik evaluasi utama dapat dihitung untuk menilai performa model secara lebih mendalam. Beberapa metrik evaluasi yang dapat diperoleh dari *confusion matrix* mencakup *accuracy*, *precision*, *recall*, dan *F1-score*, yang masing-masing memberikan perspektif berbeda dalam menilai kualitas model dalam mengklasifikasikan data dengan benar. Salah satu metrik yang paling umum digunakan adalah *accuracy*, yang memberikan gambaran mengenai proporsi keseluruhan prediksi yang benar dibandingkan dengan jumlah total sampel yang diuji. Metrik ini sangat berguna dalam memberikan indikasi awal mengenai seberapa baik model dalam mengklasifikasikan data, tetapi dalam kasus data yang tidak seimbang, *accuracy* saja mungkin tidak cukup untuk menilai performa model secara menyeluruh. Oleh karena itu, metrik lain seperti *precision*, *recall*, dan *F1-score* juga dihitung untuk mendapatkan pemahaman yang lebih komprehensif. Persamaan (5) menunjukkan rumus matematis

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

untuk menghitung *accuracy*, yang secara formal didefinisikan sebagai rasio jumlah sampel yang diklasifikasikan dengan benar terhadap total sampel yang tersedia. Dengan adanya metrik ini, dapat dilakukan perbandingan antara model sebelum dan sesudah proses optimasi, sehingga efektivitas dari langkah-langkah *preprocessing* serta tuning *hyperparameter* dapat dievaluasi dengan lebih objektif.

$$Accuracy = \frac{TP+TN}{TP=TN=FP=FN} \quad (5)$$

Selanjutnya, perhitungan F1-score dilakukan sebagai salah satu metrik evaluasi utama dalam menilai performa model klasifikasi. F1-score merupakan ukuran yang menggabungkan *precision* dan *recall* dalam satu nilai komposit, sehingga memberikan gambaran yang lebih seimbang terkait dengan kemampuan model dalam mengidentifikasi kelas positif secara akurat. Nilai *F1-score* dihitung sebagai rata-rata harmonis antara *precision* dan *recall*, yang bertujuan untuk memberikan kompromi antara kedua metrik tersebut, terutama dalam situasi di mana terdapat ketidakseimbangan kelas dalam dataset. Formula matematis yang digunakan untuk menghitung *F1-score* dapat ditemukan pada Persamaan (6). *Precision* dan *recall*, yang merupakan dua komponen utama dalam perhitungan *F1-score*, masing-masing memiliki definisi dan peran yang berbeda dalam evaluasi model klasifikasi. *Precision*, sebagaimana dinyatakan dalam Persamaan (7), mengukur seberapa banyak prediksi positif yang benar dibandingkan dengan total prediksi positif yang dihasilkan oleh model. Dengan kata lain, *precision* menggambarkan tingkat akurasi model dalam mengklasifikasikan suatu sampel sebagai positif tanpa menghasilkan terlalu banyak false positives. Sementara itu, *recall*, yang didefinisikan dalam Persamaan (8), menunjukkan kemampuan model dalam menangkap semua kasus positif yang sebenarnya ada dalam dataset. Semakin tinggi nilai *recall*, semakin baik model dalam mengenali dan mendeteksi seluruh kasus positif tanpa melewatkan terlalu banyak *instance* yang seharusnya diklasifikasikan sebagai positif. Karena terdapat keseimbangan yang harus dicapai antara *precision* dan *recall*, *F1-score* menjadi metrik yang sangat berguna karena mampu memberikan gambaran menyeluruh terhadap performa model. Dalam penelitian ini, perhitungan *F1-score*, *precision*, dan *recall* dilakukan untuk mengevaluasi efektivitas model sebelum dan sesudah dilakukan optimasi, guna memastikan bahwa peningkatan performa model tidak hanya bergantung pada satu metrik saja, tetapi juga mempertimbangkan aspek lain dalam klasifikasi yang lebih luas.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

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

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

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Dalam penelitian ini, menggunakan *Diabetes Prediction Dataset*, yang diperoleh dari platform *Kaggle*, sebagai sumber data utama dalam membangun model prediksi diabetes. Dataset ini berisi berbagai informasi medis dan demografis pasien yang berkaitan dengan faktor risiko penyakit diabetes. Data yang tersedia mencakup beragam variabel yang dapat digunakan untuk memahami pola-pola yang berkontribusi terhadap kemungkinan seseorang terdiagnosis diabetes. Secara keseluruhan, dataset ini terdiri dari 100.000 entri, di mana setiap entri mewakili satu individu pasien. Dataset ini memiliki satu variabel target dan delapan variabel fitur yang digunakan sebagai input dalam proses analisis dan prediksi diabetes. Variabel target, yaitu Diabetes, menunjukkan status kesehatan pasien terkait diabetes, yang dikategorikan dalam dua kelas: positif (1) jika pasien terdiagnosis diabetes dan negatif (0) jika tidak. Dengan menggunakan dataset ini, model yang dikembangkan dalam penelitian dapat dilatih, diuji, dan divalidasi untuk meningkatkan kemampuan dalam memprediksi risiko diabetes berdasarkan kombinasi fitur-fitur yang tersedia. Keberadaan berbagai faktor medis dan demografis dalam dataset ini memungkinkan analisis lebih mendalam mengenai hubungan antara faktor risiko dan kemungkinan seseorang terkena diabetes, sehingga dapat memberikan wawasan yang lebih baik dalam pengembangan sistem prediksi kesehatan berbasis data. Tabel 2 menyajikan informasi mengenai variabel yang terdapat dalam dataset. Tabel 2 menampilkan variabel yang digunakan dalam penelitian ini, mencakup data demografis, riwayat kesehatan, dan indikator medis. *Gender* dan *Age* menunjukkan jenis kelamin serta usia pasien, sementara *Hypertension*, *Heart Disease*, dan *Smoking History* menggambarkan kondisi kesehatan serta kebiasaan merokok. *BMI*, *HbA1c Level*, dan *Blood Glucose Level* berperan sebagai indikator medis utama.

Tabel 2. Informasi Dataset

No	Nama Variabel	Deskripsi
1	Gender	Jenis kelamin pasien
2	Age	Usia pasien dalam satuan tahun
3	Hypertension	Status hipertensi pasien
4	Heart Disease	Indikator riwayat penyakit jantung
5	Smoking History	Riwayat merokok pasien

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

6	BMI	Indeks massa tubuh pasien, menunjukkan tingkat kelebihan berat badan
7	HbA1c Level	Kadar hemoglobin terglikasi, indikator kadar gula darah jangka panjang
8	Blood Glucose Level	Tingkat kadar gula dalam darah pasien.
9	Diabetes	Variabel target: 1 jika pasien terdiagnosis diabetes, 0 jika tidak.

3.2 Data Preprocessing

Penelitian diawali dengan tahap pemrosesan data (*data preprocessing*), yang mencakup serangkaian langkah penting untuk memastikan kualitas data yang akan digunakan dalam pelatihan model. Langkah pertama yang dilakukan adalah *data cleaning*, yaitu proses pemeriksaan terhadap kemungkinan adanya nilai yang hilang (*missing values*) dalam dataset. Hasil pengecekan menunjukkan bahwa tidak terdapat *missing values* dalam dataset ini, sehingga tidak diperlukan imputasi atau penanganan khusus terhadap nilai yang hilang. Namun, analisis lebih lanjut terhadap dataset mengungkapkan adanya data duplikat sebanyak 3.854 entri. Keberadaan data duplikat ini dapat memengaruhi kualitas model dengan menyebabkan bias dalam proses pelatihan, sehingga perlu dilakukan tindakan penghapusan untuk memastikan keakuratan dan keandalan data yang digunakan. Jumlah data duplikat dalam dataset ini ditampilkan pada Gambar 3 menyajikan hasil identifikasi data duplikat yang terdapat dalam dataset. Keberadaan data duplikat yang ditampilkan pada Gambar 3 digunakan sebagai acuan dalam tahap pemrosesan data lebih lanjut, sehingga memastikan kualitas data yang lebih baik sebelum digunakan dalam analisis atau pemodelan.

index	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
2756	1	80.0	0	0	0	27.32	6.6	159	0
3272	0	80.0	0	0	0	27.32	3.5	80	0
3418	0	19.0	0	0	0	27.32	6.5	100	0
3939	0	78.0	1	0	3	27.32	3.5	130	0
3960	1	47.0	0	0	0	27.32	6.0	200	0
4106	0	51.0	0	0	0	27.32	6.6	200	0
4128	0	80.0	0	0	4	27.32	5.7	85	0
4224	1	80.0	1	0	3	27.32	6.2	130	0
4263	1	80.0	0	0	0	27.32	6.0	100	0
4486	1	50.0	0	0	0	27.32	4.8	155	0

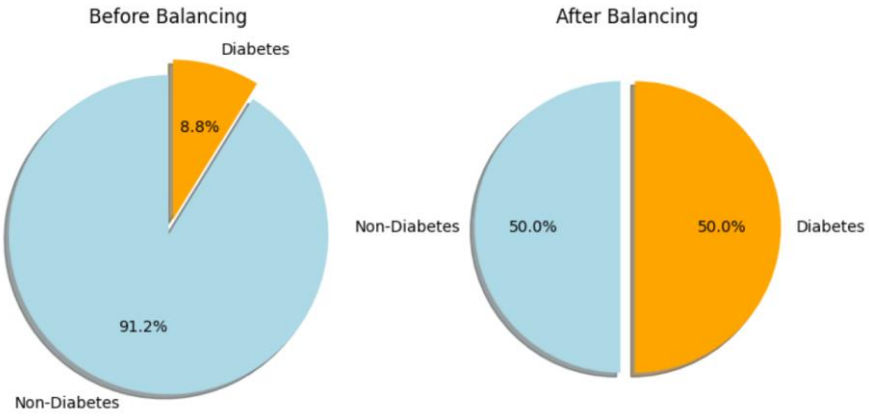
Gambar 3. Data Duplikat Pada Dataset

Pada tahapan berikutnya dalam penelitian ini, dilakukan proses pembagian dataset menjadi dua subset utama, yaitu data latih (*training data*) dan data uji (*testing data*). Langkah ini merupakan bagian yang sangat krusial dalam proses pengembangan model pembelajaran mesin, karena bertujuan untuk memastikan bahwa model yang dihasilkan memiliki kemampuan prediksi yang optimal serta tingkat akurasi yang tinggi. Dengan pembagian dataset yang terstruktur, model dapat belajar dari sejumlah besar data selama tahap pelatihan, sambil tetap menyediakan data terpisah untuk menguji kinerja model setelah proses pelatihan selesai. Pembagian dataset dilakukan dengan rasio 80:20, di mana 80% dari total dataset digunakan sebagai data latih. Data latih ini digunakan dalam tahap pembelajaran mesin untuk mengidentifikasi pola dan hubungan antara variabel-variabel dalam dataset, sehingga model dapat memahami karakteristik data dengan lebih baik. Selama proses pelatihan, model akan mengekstraksi informasi penting dari berbagai fitur dalam dataset yang berkontribusi terhadap hasil prediksi. Dengan proses pembelajaran yang menyeluruh ini, diharapkan model dapat menghasilkan prediksi yang lebih akurat dan andal. Sementara itu, 20% dari dataset dialokasikan sebagai data uji (*testing data*). Data uji memiliki peran yang sangat penting sebagai alat evaluasi yang digunakan setelah model selesai dilatih. Subset ini terdiri dari data yang tidak pernah digunakan dalam proses pelatihan, sehingga dapat memberikan gambaran yang lebih objektif mengenai kemampuan model dalam menggeneralisasi pola-pola yang telah dipelajari. Evaluasi menggunakan data uji memungkinkan penelitian ini untuk mengukur seberapa baik model dapat beradaptasi terhadap data baru, yang merupakan indikator utama dari kinerja model di dunia nyata. Selain itu, pembagian dataset yang tepat juga membantu dalam mendeteksi potensi *overfitting*, yaitu kondisi di mana model terlalu menyesuaikan diri dengan data latih hingga kehilangan kemampuannya dalam memprediksi data baru. *Overfitting* dapat menyebabkan model memiliki performa yang sangat baik pada data latih tetapi gagal dalam menggeneralisasi pola yang ada pada data uji. Oleh karena itu, dengan strategi pembagian dataset yang proporsional dan sistematis, penelitian ini bertujuan untuk mengembangkan model yang seimbang dalam hal akurasi antara data latih dan data uji, sehingga model tidak hanya bekerja dengan baik selama proses pelatihan, tetapi juga tetap stabil dan akurat ketika diterapkan pada data yang belum pernah dilihat sebelumnya. Dengan pendekatan ini, model yang dihasilkan diharapkan memiliki kemampuan prediksi yang lebih andal, menghindari risiko *overfitting*, serta dapat memberikan hasil klasifikasi yang lebih efektif dan konsisten.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

3.3 Data Processing

Pada tahap pengujian ini, terdapat ketidakseimbangan distribusi kelas dalam dataset, dilakukan proses penyeimbangan data menggunakan teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Teknik ini diterapkan khusus pada data latih agar distribusi kelas tetap seimbang, tanpa mengubah proporsi kelas pada data uji guna menjaga validitas evaluasi model. Proses penyeimbangan data dengan SMOTE diawali dengan pemilihan sampel dari kelas minoritas secara acak dengan penggantian, di mana sampel yang sama dapat dipilih lebih dari sekali. Selanjutnya, sampel hasil duplikasi dari kelas minoritas ditambahkan kembali ke dalam dataset tanpa mengubah informasi asli. Dengan demikian, distribusi data menjadi lebih seimbang, sehingga bias terhadap kelas mayoritas dapat diminimalkan dan model dapat belajar secara lebih optimal dalam mengklasifikasikan kasus diabetes. Gambar 4 menyajikan visualisasi distribusi kelas dalam dataset sebelum dan sesudah dilakukan proses penyeimbangan data menggunakan teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Pada grafik pertama (sebelah kiri), terlihat bahwa distribusi kelas awal tidak seimbang, di mana jumlah data pada kelas 0 jauh lebih besar, yaitu sebanyak 70.130 data, dibandingkan dengan kelas 1 yang hanya memiliki 6.786 data. Setelah diterapkan teknik SMOTE, sebagaimana ditampilkan pada grafik kedua (sebelah kanan), distribusi kelas menjadi lebih seimbang. Jumlah data pada kelas 1 meningkat hingga menyamai kelas 0, yaitu sebesar 70.130 data. Proses penyeimbangan ini bertujuan untuk mengatasi bias yang dapat muncul akibat dominasi kelas mayoritas, sehingga model pembelajaran mesin dapat belajar secara lebih efektif dan menghasilkan prediksi yang lebih akurat.



Gambar 4. Hasil *Balancing* Label Diabetes dan Non-Diabetes

Selanjutnya, dilakukan perbandingan kinerja antara dua model yang dikembangkan untuk klasifikasi diabetes, yaitu *initial model* dan *final model*. *Initial model* merupakan model yang dilatih langsung menggunakan data mentah tanpa melalui proses preprocessing atau optimasi *hyperparameter*. Sementara itu, *final model* adalah model yang telah mengalami berbagai tahapan preprocessing dan optimasi guna meningkatkan kualitas data sebelum digunakan dalam pelatihan. Salah satu teknik *preprocessing* yang diterapkan adalah *Synthetic Minority Over-sampling Technique* (SMOTE), yang digunakan untuk menangani ketidakseimbangan kelas dalam dataset. Selain itu, dilakukan optimasi *hyperparameter* menggunakan *Particle Swarm Optimization* (PSO) untuk menemukan kombinasi *hyperparameter* terbaik yang dapat meningkatkan akurasi dan kestabilan model dalam mengklasifikasikan data. Hasil pengujian menunjukkan bahwa penggunaan SMOTE dalam penelitian ini tidak menyebabkan *overfitting*, karena model tetap menunjukkan peningkatan performa yang stabil tanpa kehilangan kemampuan generalisasi terhadap data uji. Meskipun *recall* mengalami sedikit penurunan, hal ini lebih disebabkan oleh selektivitas model setelah optimasi *hyperparameter* menggunakan PSO, bukan karena efek negatif dari SMOTE. Bahkan, penerapan SMOTE membantu meningkatkan keseimbangan data, yang berkontribusi pada peningkatan *precision* dan *F1-score*. Untuk memastikan bahwa SMOTE tidak menyebabkan model terlalu bergantung pada pola sintetis, dilakukan validasi silang guna memastikan model tetap mampu mengenali pola dari data asli dengan baik. Penelitian ini berfokus pada analisis perbedaan performa antara model yang tidak mengalami *preprocessing* dan model yang telah diproses sebelumnya, serta mengevaluasi dampak optimasi *hyperparameter* terhadap peningkatan akurasi. Tabel 3 menampilkan hasil pengujian yang menggambarkan kinerja model sebelum dan sesudah menjalani proses preprocessing serta optimasi.

Tabel 3. Hasil Pengujian Model

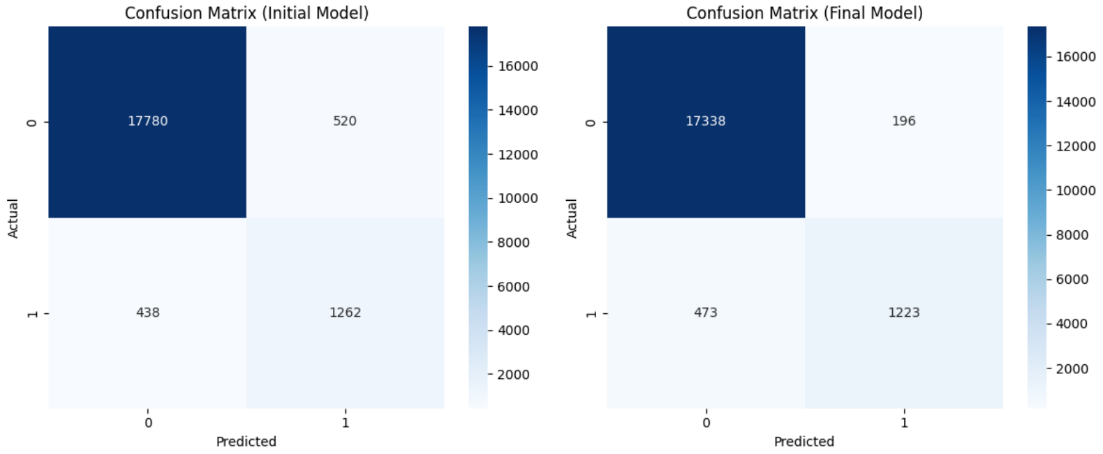
Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Initial Model	95.21%	70.82%	74.24%	72.49%
Final Model	96.52%	86.19%	72.11%	78.52%

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

Berdasarkan ringkasan pengujian dalam Tabel 3, dapat diperoleh perbandingan akurasi serta berbagai metrik evaluasi lainnya untuk kedua model. Pada tahap pengujian awal, model tanpa *preprocessing* menunjukkan performa yang lebih rendah dibandingkan model yang telah melalui *preprocessing* dan optimasi *hyperparameter*. Evaluasi menunjukkan bahwa model berbasis data mentah hanya mencapai *accuracy* sebesar 95.21%, *precision* sebesar 70.82%, *recall* sebesar 74.24%, dan *F1-score* sebesar 72.49%. Meskipun akurasinya cukup tinggi, rendahnya *precision* dan *recall* menunjukkan bahwa model masih belum mampu mengenali pola dalam data secara optimal. Sebaliknya, final model yang telah melalui *preprocessing* dan optimasi *hyperparameter* menunjukkan peningkatan signifikan dalam berbagai metrik evaluasi. Peningkatan ini membuktikan bahwa kombinasi *preprocessing* data dan optimasi *hyperparameter* berperan penting dalam meningkatkan performa model dalam klasifikasi diabetes. Selain itu, penerapan SMOTE terbukti efektif dalam meningkatkan keseimbangan data tanpa menyebabkan *overfitting*, sehingga model tetap dapat beradaptasi dengan baik terhadap data uji.

3.4 Evaluasi

Hasil pengujian model menunjukkan bahwa *preprocessing* data dan optimasi *hyperparameter* berkontribusi terhadap peningkatan kinerja model *Decision Tree* dalam memprediksi diabetes. Hal ini tercermin dalam perubahan *confusion matrix* pada Gambar 5. Gambar 5 merupakan hasil dari confusion matrix yang menggambarkan performa model sebelum dan sesudah dilakukan preprocessing serta optimasi. Selain itu, analisis terhadap confusion matrix pada Gambar 5 mengindikasikan adanya peningkatan yang signifikan dalam akurasi model dalam membedakan kelas positif dan negatif setelah melewati proses preprocessing dan optimasi. Hasil *confusion matrix* terutama pada penurunan jumlah *False Positives* (FP) dari 520 menjadi 196, yang mengindikasikan bahwa model menjadi lebih selektif dalam mengidentifikasi kasus positif. Pada model awal yang menggunakan data mentah tanpa optimasi, *confusion matrix* mencatat *True Negatives* (TN) sebesar 17.780, *True Positives* (TP) sebesar 1.262, *False Negatives* (FN) sebesar 438, dan *False Positives* (FP) sebesar 520. Angka ini menunjukkan bahwa masih terdapat kesalahan dalam mengklasifikasikan kasus positif. Setelah dilakukan *preprocessing* dan optimasi, terjadi perubahan dengan TN sebesar 17.338, TP sebesar 1.223, FN meningkat menjadi 473, dan FP berkurang menjadi 196. Meskipun akurasi model secara keseluruhan mengalami perbaikan, peningkatan jumlah *False Negatives* (FN) menunjukkan adanya konsekuensi, di mana lebih banyak kasus positif yang salah diklasifikasikan sebagai negatif. Oleh karena itu, meskipun *preprocessing* dan penyesuaian *hyperparameter* berhasil meningkatkan performa model, evaluasi lebih lanjut menggunakan metrik seperti *precision*, *recall*, atau *F1-score* tetap diperlukan untuk memberikan pemahaman yang lebih komprehensif mengenai keseimbangan antara peningkatan dan konsekuensi yang ditimbulkan..

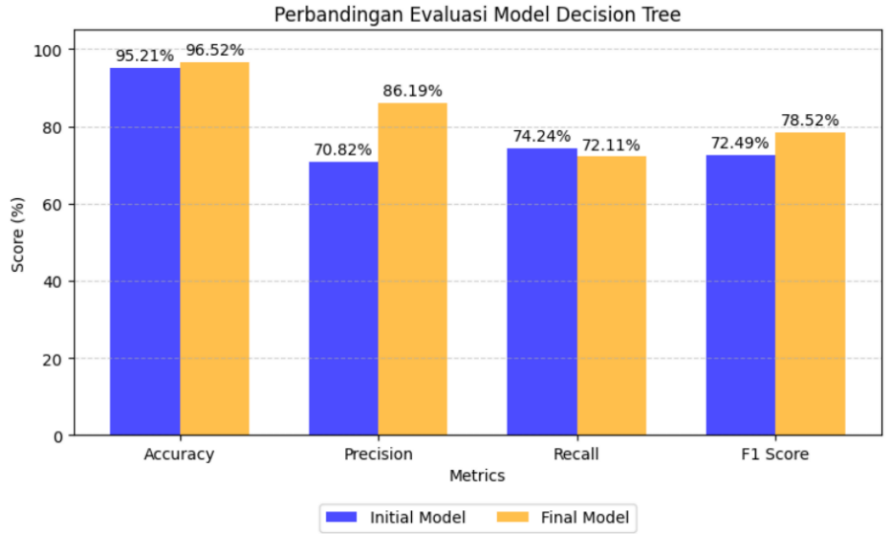


Gambar 5. Hasil *Confusion Matrix*

Evaluasi *confusion matrix* dilakukan untuk memastikan bahwa prediksi terhadap kelas positif memiliki tingkat akurasi yang optimal, sehingga model dapat digunakan secara andal dalam berbagai skenario. Analisis ini tidak hanya berfokus pada tingkat akurasi keseluruhan, tetapi juga mencakup metrik evaluasi yang lebih spesifik, seperti *precision*, *recall*, dan *F1-score*, yang masing-masing memberikan wawasan lebih mendalam tentang kinerja model dalam mengklasifikasikan data. Precision mengukur sejauh mana prediksi positif benar-benar sesuai dengan nilai sebenarnya, yang berarti semakin tinggi *precision*, semakin sedikit kesalahan dalam mengklasifikasikan sampel negatif sebagai positif. Sementara itu, *recall* menunjukkan sejauh mana model mampu menangkap seluruh kasus positif yang ada dalam dataset, sehingga metrik ini sangat penting dalam konteks di mana mengabaikan kasus positif dapat berdampak signifikan. Di sisi lain, *F1-score* merupakan rata-rata harmonis antara *precision* dan *recall*, yang memberikan gambaran menyeluruh tentang keseimbangan antara kedua metrik tersebut dan membantu mengidentifikasi apakah model lebih cenderung mengutamakan salah satu di antaranya. Gambar 6 merupakan hasil dari *evaluation matrix* yang menampilkan performa model awal dan model yang telah dilakukan *preprocessing* serta optimasi. Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 6, hasil pengujian pada model awal yang menggunakan data mentah tanpa melalui tahapan *preprocessing* dan optimasi menunjukkan bahwa nilai *accuracy*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

sebesar 95.21%. Setelah dilakukan berbagai perbaikan melalui proses *preprocessing* data serta penyempurnaan *hyperparameter* model, nilai *accuracy* meningkat menjadi 96.52%, yang menunjukkan bahwa model secara keseluruhan lebih baik dalam mengklasifikasikan data dengan benar. Namun, peningkatan *accuracy* ini tidak berarti model lebih akurat dalam mengenali semua kasus positif. Analisis lebih lanjut menunjukkan bahwa *precision* meningkat secara signifikan, tetapi *recall* mengalami sedikit penurunan. Hal ini mengindikasikan bahwa model lebih selektif dalam menentukan prediksi positif, tetapi juga lebih sering melewatkan beberapa kasus positif yang sebenarnya ada. Dengan kata lain, penerapan *preprocessing* dan pengoptimalan model memang membantu meningkatkan ketepatan model dalam mengklasifikasikan sampel yang dipilihnya sebagai positif, tetapi juga menyebabkan model lebih berhati-hati sehingga berisiko melewatkan beberapa kasus positif. Meskipun demikian, secara keseluruhan, penerapan *preprocessing* dan optimasi *hyperparameter* berhasil mengurangi kesalahan prediksi dan meningkatkan keseimbangan antara *precision* dan *recall*, yang menghasilkan sistem klasifikasi yang lebih stabil dan efisien.



Gambar 6. Hasil *Evaluation Matrix*

Recall yang ditampilkan pada Gambar 6 dievaluasi untuk mengukur sejauh mana model mampu mendeteksi kasus positif dengan akurat. *Recall* yang tinggi mencerminkan sensitivitas model yang baik dalam mengidentifikasi kasus positif, sehingga jumlah false negatives (kasus positif yang tidak terdeteksi) dapat diminimalkan. Berdasarkan hasil pengujian, model awal yang menggunakan data mentah tanpa *preprocessing* dan optimasi memiliki *recall* sebesar 74.24%. Namun, setelah dilakukan *preprocessing* dan penyempurnaan *hyperparameter*, *recall* mengalami sedikit penurunan menjadi 72.11%. Penurunan *recall* ini menunjukkan bahwa model yang telah dioptimalkan lebih berhati-hati dalam memprediksi kelas positif, tetapi sebagai konsekuensinya, lebih banyak kasus positif yang tidak terdeteksi. Dalam konteks deteksi diabetes, hal ini dapat berdampak signifikan karena model lebih cenderung mengklasifikasikan pasien positif sebagai negatif, yang berisiko menyebabkan keterlambatan dalam diagnosis dan penanganan lebih lanjut. Untuk mengevaluasi keseimbangan antara *precision* dan *recall*, dilakukan pengukuran menggunakan *F1-score*. Metrik ini memberikan gambaran yang lebih komprehensif mengenai performa model dengan mempertimbangkan hubungan antara ketepatan dalam memprediksi kelas positif dan kemampuannya dalam menangkap seluruh kasus positif. Berdasarkan Gambar 6, penerapan *preprocessing* dan optimasi berhasil meningkatkan *F1-score* menjadi 78.52%. Hal ini menunjukkan bahwa meskipun *recall* mengalami sedikit penurunan, model yang telah dioptimalkan tetap lebih akurat dalam mengklasifikasikan kasus positif yang benar, sehingga secara keseluruhan lebih efektif dalam melakukan klasifikasi dengan keseimbangan yang lebih baik antara *precision* dan *recall*.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa *preprocessing* data dan optimasi *hyperparameter* secara signifikan meningkatkan kinerja model *Decision Tree* dalam memprediksi risiko diabetes. Ketidakseimbangan kelas ditangani menggunakan teknik SMOTE, sementara optimasi *hyperparameter* dengan *Particle Swarm Optimization* (PSO) berhasil meningkatkan akurasi model dari 95.21% menjadi 96.52%. Selain akurasi, peningkatan juga terlihat pada metrik evaluasi seperti *precision*, *recall*, dan *F1-score*. *Precision* meningkat dari 70.82% menjadi 86.19%, menunjukkan bahwa model lebih akurat dalam mengidentifikasi kasus positif, dengan jumlah *False Positives* (FP) yang semakin berkurang. Namun, recall mengalami sedikit penurunan dari 74.24% menjadi 72.11%, dengan *False Negatives* (FN) meningkat dari 438 menjadi 473. Hal ini menunjukkan bahwa meskipun model lebih akurat dalam memprediksi kasus positif, ia menjadi lebih selektif sehingga lebih banyak kasus positif yang terlewat. Penurunan *recall* ini kemungkinan

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

besar terjadi akibat kombinasi optimasi PSO dan SMOTE. PSO dapat menyebabkan model lebih menekankan *precision*, sementara SMOTE mungkin mengubah distribusi data sehingga model lebih berhati-hati dalam mengklasifikasikan kasus positif. Dalam konteks deteksi diabetes, penurunan *recall* ini dapat berdampak signifikan karena model lebih cenderung mengklasifikasikan pasien positif sebagai negatif, yang dapat menyebabkan keterlambatan diagnosis dan penanganan. *F1-score* meningkat dari 72.49% menjadi 78.52%, menunjukkan adanya keseimbangan antara *precision* dan *recall*, tetapi masih terdapat indikasi bahwa model cenderung terlalu spesifik terhadap pola dalam data *training*. Selain itu, SMOTE tidak selalu mencegah *overfitting* terutama jika model terlalu kompleks atau jika optimasi *hyperparameter* dilakukan tanpa validasi yang cukup. Oleh karena itu, penelitian selanjutnya disarankan untuk membandingkan metode optimasi *hyperparameter* lain, seperti *Grid Search* atau *Bayesian Optimization*, serta mengeksplorasi model alternatif seperti *Random Forest*, *XGBoost*, atau *Neural Networks* guna meningkatkan akurasi dan generalisasi model dalam mendeteksi risiko diabetes.

REFERENCES

[1] U. M. Butt, S. Letchmunan, M. Ali, F. H. Hassan, A. Baqir, and H. H. R. Sherazi, "Machine Learning Based Diabetes Classification and Prediction for Healthcare Applications," *J. Healthc. Eng.*, vol. 2021, no. October, 2021, doi: 10.1155/2021/9930985.

[2] L. D. Prasanti and D. Trias, "Perancangan Sisitem Pendukung Keputusan Rekomendasi Menu Makanan Pada Penderita Diabetes Mellitus Menggunakan Metode Simple Additive Weighting," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 3, no. 1, pp. 11–16, 2024, doi: 10.69916/jkbti.v3i1.62.

[3] N. M. Putry, "Komparasi Algoritma Knn Dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus," *EVOLUSI J. Sains dan Manaj.*, vol. 10, no. 1, 2022, doi: 10.31294/evolusi.v10i1.12514.

[4] International Diabetes Federation, "IDF Annual Report 2022," Brussels, Belgium, 2022. [Online]. Available: https://idf.org/media/uploads/2023/07/IDF_Annual_Report_2022_Final.pdf

[5] I. Jannoud, M. Z. Masoud, Y. Jaradat, A. Manaserah, and D. Zaidan, "A Multi-Layered Hybrid Machine Learning Algorithm (MLHA) for Type II Diabetes Classification," *Procedia Comput. Sci.*, vol. 237, pp. 445–452, 2024, doi: 10.1016/j.procs.2024.05.126.

[6] D. Kaviyaadharshani, M. Nivedhidha, R. Jeyarohini, J. Lece Elizabeth Rani, M. P. Ramkumar, and G. S. R. Emil Selvan, "Diagnosing Diabetes using Machine Learning-based Predictive Models," *Procedia Comput. Sci.*, vol. 233, no. 2023, pp. 288–294, 2024, doi: 10.1016/j.procs.2024.03.218.

[7] 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, 2021, doi: 10.1016/j.icte.2021.02.004.

[8] R. G. Wardhana, G. Wang, and F. Sibuea, "Penerapan Machine Learning Dalam Prediksi Tingkat Kasus Penyakit Di Indonesia," *J. Inf. Syst. Manag.*, vol. 5, no. 1, pp. 40–45, 2023, doi: 10.24076/joism.2023v5i1.1136.

[9] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *Bina Insa. Ict J.*, vol. 7, no. 2, p. 156, 2020, doi: 10.51211/biict.v7i2.1422.

[10] M. Mustafa, "Diabetes prediction dataset." Accessed: Feb. 28, 2025. [Online]. Available: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data

[11] A. Handika Permana, F. Rakhmat Umbara, and F. Kasyidi, "Klasifikasi Penyakit Jantung Tipe Kardiovaskular Menggunakan Adaptive Synthetic Sampling dan Algoritma Extreme Gradient Boosting," *Build. Informatics, Technol. Sci.*, vol. 6, no. 1, pp. 499–508, 2024, doi: 10.47065/bits.v6i1.5421.

[12] M. Sulistiyono, Y. Pristyanto, S. Adi, and G. Gumelar, "Implementasi Algoritma Synthetic Minority Over-Sampling Technique untuk Menangani Ketidakseimbangan Kelas pada Dataset Klasifikasi," *Sistemasi*, vol. 10, no. 2, p. 445, 2021, doi: 10.32520/stmsi.v10i2.1303.

[13] W. D. Septiani and M. Marlina, "Comparison of Decision Tree, Naïve Bayes, and Neural Network Algorithm for Early Detection of Diabetes," *J. Pilar Nusa Mandiri*, vol. 17, no. 1, pp. 73–78, 2021, doi: 10.33480/pilar.v17i1.2213.

[14] D. R. Damayanti and A. Purwinarko, "Application of C4.5 Algorithm Using Synthetic Minority Oversampling Technique (SMOTE) and Particle Swarm Optimization (PSO) for Diabetes Prediction," *Recursive J. Informatics*, vol. 2, no. 1, pp. 18–27, 2024, doi: 10.15294/rji.v2i1.64928.

[15] E. Muningsih, F. F. D. Imaniawan, A. Widayanto, E. A. Pratama, Sutrisno, and S. Kiswati, "Optimized decision tree classification method for diabetes prediction," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 22, no. 4, pp. 941–948, 2024, doi: 10.12928/TELKOMNIKA.v22i4.25656.

[16] O. Mehrpour *et al.*, "Comparison of decision tree with common machine learning models for prediction of biguanide and sulfonylurea poisoning in the United States: an analysis of the National Poison Data System," *BMC Med. Inform. Decis. Mak.*, vol. 23, no. 1, pp. 1–11, 2023, doi: 10.1186/s12911-022-02095-y.

[17] B. A. Candra Permana and I. K. Dewi Patwari, "Komparasi Metode Klasifikasi Data Mining Decision Tree dan Naïve Bayes Untuk Prediksi Penyakit Diabetes," *Infotek J. Inform. dan Teknol.*, vol. 4, no. 1, pp. 63–69, 2021, doi: 10.29408/jit.v4i1.2994.

[18] M. Solehuddin, W. A. Syafei, and R. Gernowo, "Metode Decision Tree untuk Meningkatkan Kualitas Rencana Pelaksanaan Pembelajaran dengan Algoritma C4.5," *J. Penelit. dan Pengemb. Pendidik.*, vol. 6, no. 3, pp. 510–519, 2022, doi: 10.23887/jppp.v6i3.52840.

[19] T. W. Pratiwi and T. Arifin, "Optimasi Decision Tree Menggunakan Particle Swarm Optimization untuk Klasifikasi Kesuburan pada Pria," *Sistemasi*, vol. 10, no. 1, p. 13, 2021, doi: 10.32520/stmsi.v10i1.967.

[20] M. Sukron, A. Supriadi, and R. Sulton, "Optimasi Metode Naïve Bayes Menggunkanan Algoritma Particle Swarm Optimization (Pso) Untuk Prediksi Penyakit Diabetes Mellitus," *COREAI J. Kecerdasan Buatan, Komputasi dan Teknol. Inf.*, vol. 2, no. 2, pp. 18–24, 2022, doi: 10.33650/coreai.v2i2.3304.

[21] T. S. Lestari, I. Ismaniah, and W. Priatna, "Particle Swarm Optimization for Optimizing Public Service Satisfaction Level Classification," *J. Nas. Pendidik. Tek. Inform.*, vol. 13, no. 1, pp. 147–155, 2024, doi: 10.23887/janapati.v13i1.69612.

[22] A. Massahiro *et al.*, "The evolution of CRISP-DM for Data Science : Methods , Processes and Frameworks," *SBC Rev.*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2659−2671
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7111

Comput. Sci., vol. 4, no. 1, pp. 28–43, 2024, doi: 10.5753/reviews.2024.3757.
[23] A. Anggrawan and M. Mayadi, "Application of KNN Machine Learning and Fuzzy C-Means to Diagnose Diabetes," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 2, pp. 405–418, 2023, doi: 10.30812/matrik.v22i2.2777.
[24] G. Y. Lee, L. Alzamil, B. Doskenov, and A. Termehchy, "A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance," *arXiv*, vol. 2021, no. September, pp. 1–6, 2021, doi: 10.48550/arXiv.2109.07127.
[25] B. F. Rochman, A. Rahim, and T. A. Y. Siswa, "Optimasi Algoritma KNN dengan Parameter K dan PSO Untuk Klasifikasi Status Gizi Balita," *J. Media Inform. Budidarma*, vol. 8, no. 3, p. 1609, 2024, doi: 10.30865/mib.v8i3.7841.
[26] E. Fauziningrum and E. I. Suryaningsih, *PENERAPAN DATA MINING METODE DECISION TREE UNTUK MENGUKUR PENGUASAAN BAHASA INGGRIS MARITIM*. Semarang: CV. Pustaka STIMART AMNI Semarang, 2021. [Online]. Available: https://penerbit.unimar-amni.ac.id/product/penerapan-data-mining-metode-decision-tree-untuk-mengukur-penguasaan-bahasa-inggris-maritim/
[27] S. Sza *et al.*, "Penerapan Decision Tree Dan Random Forest Dalam Deteksi the Application of Decision Tree and Random Forest in Detecting Human Stress Levels Based on Sleep Conditions," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 7, pp. 1503–1510, 2023, doi: 10.25126/jtiik.2024117993.
[28] A. T. Akbar, R. Husaini, B. M. Akbar, and S. Saifullah, "A proposed method for handling an imbalance data in classification of blood type based on Myers-Briggs type indicator," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 4, pp. 276–283, 2020, doi: 10.14710/jtsiskom.2020.13625.
[29] A. Salehi and M. Khedmati, "Hybrid clustering strategies for effective oversampling and undersampling in multiclass classification," *Sci. Rep.*, vol. 15, no. 1, p. 3460, 2025, doi: 10.1038/s41598-024-84786-2.
[30] D. Kurniawan *et al.*, "Region of Interest-Based Breast Cancer Detection with Oversampling Technique," *Ing. des Syst. d'Information*, vol. 30, no. 1, pp. 149–156, 2025, doi: 10.18280/isi.300112.
[31] J. B. E. Putry, A. T. Sasongko, and W. Hadikristanto, "Optimasi Decision Tree Menggunakan Particle Swarm Optimization (PSO) pada Risiko Kredit KMG Bank DKI," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 4, pp. 1403–1410, 2024, doi: 10.57152/malcom.v4i4.1521.
[32] H. A. Aliyu, I. O. Muritala, H. Bello-Salau, S. Mohammed, A. J. Onumanyi, and O.-O. Ajayi, "Optimizing machine learning algorithms for diabetes data: A metaheuristic approach to balancing and tuning classifiers parameters," *Franklin Open*, vol. 8, no. December 2023, p. 100153, 2024, doi: 10.1016/j.fraope.2024.100153.
[33] N. C. Sari and T. L. Larasati, "Komparasi Algoritma Naïve Bayes dan Gradient Boosting untuk Prediksi Pasien Diabetes," *J. Nas. Teknol. dan Sist. Inf.*, vol. 10, no. 2, pp. 118–125, 2024, doi: 10.25077/TEKNOSI.v10i2.2024.118-125.