Vol. 01, No. 01, November 2025
Hal. 1 – 9

Sistem Deteksi Dini Penyakit Diabetes Menggunakan Algoritma Random Forest

Viona Leny Anjani[1], Shanny Novalina Turnip[2], Ulfah Nur Uzlifah[3], Rajnaparamitha Kusumastuti[4]
[1,2,3]Prodi Informatika, STMIK Amikom Surakarta
[4]Prodi teknologi Informasi, STMIK Amikom Surakarta
[1,2,3,4]Sukoharjo Indonesia
e-mail: [1]viona.10473@mhs.amikomsolo.ac.id, [2]shanny.10481@mhs.amikomsolo.ac.id, [3]ulfah.10488@mhs.amikomsolo.ac.id, [4]rajna@dosen.amikomsolo.ac.id

Abstrak

Dengan menggunakan algoritma "Random Forest" dan metode "Synthetic Minority Over-sampling Technique" (SMOTE), penelitian ini bertujuan untuk membangun sistem deteksi dini diabetes. Tujuan dari metode ini adalah untuk mengatasi masalah ketidakseimbangan kelas dalam data. Data yang digunakan adalah dari Pima Indian Diabetes, yang terdiri dari berbagai fitur klinis pasien, termasuk kadar glukosa, indeks massa tubuh (BMI), usia, dan lainnya. Pengembangan model melibatkan tahap pra-pemrosesan data, penerapan SMOTE, pelatihan model dengan Random Forest, dan pengaturan parameter dengan GridSearchCV untuk mendapatkan konfigurasi terbaik. Hasil evaluasi menunjukkan bahwa model berhasil mencapai akurasi sebesar 82,80%, dengan glukosa, BMI, dan usia menjadi fitur yang paling berpengaruh dalam prediksi.

Hasilnya menunjukkan bahwa algoritma pembelajaran mesin dapat digunakan secara efektif untuk membantu proses diagnosis awal diabetes berbasis data secara otomatis. Metode ini diharapkan dapat membantu pengambilan keputusan medis, terutama tentang pencegahan dan penanganan diabetes tipe 2 dini.

Kata Kunci—Deteksi Dini, Diabetes, Random Forest, SMOTE, Machine Learning.

Submitted :23 September 2025 | Accepted : 11 November 2025 | Published : 12 November 2025

1. PENDAHULUAN

Penyakit diabetes melitus merupakan gangguan metabolisme kronis yang ditandai oleh tingginya kadar gula darah akibat gangguan produksi atau kerja insulin. Menurut World Health Organization (WHO), jumlah penderita diabetes terus meningkat setiap tahunnya, menempatkan penyakit ini sebagai salah satu ancaman besar bagi kesehatan masyarakat di seluruh dunia. Tren serupa juga terlihat di Indonesia dengan peningkatan prevalensi pada usia produktif. Komplikasi serius seperti penyakit jantung, gagal ginjal, dan stroke dapat dicegah dengan mendeteksi diabetes sejak dini.[1]

Pemanfaatan teknologi *machine learning* dalam proses diagnosis telah menunjukkan hasil yang menguntungkan untuk mengatasi masalah ini. Algoritma Hutan Random (RF) adalah salah satu metode klasifikasi yang efektif untuk menganalisis data kesehatan, terutama untuk memprediksi penyakit diabetes[2]. RF memiliki keunggulan dalam manajemen dataset yang kompleks dan memiliki kemampuan untuk mengurangi risiko overfitting[3].

Namun, ketidakseimbangan data antara penderita dan non-penderita adalah masalah dalam klasifikasi data kesehatan seperti diabetes. Hal ini dapat menyebabkan prediksi yang tidak akurat terhadap kelas mayoritas. Oleh karena itu, untuk meningkatkan kinerja model, banyak penelitian telah

menggunakan teknik balancing data sepert Synthetic Minority Over-sampling Techniquei (SMOTE)[4][5].

Random Forest digunakan sebagai model utama dalam beberapa penelitian, dan digunakan bersama dengan teknik lain untuk meningkatkan ketepatan dan interpretabilitas. Misalnya, [6] menggabungkan RF dengan ADASYN dan SHAP untuk menghasilkan prediksi yang akurat dan dapat dijelaskan. [7] membandingkan kinerja RF dan CatBoost dalam klasifikasi diabetes dengan pendekatan SMOTE sebagai metode balancing data. Di sisi lain, [8] melakukan penelitian lain yang meningkatkan kemampuan klasifikasi data yang tidak seimbang dengan menggunakan pendekatan kelompok.

Untuk mengoptimalkan kinerja model RF, pengaturan hyperparameter seperti jumlah pohon (n_estimators), kedalaman maksimum (max_depth), dan jumlah sampel minimum pada daun (min_samples_leaf) sangat penting[9]. Menurut penelitian[10], hasil prediksi yang lebih stabil dan dapat dipercaya diperoleh dengan menggabungkan balancing data dan kecerdasan buatan yang dapat dijelaskan.

Dalam penelitian ini, algoritma Random Forest digunakan untuk membangun sistem prediksi dini diabetes dengan pendekatan balancing data SMOTE dan optimasi parameter. Diharapkan sistem ini dapat memberikan hasil prediksi yang akurat dan membantu pengambilan keputusan dalam layanan Kesehatan.

Penelitian tentang mendeteksi diabetes secara dini menggunakan pendekatan machine learning sudah dilakukan dengan berbagai cara.[11] Mereka membandingkan beberapa algoritma klasifikasi seperti Logistic Regression, SVM, KNN, Random Forest, XGBoost, LightGBM, dan CatBoost menggunakan dataset Pima. Sebelum memulai pembelajaran, mereka melakukan beberapa langkah pra-pemrosesan, yaitu mengisi nilai yang kosong, melakukan standarisasi data, memilih fitur yang penting, serta mengurangi jumlah dimensi dengan metode PCA. Hasil penelitian menunjukkan bahwa algoritma ensemble seperti Random Forest, XGBoost, dan CatBoost memberikan hasil terbaik, terutama ketika digunakan bersamaan dengan teknik feature engineering dan penyesuaian parameter hyperparameter.

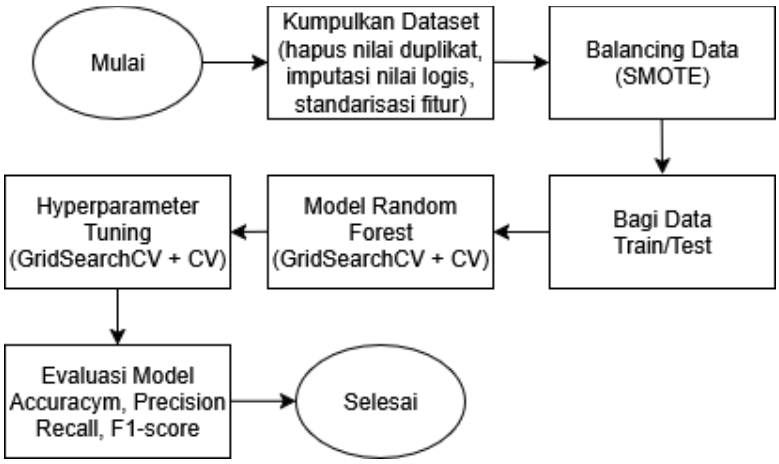
Selain memilih algoritma, keseimbangan data juga sangat memengaruhi hasil prediksi.[12]Membandingkan beberapa metode penyeimbangan data, yaitu RUS, UPS, SMOTE, ADASYN, SMOTE-Tomek, dan SMOTEENN, dengan algoritma Random Forest dan XGBoost. Hasilnya menunjukkan bahwa gabungan SMOTE-Tomek dan SMOTEENN mampu memberikan keseimbangan yang lebih baik antara precision dan recall.[13] Menggunakan SMOTE-Tomek Link di Random Forest lebih efektif dibandingkan SMOTE biasa, karena mampu mengurangi data yang tidak pasti dan meningkatkan kemampuan model dalam mengenali kelas minoritas.

Kualitas data dasar juga sangat penting dalam meningkatkan tingkat keakuratan prediksi.[14] Mereka mengusulkan cara untuk meningkatkan kualitas dataset Pima yang sering memiliki data yang hilang dan tidak seimbang. Pendekatan ini melibatkan membersihkan data, menyamakan jumlah kelas, dan memperkaya fitur. Hasil penelitian menunjukkan bahwa dengan meningkatkan kualitas data, performa model jauh lebih baik dibandingkan hanya menggunakan metode pembersihan data biasa.

Pendekatan yang lebih modern adalah menggabungkan deep learning dengan metode augmentasi.[15] Menggunakan jaringan CNN yang digabungkan dengan Sparse Autoencoder untuk meningkatkan fitur dan Variational Autoencoder untuk meningkatkan data. Model gabungan ini memberikan akurasi lebih dari 92% pada dataset Pima, jauh lebih baik dibandingkan metode machine learning biasa. Hasil ini menunjukkan bahwa kombinasi deep learning dengan teknik augmentasi data dapat meningkatkan efektivitas sistem deteksi dini diabetes secara signifikan.

Vol. 01, No. 01, November 2025
Hal. 1 – 9

2. METODE



Gambar 1. Alur Metode Penelitian

Alur proses penelitian ditampilkan pada Gambar 1. Langkah pertama adalah preprocessing data untuk membersihkan dataset Pima Indian dari nilai-nilai yang kosong atau tidak masuk akal, seperti tekanan darah yang bernilai 0. Nilai-nilai tersebut diganti dengan NaN, kemudian diisi dengan nilai rata-rata atau median, dan selanjutnya distandarisasi agar semua fitur memiliki skala yang sama. Proses pengisian nilai yang tidak logis bertujuan untuk mengganti nilai-nilai aneh, seperti BMI yang bernilai 0, dengan nilai median dari fitur tersebut, sehingga distribusi data tetap realistis dan tidak mengganggu hasil analisis. Setelah preprocessing selesai, dataset dibagi menjadi dua bagian, yaitu data latih dan data uji, agar pengevaluasian performa model dapat berjalan secara adil. Selanjutnya, digunakan metode Synthetic Minority Over-sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan antar kelas dengan menambahkan sampel-sampel sintetis pada kelas minoritas, yaitu kelas penderita diabetes, melalui interpolasi antar data, sehingga distribusi kelas menjadi lebih seimbang.

Random Forest dipilih sebagai model utama karena mampu mengolah data yang rumit, lebih stabil dibandingkan hanya menggunakan satu pohon keputusan, serta bisa menunjukkan fitur mana yang paling berpengaruh dalam proses pengambilan keputusan. Prediksi dilakukan dengan cara mengambil keputusan mayoritas dari banyak pohon keputusan yang ada. Untuk meningkatkan kinerja model, digunakan GridSearchCV yang mengatur parameter-parameter model secara terencana dengan mencoba berbagai variasi nilai parameter, seperti jumlah pohon (n_estimators) dan tingkat kedalaman pohon (max_depth), menggunakan metode validasi silang k-fold untuk menemukan kombinasi parameter terbaik. Evaluasi model dilakukan dengan empat metrik utama, yaitu akurasi (jumlah prediksi yang benar), precision (tingkat ketepatan dalam mengklasifikasikan kasus positif), recall (kemampuan model dalam mendeteksi kasus positif), dan F1-score (rata-rata harmonis dari precision dan recall). Evaluasi ini bertujuan agar performa model dapat dinilai secara lebih menyeluruh.

3. HASIL DAN PEMBAHASAN

Setelah melewati proses pra-pemrosesan, mengimbangi data menggunakan SMOTE, dan melakukan pelatihan dengan algoritma Random Forest, diperoleh model yang mampu mengenali pola-pola penting dalam data diabetes. Faktor seperti kadar glukosa, indeks massa tubuh, dan usia terbukti menjadi variabel utama yang memengaruhi hasil prediksi. Hasil ini sesuai dengan penelitian sebelumnya yang menunjukkan bahwa fitur-fitur tersebut sangat penting dalam mendiagnosis diabetes, karena ketiganya berkaitan langsung dengan kondisi metabolisme tubuh dan tingkat risiko terkena penyakit diabetes.

3.1 Hasil Preprocessing Data

Untuk memastikan bahwa data bersih dan siap digunakan oleh algoritma machine learning, tahap preprocessing sangat penting sebelum model dilatih.

- Cek informasi Dataset

  Pengecekan informasi dataset dilakukan agar kita tahu struktur serta ciri-ciri data yang digunakan. Dalam Gambar 2, terlihat bahwa dataset ini memiliki 768 data dengan 9 kolom, yaitu Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, dan Outcome. Semua kolom memiliki jumlah data yang lengkap, yaitu 768, sehingga tidak ada data yang hilang atau kosong. Tipe data yang digunakan adalah int64 untuk variabel numerik diskrit seperti jumlah kehamilan, tekanan darah, dan usia, serta float64 untuk variabel numerik kontinu seperti BMI dan DiabetesPedigreeFunction. Informasi ini penting untuk memastikan bahwa data memiliki kualitas yang baik sebelum dilakukan proses preprocessing lebih lanjut.

```
 Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None
```

Gambar 2. Informasi Dataset

- Cek jumlah nilai 0 pada kolom yang seharusnya tidak bernilai 0

  Mengecek ada berapa nilai 0 di kolom yang seharusnya tidak boleh bernilai nol membantu menemukan data yang tidak masuk akal. Pada Gambar 3, tampak beberapa kolom penting masih berisi nilai 0, yaitu Glucose sebanyak 5 data, BloodPressure sebanyak 35 data, SkinThickness sebanyak 227 data, Insulin sebanyak 374 data, dan BMI sebanyak 11 data. Nilai 0 di kolom tersebut tidak tepat karena secara medis, seseorang tidak mungkin memiliki kadar glukosa, tekanan darah, ketebalan kulit, insulin, atau indeks massa tubuh bernilai nol. Oleh karena itu, data ini harus diperbaiki pada tahap preprocessing, misalnya dengan mengisi nilai 0 menggunakan rata-rata, median, atau metode lain yang sesuai agar kualitas data menjadi lebih baik untuk pembuatan model.

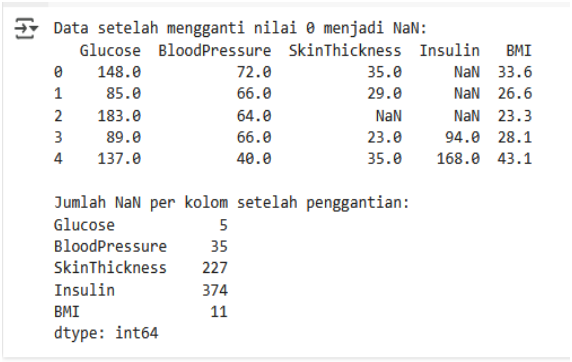
```
 Jumlah nilai 0 pada kolom penting:
Glucose            5
BloodPressure     35
SkinThickness    227
Insulin          374
BMI               11
dtype: int64
```

Gambar 3. Data Nol per Kolom

3.1.1 Mengganti Nilai 0 menjadi NaN

Langkah berikutnya adalah mengubah nilai 0 pada kolom-kolom yang seharusnya tidak mungkin bernilai nol menjadi NaN agar lebih mudah diolah saat proses pengisian ulang data. Pada Gambar 4, terlihat bahwa setelah dilakukan perubahan, data pada kolom Glucose, BloodPressure, SkinThickness, Insulin, dan BMI yang sebelumnya bernilai 0 kini ditandai sebagai NaN. Hasil

perhitungan menunjukkan jumlah NaN per kolom sama dengan jumlah nilai 0 sebelumnya, yaitu Glucose sebanyak 5, BloodPressure sebanyak 35, SkinThickness sebanyak 227, Insulin sebanyak 374, dan BMI sebanyak 11. Dengan langkah ini, dataset menjadi lebih representatif karena data yang tidak logis sudah diubah menjadi nilai yang hilang, sehingga dapat diproses lebih lanjut menggunakan metode pengisian data yang sesuai.



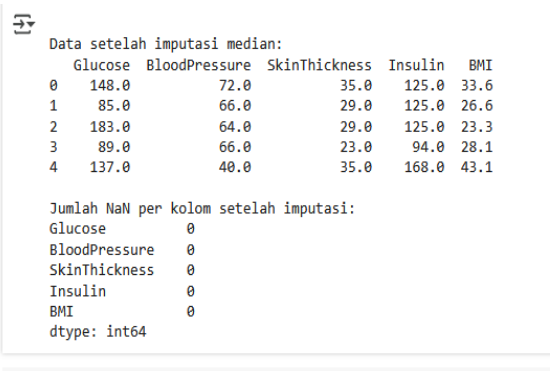
```
Data setelah mengganti nilai 0 menjadi NaN:
   Glucose  BloodPressure  SkinThickness  Insulin   BMI
0    148.0           72.0           35.0      NaN  33.6
1     85.0           66.0           29.0      NaN  26.6
2    183.0           64.0            NaN      NaN  23.3
3     89.0           66.0           23.0     94.0  28.1
4    137.0           40.0           35.0    168.0  43.1

Jumlah NaN per kolom setelah penggantian:
Glucose            5
BloodPressure     35
SkinThickness    227
Insulin          374
BMI               11
dtype: int64
```

Gambar 4. Hasil Jumlah Nilai Na N per Kolom Setelah Penggantian

3.1.2 Imputasi nilai NaN dengan median

Imputasi dilakukan untuk menggantikan nilai yang hilang atau kosong (NaN) dengan nilai yang lebih tepat representatif, agar dataset bisa digunakan dalam proses pembuatan model. Pada Gambar 5, terlihat bahwa metode yang digunakan adalah median, yaitu menggantikan nilai NaN dengan nilai tengah dari setiap kolom. Setelah dilakukan imputasi, jumlah NaN di semua kolom (Glucose, BloodPressure, SkinThickness, Insulin, dan BMI) berkurang menjadi 0, artinya semua data sudah lengkap. Median dipilih karena lebih tahan terhadap nilai ekstrem (outlier) dibandingkan rata-rata, sehingga distribusi data tetap seimbang dan kualitas dataset meningkat untuk analisis serta pemodelan selanjutnya.



```
Data setelah imputasi median:
   Glucose  BloodPressure  SkinThickness  Insulin   BMI
0    148.0           72.0           35.0    125.0  33.6
1     85.0           66.0           29.0    125.0  26.6
2    183.0           64.0           29.0    125.0  23.3
3     89.0           66.0           23.0     94.0  28.1
4    137.0           40.0           35.0    168.0  43.1

Jumlah NaN per kolom setelah imputasi:
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
dtype: int64
```

Gambar 5. Hasil Nilai NaN Setelah Imputasi

3.1.3 Standarisasi Fitur

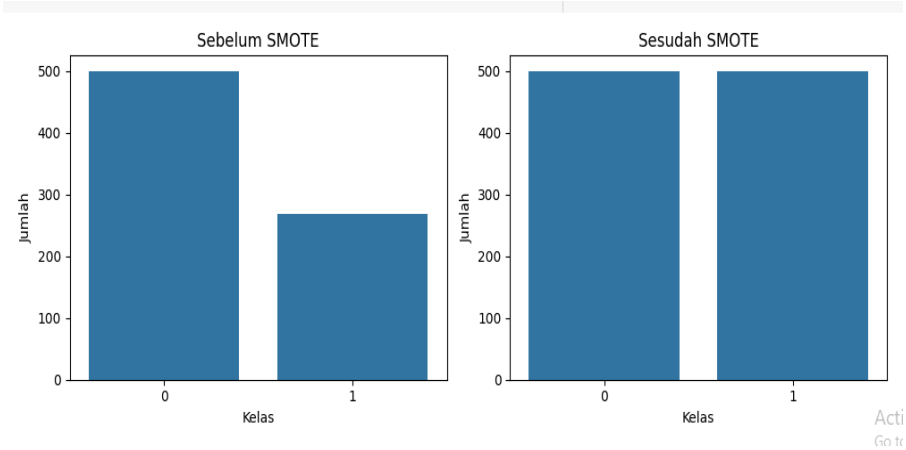
Standarisasi fitur dilakukan agar semua variabel memiliki skala yang sama, sehingga tidak ada fitur yang terlalu besar atau kecil sehingga memengaruhi hasil pemodelan. Dalam Gambar 6, proses standarisasi dilakukan dengan StandardScaler(), yang mengubah data agar memiliki rata-rata (mean) sama dengan nol dan standar deviasi sama dengan satu. Hasil perubahan ini disimpan dalam variabel X_scaled, lalu digunakan pada tahap pembuatan model. Langkah ini sangat penting terutama untuk algoritma yang mengandalkan jarak antar data, seperti K-Nearest Neighbors, Support Vector Machine, atau metode optimasi yang bergantung pada gradien, karena dapat membuat model bekerja lebih baik.

```
[16] # Standardisasi fitur
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)
```

Gambar 6. Standarisasi Fitur

3.1.4 Balancing Kelas Menggunakan SMOTE

Di dalam dataset ini terdapat ketidakseimbangan antar kelas, yaitu jumlah data pada kelas 0 jauh lebih banyak dibandingkan kelas 1. Hal ini bisa menyebabkan model cenderung mengenali pola dari kelas yang lebih banyak terlebih dahulu. Untuk mengatasi masalah tersebut, digunakan metode SMOTE (Synthetic Minority Oversampling Technique) yang bertujuan menambah data buatan pada kelas minoritas agar jumlah data setiap kelas menjadi seimbang. Pada Gambar 7, grafik sebelah kiri menunjukkan kondisi awal sebelum diterapkan SMOTE, di mana data kelas 0 lebih dominan. Setelah dilakukan SMOTE (grafik sebelah kanan), jumlah data kelas 0 dan kelas 1 menjadi sama. Dengan proses penyeimbangan ini, model diharapkan dapat mempelajari pola dari kedua kelas secara lebih baik dan menghasilkan prediksi yang lebih akurat.



Gambar 7. Hasil Balancing Kelas Menggunakan SMOTE

3.2 Pemodelan Menggunakan Random Forest

3.2.1 Optimasi hyperparameters

Optimasi hyperparameter dilakukan untuk menemukan kombinasi parameter terbaik yang bisa meningkatkan hasil kerja model. Dalam Gambar 8, proses optimasi menggunakan GridSearchCV dengan model Random Forest Classifier. Beberapa parameter yang diuji meliputi jumlah pohon (n_estimators), kedalaman maksimum setiap pohon (max_depth), jumlah minimum sampel yang diperlukan untuk membagi node (min_samples_split), jumlah minimum sampel di setiap daun (min_samples_leaf), serta jumlah fitur maksimum yang diperiksa saat membagi node (max_features). Untuk memastikan hasil yang lebih akurat dan konsisten, proses validasi silang digunakan. Setelah dilakukan pelatihan dengan berbagai kombinasi parameter, GridSearchCV memilih model terbaik dan menyimpannya dalam variabel best_model. Dengan cara ini, model yang dihasilkan lebih efektif dalam memproses data dan diharapkan bisa memberikan hasil prediksi yang lebih baik.

```
param_grid = {
    'n_estimators': [200, 300],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
    'max_features': ['sqrt']
}

cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

grid_search = GridSearchCV(
    estimator=RandomForestClassifier(random_state=42, class_weight='balanced'),
    param_grid=param_grid,
    cv=cv,
    scoring='f1',
    n_jobs=-1,
    verbose=1
)

grid_search.fit(X_train, y_train)
best_model = grid_search.best_estimator_
```

Gambar 8. Optimasi Hyperparameters

3.2.2 Best hyperparameters

Hasil dari proses optimasi menunjukkan kombinasi parameter terbaik yang bisa meningkatkan kinerja model. Dalam Gambar 9, ditemukan parameter terbaik untuk model Random Forest, yaitu max_depth = 10, max_features = 'sqrt', min_samples_leaf = 1, min_samples_split = 2, dan n_estimators = 300. Parameter-parameter ini membantu menjaga keseimbangan antara tingkat kesulitan model dan kemampuannya dalam beradaptasi dengan data baru. Dengan pembatasan kedalaman pohon, jumlah estimator yang cukup banyak, serta pembagian data yang optimal, model diharapkan bisa membuat prediksi yang lebih tepat dan konsisten tanpa mengalami kelebihan sesuaian dengan data pelatihan.

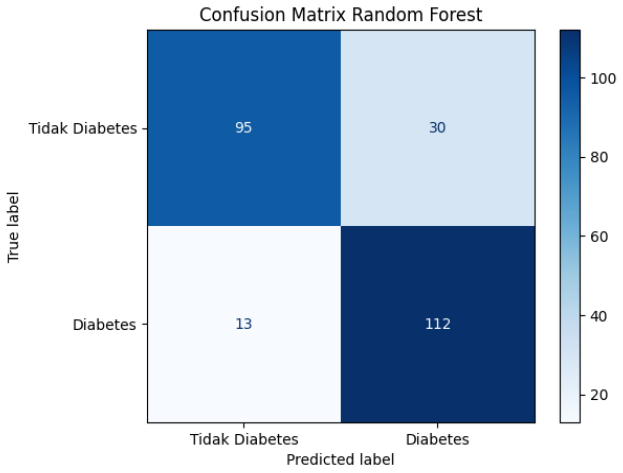
```
Best Hyperparameters:
{'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}
```

Gambar 9. Best Hyperparameters

3.3 Evaluasi Model

3.3.1 Confusion Matrix

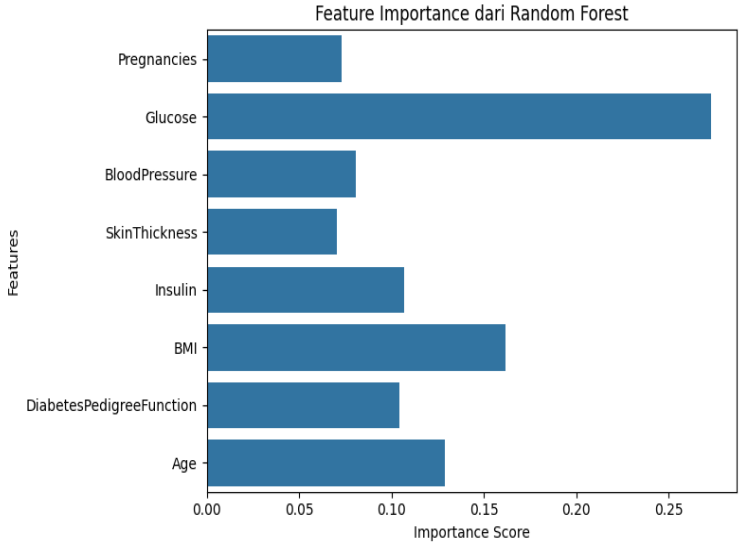
Confusion matrix digunakan untuk mengevaluasi seberapa baik model klasifikasi bekerja dengan membandingkan hasil prediksi model dengan label yang benar. Dalam Gambar 10, model Random Forest berjalan cukup baik, dengan 95 data dinyatakan benar sebagai Tidak Diabetes (True Negative), 112 data dinyatakan benar sebagai Diabetes (True Positive), 30 data salah dinyatakan sebagai Diabetes (False Positive), dan 13 data salah dinyatakan sebagai Tidak Diabetes (False Negative). Hasil ini didukung oleh laporan classification yang menunjukkan nilai precision, recall, dan f1-score yang cukup seimbang, yaitu untuk kelas Tidak Diabetes (precision 0.88, recall 0.76, f1-score 0.82) dan kelas Diabetes (precision 0.79, recall 0.90, f1-score 0.84). Secara keseluruhan, model mencapai tingkat akurasi 0.83, menunjukkan kemampuan yang baik dan seimbang dalam membedakan kedua kelas.



Gambar 10. Hasil Confusion Matrix

3.4 Feature Importance

Gambar 11 menampilkan pentingnya setiap fitur dalam model Random Forest yang digunakan. Dari grafik tersebut terlihat bahwa variabel Glucose memiliki bobot tertinggi dibandingkan fitur lainnya, artinya kadar glukosa merupakan faktor utama yang memengaruhi hasil prediksi. Beberapa fitur lainnya seperti BMI dan Age juga memiliki dampak yang cukup besar, sementara variabel seperti SkinThickness, BloodPressure, dan Pregnancies memiliki pengaruh yang lebih kecil. Dengan demikian, Gambar 11 menunjukkan bahwa model Random Forest lebih banyak bergantung pada variabel-variabel utama seperti Glucose, BMI, dan Age untuk membuat prediksi, sedangkan variabel-variabel lainnya meskipun tetap digunakan, kontribusinya tidak sebesar tiga variabel utama tersebut.



Gambar 11. Hasil Feauture Importance

4. KESIMPULAN

Berdasarkan hasil penelitian, model Random Forest mampu mendeteksi diabetes dengan tingkat akurasi sebesar 82,80%. Penggunaan teknik SMOTE berhasil menangani masalah ketidakseimbangan antar kelas, sedangkan penyesuaian parameter hyperparameter secara spesifik meningkatkan kinerja dan kemampuan model dalam menggeneralisasi hasil. Faktor-faktor yang berpengaruh besar dalam prediksi, yaitu kadar glukosa, indeks massa tubuh (BMI), dan usia, terbukti memiliki hubungan medis yang signifikan dengan risiko terkena diabetes. Secara keseluruhan, hasil evaluasi menunjukkan bahwa model ini cukup akurat dan bisa menjadi alat bantu skrining otomatis yang efektif untuk mendeteksi diabetes secara dini.

DAFTAR PUSTAKA

[1] D. C. P. Buani, "Deteksi Dini Penyakit Diabetes dengan Menggunakan Algoritma Random Forest," *EVOLUSI J. Sains dan Manaj.*, vol. 12, no. 1, pp. 1–8, 2024, doi: 10.31294/evolusi.v12i1.21005.

[2] A. Syahri, U. Fariha, R. Afandi, and I. Nurliyana, "Comparison of Logistic Regression, Random Forest and Adaboost Algorithms for Diabetes Mellitus Classification," *IJATIS Indones. J. Appl. Technol. Innov. Sci.*, vol. 1, no. 1, pp. 41–46, 2024, doi: 10.57152/ijatis.v1i1.1116.

[3] A. Aji Septa, Amar Al Farizi, Anas Nur Khafid, Didi Prasetyo, Nur Cholis Romadhon, and Fandy Setyo Utomo, "Diabetes Detection Optimisation with Hyperparameter Tuning in Random Forest Algorithm," *J. Informatics Interact. Technol.*, vol. 1, no. 3, pp. 165–177, 2024, doi: 10.63547/jiite.v1i3.42.

[4] I. Nurzari, E. Sari, D. I. Harris, A. M. Priyatno, and H. Rusnedy, "Inter-Cluster Distance-Based Smote Modification for Enhanced Diabetes Classification," *J. Eng. Technol. Ind. Appl.*, vol. 11, no. 51, pp. 190–196, 2025, doi: 10.5935/jetia.v11i51.1453.

[5] N. Nurussakinah, M. Faisal, and I. B. Santoso, "Algoritma Random Forest dan Synthetic Minority Oversampling Technique (SMOTE) untuk Deteksi Diabetes," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 10, no. 2, pp. 221–234, 2025, doi: 10.14421/jiska.2025.10.2.221-234.

[6] H. Aulia, A. Wibowo, and S. Sutrisno, "Integration of Random Forest, ADASYN, and SHAP for Diabetes Prediction and Interpretation," *Sci. J. Informatics*, vol. 12, no. 2, pp. 211–222, 2025, doi: 10.15294/sji.v12i2.24314.

[7] R. Irfannandhy, L. B. Handoko, and N. Ariyanto, "Analisis Performa Model Random Forest dan CatBoost dengan Teknik SMOTE dalam Prediksi Risiko Diabetes," *Edumatic J. Pendidik. Inform.*, vol. 8, no. 2, pp. 714–723, 2024, doi: 10.29408/edumatic.v8i2.27990.

[8] Y. Jang, "Feature-based ensemble modeling for addressing diabetes data imbalance using the SMOTE, RUS, and random forest methods: a prediction study," *Ewha Med. J.*, vol. 48, no. 2, p. e32, 2025, doi: 10.12771/emj.2025.00353.

[9] A. Alsyar *et al.*, "Jurnal Computer Science and Information Technology (CoSciTech) Pemodelan Prediktif Diabetes Menggunakan Pendekatan Multimodel Machine Learning dan Deep Predictive Modeling of Diabetes Using Multimodel Machine learning and Deep learning Approaches," vol. 6, no. 2, pp. 158–165, 2025.

[10] K. Poorani, S. P. Balakannan, and M. Karuppasamy, "Mitigating Data Imbalance for Robust Diabetes Diagnosis Using Machine Learning and Explainable Artificial Intelligence," *J. Curr. Sci. Technol.*, vol. 15, no. 3, pp. 1–10, 2025, doi: 10.59796/jcst.V15N3.2025.111.

[11] A. A. Ali, G. R. Galal, and H. S. Hassan, "Diabetes Prediction on Pima Indian Dataset Using Machine Learning Techniques," vol. 11, no. 7, 2025.

[12] F. O. Aghware *et al.*, "Effects of Data Balancing in Diabetes Mellitus Detection: A Comparative XGBoost and Random Forest Learning Approach," *NIPES - J. Sci. Technol. Res.*, vol. 7, no. 1, pp. 1–11, 2025, doi: 10.37933/nipes/7.1.2025.1.

[13] H. Hairani, A. Anggrawan, and D. Priyanto, "Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote-Tomek Link," *Int. J. Informatics Vis.*, vol. 7, no. 1, pp. 258–264, 2023, doi: 10.30630/joiv.7.1.1069.

[14] M. F. Zamil, D. H. Hameed, and U. S. Mahmoud, "A Comprehensive Data Enhancement Method for the Pima Dataset to Improve Diabetes Prediction Performance," *J. Port Sci. Res.*, vol. 8, no. 4, pp. 314–320, 2025, doi: 10.36371/port.2025.4.1.

[15] M. Teresa García-Ordás, C. Benavides, J. Alberto Benítez-Andrades, H. Alaiz-Moretón, and I. García-Rodríguez, "Computer Methods and Programs in Biomedicine Diabetes detection using deep learning techniques with oversampling and feature augmentation," 2024.