PRINSIP : Portal Riset & Inovasi Sistem Perangkat Lunak
Volume 4, Nomor 3, Juli-September 2026
http://doi.org/10.59696/prinsip.v4i3.267
e-ISSN : 2985-802X
p-ISSN : 2987-0534

Implementasi Algoritma K-NN, *Decision Tree*, dan *Random Forest* pada Sistem Deteksi Diabetes Melitus

[1*]Muhamad Daniel Rivai, [2]Topan Sukma Saputra, [3]Nurasiah, [4]Muji Lestari, [5]Sandhi Prajaka, [6]Ali Akbar
[1,3,5] Prodi Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi,
[2] Prodi Teknik Informatika, Fakultas Teknologi Industri,
[4,6] Prodi Teknik Industri, Fakultas Teknologi Industri,
Universitas Gunadarma, Jakarta, Indonesia
daniel_rivai@staff.gunadarma.ac.id

Submit : 30 Jun 26 | **Diterima** : 18 Jul 2026 | **Terbit** : 21 Jul 2026

ABSTRAK

Diabetes Melitus (DM) merupakan penyakit metabolik yang ditandai oleh peningkatan kadar glukosa darah akibat gangguan produksi maupun efektivitas kerja insulin. Sebagai salah satu masalah kesehatan yang terus mengalami peningkatan dan berpotensi menyebabkan komplikasi serius hingga kematian, diperlukan metode klasifikasi yang mampu mengidentifikasi faktor-faktor yang berkaitan dengan kejadian Diabetes Melitus secara akurat. Penelitian ini bertujuan menganalisis dan membandingkan kinerja algoritma *K-Nearest Neighbors* (KNN), *Decision Tree*, dan *Random Forest* dalam mengklasifikasikan data Diabetes Melitus berdasarkan sejumlah atribut, meliputi usia, jenis kelamin, *polyuria*, *polydipsia*, *polyphagia*, *sudden weight loss*, *weakness*, *genital thrush*, *visual blurring*, *itching*, *irritability*, *delayed healing*, *partial paresis*, *muscle stiffness*, *alopecia*, dan *obesity*. Metode penelitian dilakukan melalui penerapan ketiga algoritma dengan evaluasi berdasarkan akurasi, presisi, *recall*, *F1-Score*, dan *Area Under Curve* (AUC). Hasil pengujian menunjukkan bahwa *Random Forest* memberikan performa terbaik dengan akurasi 92,16%, presisi 91,89%, *recall* 97,14%, *F1-Score* 94,44%, dan AUC 0,9768. Nilai tersebut lebih tinggi dibandingkan *Decision Tree* dengan akurasi 88,24% dan AUC 0,8464 serta KNN dengan akurasi 74,51% dan AUC 0,8393. Dengan demikian, *Random Forest* merupakan algoritma paling optimal dalam mengklasifikasikan dan membedakan pasien Diabetes Melitus berdasarkan atribut yang digunakan.

Kata Kunci: Diabetes Melitus, *K-Nearest Neighbors*, *Decision Tree*, *Random Forest*, Klasifikasi

PENDAHULUAN

Diabetes Melitus (DM) merupakan salah satu penyakit metabolik kronis dengan prevalensi yang terus meningkat dan menjadi tantangan kesehatan global. Penyakit ini ditandai oleh peningkatan kadar glukosa darah yang berlangsung secara persisten akibat gangguan metabolisme tubuh. Berdasarkan International Diabetes Federation (IDF, 2021), sekitar 537 juta orang dewasa di seluruh dunia hidup dengan diabetes dan jumlah tersebut diproyeksikan meningkat menjadi 643 juta pada tahun 2030. Tingginya prevalensi tersebut turut berkontribusi terhadap angka kematian global, dengan sekitar 1,5 juta kematian pada tahun 2019 yang secara langsung disebabkan oleh diabetes dan 2,2 juta kematian lainnya berkaitan dengan kadar glukosa darah yang tinggi (*World Health Organization* [WHO], 2022). Di Indonesia, prevalensi diabetes pada penduduk berusia 15 tahun ke atas juga mengalami peningkatan dari 6,9% menjadi 10,9% berdasarkan hasil Riskesdas.

Secara klinis, diabetes dapat ditandai oleh berbagai gejala, seperti poliuria, polidipsia, polifagia, penurunan berat badan secara tidak terduga, kelemahan tubuh, gangguan penglihatan, luka yang sulit sembuh, serta gejala lainnya yang berkaitan dengan gangguan metabolisme dan komplikasi penyakit (Kemenkes RI, 2023). Keragaman karakteristik klinis tersebut menunjukkan pentingnya pendekatan berbasis data untuk membantu proses identifikasi dan klasifikasi diabetes secara lebih efektif. Perkembangan *Machine Learning* memberikan peluang dalam bidang kesehatan untuk mengolah berbagai karakteristik klinis dan menghasilkan model klasifikasi yang dapat mendukung proses deteksi penyakit secara lebih cepat dan objektif (Marlim et al., 2022).

e-ISSN : 2985-802X
p-ISSN : 2987-0534

Sejumlah algoritma *Machine Learning* telah digunakan dalam klasifikasi penyakit, di antaranya *K-Nearest Neighbors* (KNN), *Random Forest*, dan *Decision Tree*. KNN merupakan algoritma berbasis *instance* yang melakukan prediksi berdasarkan kedekatan suatu data terhadap sejumlah data terdekat pada data pelatihan. Penelitian sebelumnya menunjukkan bahwa penerapan KNN dapat menghasilkan akurasi sebesar 79% pada nilai K=5 (Yasin et al., 2025). Sementara itu, *Random Forest* merupakan metode *ensemble learning* yang dapat digunakan untuk tugas klasifikasi dan regresi (Ho, 1995). Algoritma ini memiliki kemampuan dalam menangani data berukuran besar dan menghasilkan tingkat kesalahan klasifikasi yang relatif rendah (Breiman, 2001). Penelitian Triyono et al. (2021) menunjukkan bahwa model *Random Forest* mampu mencapai akurasi sebesar 98,27%, presisi 97,69%, dan *recall* 98%. Adapun *Decision Tree* membentuk struktur pohon keputusan berdasarkan karakteristik data untuk menghasilkan aturan klasifikasi terhadap data baru. Penggunaan *Decision Tree* juga telah menunjukkan kinerja yang baik dalam klasifikasi penyakit, termasuk pada penelitian Dewi (2023) yang membandingkan KNN, *Decision Tree*, dan *Random Forest* untuk diagnosis penyakit jantung. Hasil penelitian tersebut menunjukkan bahwa *Decision Tree* memberikan kinerja terbaik dengan akurasi 90%, presisi 88%, dan *recall* 93%.

Berdasarkan perkembangan tersebut, diperlukan penelitian yang membandingkan kemampuan beberapa algoritma *Machine Learning* dalam mengklasifikasikan Diabetes Melitus berdasarkan karakteristik klinis pasien. Penelitian ini menggunakan variabel yang merepresentasikan karakteristik demografis dan gejala klinis, meliputi usia, jenis kelamin, poliuria, polidipsia, polifagia, penurunan berat badan secara tiba-tiba, kelemahan, *genital thrush*, penglihatan kabur, gatal, mudah marah, keterlambatan penyembuhan luka, *partial paresis*, kekakuan otot, alopesia, obesitas, dan kelas diagnosis. Algoritma KNN, *Decision Tree*, dan *Random Forest* dibandingkan untuk memperoleh model klasifikasi yang memiliki kinerja terbaik. Evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, *F1-score*, dan *Area Under the Curve* (AUC). Hasil penelitian diharapkan dapat memberikan gambaran mengenai efektivitas ketiga algoritma dalam klasifikasi Diabetes Melitus serta mendukung pengembangan pendekatan berbasis data untuk membantu identifikasi kondisi diabetes secara lebih dini dan akurat.

TINJAUAN PUSTAKA

Diabetes Melitus

Diabetes Melitus (DM) merupakan penyakit metabolik kronis yang ditandai oleh peningkatan kadar glukosa darah (*hiperglikemia*) akibat gangguan produksi insulin, fungsi insulin, atau keduanya. Insulin merupakan hormon yang diproduksi oleh pankreas dan berperan dalam mengatur kadar glukosa darah serta membantu sel tubuh memanfaatkan glukosa sebagai sumber energi. Gangguan produksi atau penggunaan insulin yang berlangsung dalam jangka panjang dapat menyebabkan kerusakan pada pembuluh darah, saraf, ginjal, dan mata (*World Health Organization*, 2023). Berdasarkan karakteristiknya, DM terdiri atas beberapa tipe, di antaranya diabetes tipe 1, diabetes tipe 2, dan diabetes gestasional. Diabetes tipe 1 terjadi akibat kerusakan sel beta pankreas yang menyebabkan tubuh mengalami kekurangan insulin, sedangkan diabetes tipe 2 berkaitan dengan resistensi insulin yang dapat disertai penurunan produksi insulin. Gejala umum DM meliputi frekuensi buang air kecil yang meningkat, rasa haus berlebihan, penurunan berat badan tanpa sebab yang jelas, serta luka yang sulit sembuh (Alberti & Zimmet, 1998). Pengelolaan DM memerlukan pendekatan komprehensif melalui perubahan gaya hidup, pengaturan pola makan, aktivitas fisik, serta penggunaan obat antidiabetes atau insulin sesuai kondisi pasien. Penatalaksanaan yang tepat diperlukan untuk menjaga kadar glukosa darah dan menurunkan risiko komplikasi sehingga dapat meningkatkan kualitas hidup pasien (Perkeni, 2021).

Machine Learning

Machine Learning (ML) merupakan salah satu cabang kecerdasan buatan (*Artificial Intelligence*/AI) yang mengembangkan metode komputasi agar sistem mampu mempelajari pola dari data dan meningkatkan kinerjanya tanpa memerlukan pemrograman eksplisit untuk setiap tugas. ML dapat digunakan untuk berbagai kebutuhan, seperti prediksi, klasifikasi, dan pengambilan keputusan berdasarkan pola yang diperoleh dari data (Alpaydin, 2020). Proses pembelajaran dilakukan dengan memanfaatkan data pelatihan (*training data*) untuk membangun model matematis yang mampu mengenali hubungan atau pola tertentu. Model yang telah terbentuk selanjutnya digunakan untuk melakukan prediksi atau klasifikasi terhadap data baru. Dengan demikian, kualitas dan karakteristik

e-ISSN : 2985-802X
p-ISSN : 2987-0534

data pelatihan menjadi faktor penting yang memengaruhi kemampuan model dalam menghasilkan prediksi yang akurat (Universitas Malikussaleh, 2021). Dalam bidang kesehatan, penerapan ML dapat dimanfaatkan untuk membantu proses analisis data medis dan pengembangan sistem prediksi penyakit, termasuk klasifikasi risiko Diabetes Melitus berdasarkan karakteristik pasien.

Random Forest

Random Forest merupakan algoritma *machine learning* berbasis *ensemble* yang menggabungkan sejumlah *Decision Tree* untuk menghasilkan prediksi yang lebih stabil dan akurat. Algoritma ini membangun setiap pohon menggunakan sampel *bootstrap* dari data pelatihan dan subset fitur yang dipilih secara acak pada setiap proses pemisahan (*split*). Hasil prediksi akhir diperoleh melalui mekanisme pemungutan suara mayoritas (*majority voting*) pada kasus klasifikasi (Heryadi & Wahyuno, 2020; Liu et al., 2012). Pendekatan tersebut memungkinkan *Random Forest* mengurangi risiko *overfitting* yang dapat terjadi pada satu *Decision Tree* dan meningkatkan kemampuan generalisasi model. Secara umum, proses *Random Forest* dimulai dengan pengambilan sampel *bootstrap* secara acak dari data pelatihan. Selanjutnya, pada setiap *node*, sejumlah fitur dipilih secara acak untuk menentukan pemisahan data terbaik berdasarkan fungsi objektif tertentu. Proses pembentukan pohon dilakukan secara berulang hingga terbentuk sejumlah *Decision Tree*. Pada tahap akhir, hasil klasifikasi dari seluruh pohon digabungkan melalui mekanisme *majority voting* untuk menentukan kelas prediksi akhir. Penerapan *Random Forest* pada prediksi Diabetes Melitus menunjukkan kinerja yang baik. Penelitian Sriyanto dan Supriyatna (2023), misalnya, melaporkan akurasi sebesar 99,3%, *recall* 99,5%, presisi 99,1%, *F1-Score* 99%, dan *AUC* 100%. Hasil tersebut menunjukkan bahwa *Random Forest* memiliki potensi yang kuat dalam mendukung klasifikasi dan prediksi penyakit diabetes.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) merupakan algoritma klasifikasi berbasis *instance-based learning* yang menentukan kelas suatu data baru berdasarkan kelas mayoritas dari sejumlah k tetangga terdekat pada data pelatihan. Algoritma ini termasuk *lazy learning* karena tidak membangun model eksplisit selama proses pelatihan, melainkan melakukan perhitungan kedekatan ketika proses prediksi dilakukan. Oleh karena itu, KNN membutuhkan data pelatihan yang memadai agar mampu menghasilkan klasifikasi yang baik (Begum et al., 2016). Proses klasifikasi KNN diawali dengan menentukan nilai k, kemudian menghitung jarak antara data baru dan seluruh data pelatihan untuk memperoleh k data dengan jarak terdekat. Selanjutnya, frekuensi kelas dari k tetangga tersebut dihitung dan kelas dengan jumlah terbanyak ditetapkan sebagai kelas data baru (Heryadi & Wahyuno, 2020). Salah satu metode yang umum digunakan untuk menghitung jarak adalah *Euclidean Distance*. Pada ruang berdimensi n, jarak *Euclidean* antara data latih dan data uji dapat dirumuskan sebagai berikut :

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Keterangan:
$d(x.y)$= jarak antara data $x$dan y
x_i= nilai atribut ke-i pada data pertama
y_i= nilai atribut ke-i pada data kedua
n= jumlah dimensi atau atribut data
Nilai jarak yang lebih kecil menunjukkan tingkat kedekatan yang lebih tinggi antara data baru dan data pelatihan. Selanjutnya, kelas data baru ditentukan berdasarkan kelas mayoritas dari k tetangga terdekat. Pemilihan nilai k menjadi salah satu faktor penting karena nilai yang terlalu kecil dapat meningkatkan sensitivitas terhadap *noise*, sedangkan nilai yang terlalu besar dapat mengurangi kemampuan model dalam membedakan karakteristik antar kelas.

Decision Tree

Decision Tree merupakan algoritma *machine learning* yang digunakan untuk membangun model prediksi dengan merepresentasikan proses pengambilan keputusan dalam struktur pohon.

e-ISSN : 2985-802X
p-ISSN : 2987-0534

Struktur tersebut terdiri atas *root node*, *internal node*, dan *leaf node*, dengan setiap percabangan merepresentasikan keputusan berdasarkan atribut tertentu. Algoritma ini memiliki keunggulan berupa kemudahan interpretasi dan kemampuan menangani data numerik maupun kategorikal dengan tahapan persiapan data yang relatif sederhana (Bell, 2020). Dalam proses pembentukan pohon, *Decision Tree* menentukan atribut terbaik untuk melakukan pemisahan data berdasarkan tingkat *impurity*. Salah satu ukuran yang umum digunakan adalah *entropy*, sedangkan pemilihan atribut dilakukan berdasarkan nilai *Information Gain*. Nilai *Information Gain* dapat dihitung menggunakan persamaan berikut (Ramadhan, 2021):

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy(S_i)$$

Keterangan:
S= himpunan data
A= atribut
n= jumlah partisi atribut A
$|\ S_i\ |$= jumlah data pada partisi ke-i
$|\ S\ |$= jumlah keseluruhan data pada himpunan S
Sementara itu, *entropy* digunakan untuk mengukur tingkat ketidakmurnian (*impurity*) suatu himpunan data dan dirumuskan sebagai berikut:

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

Keterangan:
S= himpunan data
n= jumlah kelas
p_i= proporsi data pada kelas ke-i.
Proses pembentukan pohon dilakukan secara berulang dengan memilih atribut yang memiliki nilai *Information Gain* terbaik hingga kondisi penghentian terpenuhi. Meskipun mudah diinterpretasikan, *Decision Tree* memiliki risiko *overfitting*, terutama ketika pohon berkembang terlalu kompleks dan terlalu menyesuaikan diri dengan data pelatihan (Bell, 2020). Oleh karena itu, pengendalian kompleksitas model diperlukan agar kemampuan generalisasi terhadap data baru tetap optimal.

Google Colab

Google Colab merupakan lingkungan pemrograman berbasis *cloud* yang memungkinkan pengguna menjalankan kode Python secara daring tanpa memerlukan instalasi lingkungan pengembangan secara lokal. Platform ini mendukung integrasi dengan penyimpanan seperti Google Drive dan repositori GitHub sehingga memudahkan pengelolaan dan akses data maupun kode penelitian (Sarosa et al., 2022). Selain menyediakan lingkungan komputasi berbasis *cloud*, *Google Colab* juga menawarkan akses terhadap sumber daya komputasi seperti *Graphics Processing Unit* (GPU) dan *Tensor Processing Unit* (TPU), yang dapat dimanfaatkan untuk mempercepat proses komputasi pada penelitian berbasis *machine learning* (Setiawan, 2021).

Pandas

Pandas merupakan pustaka Python yang digunakan untuk manipulasi, pengolahan, dan analisis data secara efisien. Pustaka ini menyediakan struktur data utama berupa *Series* dan *DataFrame* yang mendukung pengelolaan data dalam berbagai bentuk. *Pandas* juga menyediakan fungsi untuk menangani data yang tidak lengkap, tidak terstruktur, maupun tidak teratur, serta mendukung proses penggabungan, transformasi, dan analisis data (Id, 2021; Sarosa et al., 2022). Dalam penelitian berbasis *machine learning*, *Pandas* dapat digunakan untuk membaca dataset, melakukan pembersihan dan transformasi data, serta menyiapkan data sebelum digunakan dalam proses pemodelan.

Scikit-learn

Scikit-learn merupakan pustaka Python yang menyediakan berbagai algoritma *machine learning* untuk pembelajaran terawasi (*supervised learning*) maupun tidak terawasi (*unsupervised learning*). Pustaka ini menyediakan implementasi berbagai algoritma klasifikasi, regresi,

PRINSIP : Portal Riset & Inovasi Sistem Perangkat Lunak
Volume 4, Nomor 3, Juli-September 2026
http://doi.org/10.59696/prinsip.v4i3.267
e-ISSN : 2985-802X
p-ISSN : 2987-0534

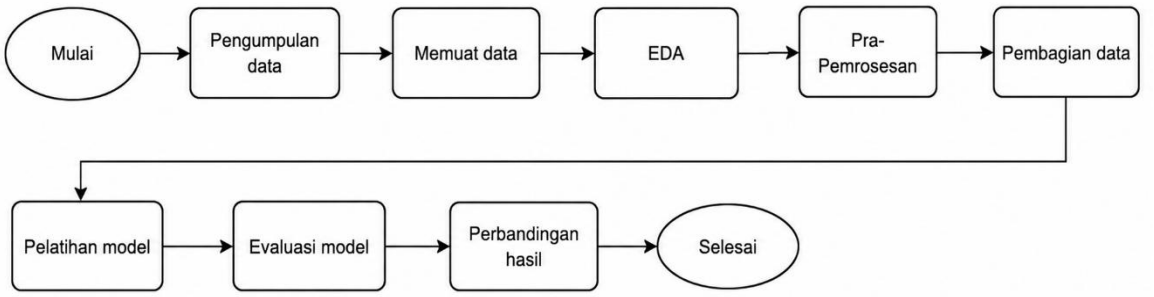
pengelompokan, serta fungsi pendukung untuk prapemrosesan data dan evaluasi model (Depari et al., 2022). *Scikit-learn* banyak digunakan dalam penelitian karena memiliki antarmuka pemrograman aplikasi (*Application Programming Interface*/API) yang konsisten, dokumentasi yang baik, serta kemudahan penggunaan dalam pengembangan dan pengujian model *machine learning* (Silitonga et al., 2019). Dalam penelitian klasifikasi penyakit, pustaka ini dapat digunakan untuk membangun model, melakukan pembagian data pelatihan dan pengujian, serta menghitung metrik evaluasi kinerja model.

Area Under Curve (AUC)

Evaluasi kinerja diperlukan untuk mengukur kemampuan model klasifikasi dalam menghasilkan prediksi yang tepat. Beberapa metrik yang umum digunakan meliputi akurasi, presisi, *recall*, *F1-Score*, dan *Area Under the Curve* (AUC). Akurasi menunjukkan proporsi prediksi yang benar terhadap keseluruhan data, sedangkan presisi mengukur ketepatan model dalam memprediksi kelas positif. *Recall* menunjukkan kemampuan model dalam mengidentifikasi seluruh data positif, sementara *F1-Score* merupakan rata-rata harmonis antara presisi dan *recall*. AUC digunakan untuk menilai kemampuan model dalam membedakan kelas positif dan negatif pada berbagai nilai ambang (*threshold*). Penggunaan AUC menjadi penting ketika distribusi kelas tidak seimbang karena memberikan gambaran kemampuan diskriminatif model secara lebih menyeluruh. Dalam penelitian klasifikasi penyakit diabetes, Wibisono et al. (2024) menunjukkan bahwa AUC dapat digunakan sebagai indikator pembanding kinerja antarmodel, dengan model *Gradient Boosting* memperoleh nilai AUC tertinggi dibandingkan model lainnya.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental untuk membangun dan membandingkan model *machine learning* dalam klasifikasi penyakit Diabetes Melitus (DM). Tiga algoritma yang digunakan adalah *K-Nearest Neighbors* (KNN), *Decision Tree*, dan *Random Forest Classifier*. Seluruh proses analisis dan pemodelan diimplementasikan menggunakan bahasa pemrograman Python melalui lingkungan *Google Colaboratory*, dengan pustaka *Pandas*, *NumPy*, *Matplotlib*, *Seaborn*, dan *Scikit-learn*. Alur penelitian terdiri atas beberapa tahapan seperti pada gambar 1.



Gambar 1. Alur Peneltian

Pengumpulan Data

Data penelitian menggunakan *dataset Early Classification of Diabetes* yang diperoleh dari platform Kaggle. Dataset terdiri atas 251 data dan 17 atribut yang merepresentasikan karakteristik demografis serta gejala yang berkaitan dengan Diabetes Melitus. Atribut yang digunakan meliputi *Age*, *Gender*, *Polyuria*, *Polydipsia*, *Sudden Weight Loss*, *Weakness*, *Polyphagia*, *Genital Thrush*, *Visual Blurring*, *Itching*, *Irritability*, *Delayed Healing*, *Partial Paresis*, *Muscle Stiffness*, *Alopecia*, *Obesity*, dan *Class*. Variabel *Class* digunakan sebagai target klasifikasi dengan dua kategori, yaitu *Negative* dan *Positive*.

Tabel 1. Atribut Dataset Diabetes

Label	Makna Label	Nilai
Age	Umur	*Continuous*
Gender	Jenis Kelamin	Perempuan, laki laki

e-ISSN : 2985-802X
p-ISSN : 2987-0534

Label	Makna Label	Nilai
Polyuria	Menghasilkan urine melebihi batas normal	0 = no 1 = yes
polydipsia	Merasakan rasa haus berlebihan	0 = no 1 = yes
Sudden weight loss	Kehilangan berat badan yang signifikan	0 = no 1 = yes
Weakness	Kelemahan yang dimiliki oleh suatu entitas	0 = no 1 = yes
Polyphagia	Merasakan nafsu makan meningkat bahkan setelah makan	0 = no 1 = yes
Genital thrush	Infeksi jamur pada kelamin	0 = no 1 = yes
Visual blurring	Kondisi Penglihatan kurang tajam	0 = no 1 = yes
Itching	Sensasi Gatal pada kulit	0 = no 1 = yes
Irritability	Mudah marah atau tersinggung	0 = no 1 = yes
Delayed healing	Kondisi luka membutuhkan waktu untuk sembuh	0 = no 1 = yes
Patial paresis	Kondisi otot menjadi lemah	0 = no 1 = yes
Muscle stiffness	Kondisi otot terasa kaku,tegang sulit digerakan	0 = no 1 = yes
Alopecia	Kerontokan rambut	0 = no 1 = yes
Obesity	Penumpukan lemak berlebih	0 = no 1 = yes
Class	Melihat hasil positive negative	0 = negative 1 = positive

Eksplorasi Data

Tahap *Exploratory Data Analysis* (EDA) dilakukan untuk memahami karakteristik awal dataset sebelum pemodelan. Analisis mencakup pemeriksaan struktur dan dimensi data, tipe atribut, statistik deskriptif, distribusi variabel numerik, serta distribusi kategori pada variabel demografis, gejala, dan kelas target. Visualisasi data digunakan untuk mengidentifikasi pola distribusi dan mengetahui proporsi kelas *Positive* dan *Negative* sebagai dasar dalam memahami karakteristik dataset.

Prapemrosesan Data

Prapemrosesan dilakukan untuk menyiapkan data agar dapat diproses oleh algoritma klasifikasi. Tahap ini meliputi pemeriksaan *missing values* dan pengodean variabel kategorikal. Hasil pemeriksaan menunjukkan tidak terdapat nilai hilang dalam dataset sehingga tidak diperlukan proses imputasi. Selanjutnya, atribut bertipe kategorikal, seperti *Yes/No* dan *Male/Female*, dikonversi ke bentuk numerik menggunakan *LabelEncoder*. Setelah proses pengodean, data dipisahkan menjadi variabel prediktor X dan variabel target y, dengan atribut *Class* sebagai target klasifikasi.

Pembagian Data

Dataset dibagi menjadi data latih (*training data*) dan data uji (*testing data*) dengan proporsi 80:20. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengevaluasi kemampuan generalisasi model terhadap data yang belum digunakan selama proses pelatihan. Pembagian data dilakukan menggunakan fungsi *train_test_split* dengan *random state* sebesar 42 untuk menjaga konsistensi hasil eksperimen.

e-ISSN : 2985-802X
p-ISSN : 2987-0534

Pemodelan

Pemodelan dilakukan menggunakan tiga algoritma klasifikasi untuk memperoleh model dengan kinerja terbaik dalam memprediksi kondisi Diabetes Melitus.

K-Nearest Neighbors (KNN)

Model KNN dibangun menggunakan *KNeighborsClassifier* dengan jumlah tetangga terdekat (*n_neighbors*) sebesar 3. Model dilatih menggunakan data latih dan selanjutnya digunakan untuk menghasilkan prediksi pada data uji.

Decision Tree

Model *Decision Tree Classifier* dibangun menggunakan parameter *random_state* sebesar 42. Model dilatih pada data latih untuk mempelajari pola klasifikasi, kemudian digunakan untuk memprediksi kelas pada data uji.

Random Forest

Model *Random Forest Classifier* dibangun dengan jumlah *decision tree* (*n_estimators*) sebanyak 100 dan *random_state* sebesar 42. Model dilatih menggunakan data latih dan digunakan untuk melakukan klasifikasi pada data uji.

Evaluasi Model

Evaluasi dilakukan untuk membandingkan kinerja ketiga algoritma berdasarkan hasil prediksi terhadap data uji. Evaluasi menggunakan *confusion matrix* dan metrik klasifikasi yang meliputi akurasi, presisi, *recall*, dan *F1-Score*. Selain itu, nilai *Area Under the Curve* (AUC) digunakan untuk mengukur kemampuan model dalam membedakan kelas positif dan negatif. *Confusion matrix* menghasilkan empat komponen, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai dari masing-masing metrik dihitung menggunakan persamaan berikut:

Akurasi

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Presisi

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Keterangan :

TP(*True Positive*) : Merupakan jumlah data positif yang diprediksi dengan benar.
TN(*True Negative*) : Merupakan jumlah data negatif yang diprediksi dengan benar.
FP(*False Positive*) : Merupakan jumlah data negatif yang salah diprediksi sebagai positif.
FN(*False Negative*) : Merupakan jumlah data positif yang salah diprediksi sebagai negatif.

Seluruh hasil evaluasi ketiga algoritma kemudian dibandingkan berdasarkan akurasi, presisi, *recall*, *F1-Score*, dan AUC untuk menentukan model yang memiliki kinerja klasifikasi terbaik. Proses evaluasi dan perbandingan model menjadi tahap akhir untuk menentukan algoritma yang paling sesuai dalam melakukan klasifikasi Diabetes Melitus pada dataset yang digunakan.

ASIL DAN PEMBAHASAN

Hasil Eksplorasi Data

Dataset yang digunakan dalam penelitian ini merupakan dataset klasifikasi Diabetes Melitus yang diperoleh dari Kaggle. Dataset terdiri atas 251 data dengan 17 atribut yang merepresentasikan karakteristik demografis dan gejala yang berkaitan dengan Diabetes Melitus. Variabel target (*class*)

PRINSIP : Portal Riset & Inovasi Sistem Perangkat Lunak
Volume 4, Nomor 3, Juli-September 2026
http://doi.org/10.59696/prinsip.v4i3.267
e-ISSN : 2985-802X
p-ISSN : 2987-0534

terdiri atas dua kategori, yaitu *Positive* dan *Negative*. Eksplorasi data dilakukan untuk memahami karakteristik dataset sebelum digunakan dalam proses klasifikasi menggunakan algoritma KNN, *Decision Tree*, dan *Random Forest*. Tahapan eksplorasi mencakup pemeriksaan struktur data, distribusi kelas target, serta karakteristik atribut yang digunakan sebagai variabel prediktor.

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
count	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000	251.000000
mean	48.864542	0.637450	0.525896	0.494024	0.414343	0.633466	0.466135	0.266932	0.442231	0.505976	0.282869	0.498008	0.446215	0.390438	0.358566	0.175299	0.689243
std	12.526036	0.481697	0.500327	0.500963	0.493592	0.482820	0.499849	0.443241	0.497644	0.500963	0.451293	0.500995	0.498092	0.488823	0.480538	0.380982	0.463728
min	16.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	39.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	48.000000	1.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
75%	58.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000
max	90.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Gambar 2. Statistik Deskriptif Dataset

Pada gambar 2 menunjukkan visualisasi distribusi data numerik pada dataset *Diabetes* melalui histogram untuk setiap atribut bertipe data *integer* (*int*) dan *float*. Visualisasi ini digunakan untuk mengidentifikasi pola dan sebaran nilai pada setiap fitur, seperti *Age*, *Gender*, dan atribut lainnya, sehingga karakteristik distribusi data dapat dipahami dengan lebih baik sebelum dilakukan tahap analisis selanjutnya.

Hasil Pra-pemrosesan Data

Pra-pemrosesan dilakukan untuk memastikan dataset berada dalam format yang sesuai dengan kebutuhan algoritma *machine learning*. Tahapan ini meliputi pemeriksaan kualitas data, *data cleaning*, dan *encoding* terhadap atribut kategorikal. Pemeriksaan terhadap nilai hilang (*missing values*) menunjukkan bahwa tidak terdapat data kosong dalam dataset yang digunakan. Dengan demikian, tidak diperlukan proses imputasi atau penghapusan data akibat nilai yang hilang. Kondisi tersebut menunjukkan bahwa dataset telah memiliki kelengkapan data yang memadai untuk digunakan pada tahap pemodelan.

```
total_missing_values = df.isnull().sum().sum()
print(total_missing_values)
```

```
0
```

Gambar 3. Jumlah Data yang hilang

Tahap berikutnya adalah *encoding*, yaitu mengubah atribut kategorikal ke dalam bentuk numerik agar dapat diproses oleh algoritma klasifikasi. Transformasi ini diterapkan pada atribut yang memiliki nilai kategorikal, seperti jenis kelamin dan gejala dengan kategori *Yes/No*. Setelah proses *encoding*, seluruh atribut yang digunakan sebagai masukan model berada dalam format numerik sehingga dapat digunakan dalam proses pelatihan dan pengujian model.

Info data setelah encoding:

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
0	40	1	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1
1	58	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	1
2	41	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	1
3	45	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1
4	60	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1

Gambar 4. Hasil Proses *Encoding*

Setelah tahapan pra-pemrosesan selesai, dataset dibagi menjadi data latih dan data uji dengan proporsi 80:20. Pembagian ini dilakukan untuk memisahkan data yang digunakan dalam proses pembelajaran model dari data yang digunakan untuk menguji kemampuan model dalam melakukan klasifikasi terhadap data yang belum pernah dipelajari. Dengan jumlah 251 data, pembagian tersebut menghasilkan sekitar 200 data latih dan 51 data uji. Selanjutnya, data latih digunakan untuk membangun model KNN, *Decision Tree*, dan *Random Forest*, sedangkan data uji digunakan sebagai dasar evaluasi kinerja ketiga model.

PRINSIP : Portal Riset & Inovasi Sistem Perangkat Lunak
Volume 4, Nomor 3, Juli-September 2026
http://doi.org/10.59696/prinsip.v4i3.267
e-ISSN : 2985-802X
p-ISSN : 2987-0534

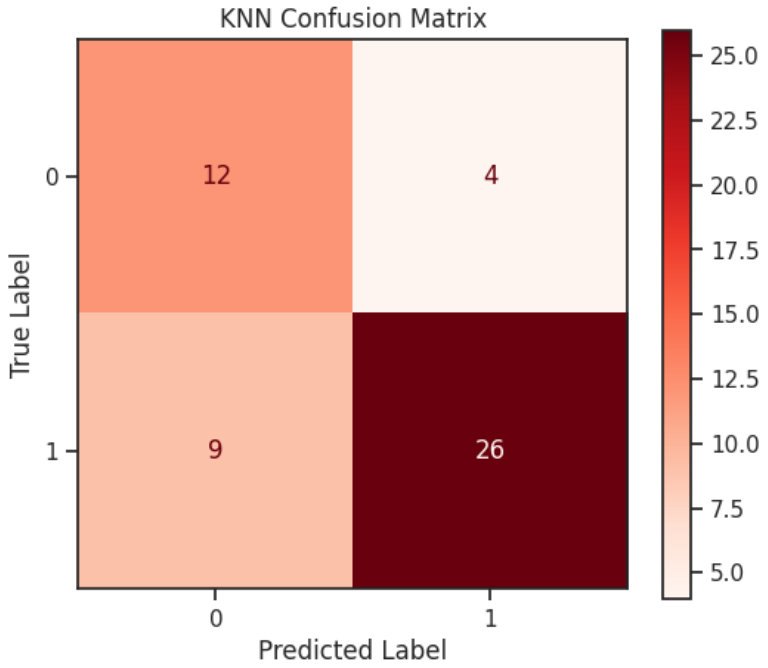
Tabel 2. Pembagian Dataset Menjadi Data Latih dan Data Uji

Pembagian Dataset	Jumlah	Persentase
Data latih	±200	80%
Data uji	±51	20%
Total	**251**	**100%**

Pembagian data dengan proporsi 80:20 selanjutnya diterapkan secara konsisten pada ketiga algoritma untuk memastikan proses perbandingan kinerja dilakukan pada kondisi pengujian yang sama. Dengan pendekatan tersebut, perbedaan hasil evaluasi yang diperoleh dapat digunakan untuk membandingkan kemampuan KNN, *Decision Tree*, dan *Random Forest* dalam melakukan klasifikasi Diabetes Melitus.

Hasil Klasifikasi Menggunakan K-Nearest Neighbors

Model *K-Nearest Neighbors* (KNN) dibangun dengan nilai *k* sebesar 3. Hasil pengujian menunjukkan bahwa model KNN menghasilkan nilai *True Positive* (TP) sebanyak 26, *True Negative* (TN) sebanyak 12, dan *False Positive* (FP) sebanyak 4. Dokumen penelitian mencatat nilai *False Negative* (FN) sebesar 9 pada hasil evaluasi model. Berdasarkan hasil tersebut, KNN mampu mengidentifikasi sebagian besar data positif, tetapi masih terdapat data positif yang diklasifikasikan sebagai negatif.

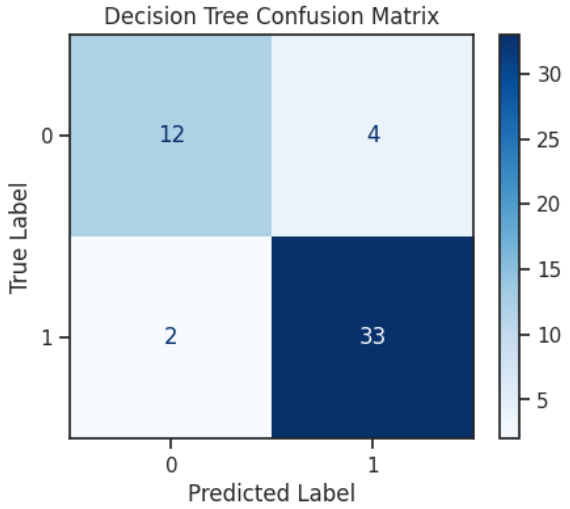


Gambar 5. *Output Confusion Matrix* KNN

Hasil Klasifikasi Menggunakan Decision Tree

Model *Decision Tree* menghasilkan *True Positive* (TP) sebanyak 33, *True Negative* (TN) sebanyak 12, *False Positive* (FP) sebanyak 4, dan *False Negative* (FN) sebanyak 2. Dibandingkan dengan KNN, model ini menghasilkan jumlah kesalahan klasifikasi positif yang lebih rendah, terutama pada kategori *False Negative*. Hal tersebut menunjukkan peningkatan kemampuan model dalam mengenali data yang termasuk kelas positif.

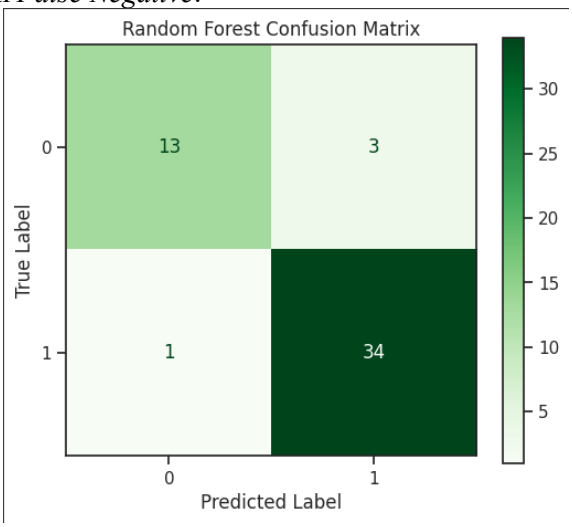
PRINSIP : Portal Riset & Inovasi Sistem Perangkat Lunak
Volume 4, Nomor 3, Juli-September 2026
http://doi.org/10.59696/prinsip.v4i3.267
e-ISSN : 2985-802X
p-ISSN : 2987-0534



Gambar 6. *Output Confusion Matrix Decision Tree*

Hasil Klasifikasi Menggunakan Random Forest

Model *Random Forest* dibangun menggunakan 100 pohon keputusan (*n_estimators* = 100). Hasil *confusion matrix* menunjukkan 34 data *True Positive* (TP), 13 *True Negative* (TN), 3 *False Positive* (FP), dan 1 *False Negative* (FN). Hasil tersebut menunjukkan bahwa *Random Forest* menghasilkan jumlah kesalahan klasifikasi paling rendah dibandingkan KNN dan *Decision Tree*, khususnya pada kesalahan *False Negative*.



Gambar 7. *Output Confusion Matrix Random Forest*

Perbandingan Kinerja Model

Perbandingan ini bertujuan untuk memperoleh gambaran komprehensif mengenai kinerja masing-masing model, tidak hanya berdasarkan nilai akurasi semata, tetapi juga metrik lain seperti *Precision*, *Recall*, dan *F1-Score*, guna menentukan model mana yang memiliki kemampuan prediksi terbaik pada dataset yang digunakan. Perbandingan ketiga algoritma dilakukan berdasarkan hasil evaluasi pada data uji. Hasil penelitian menunjukkan bahwa *Random Forest* menghasilkan kinerja terbaik dengan akurasi 92,16%, diikuti *Decision Tree* sebesar 88,24% dan KNN sebesar 74,51%. Pada metrik AUC, *Random Forest* juga memperoleh nilai tertinggi sebesar 0,9768, sedangkan *Decision Tree* sebesar 0,8464 dan KNN sebesar 0,8393.

Tabel 3. Perbandingan kinerja ketiga algoritma

Perbandingan Kinerja Model	**KNN**	***Decision Tree***	***Random Forest***
Akurasi	74,51%	88,24%	92,16%
AUC	0,8393	0,8464	0,9768

e-ISSN : 2985-802X
p-ISSN : 2987-0534

Hasil perbandingan memperlihatkan adanya perbedaan kinerja yang cukup jelas antaralgoritma. KNN menghasilkan kinerja terendah dengan akurasi 74,51%, sedangkan *Decision Tree* menunjukkan peningkatan akurasi sebesar 13,73 poin persentase. *Random Forest* memberikan hasil paling optimal dengan peningkatan akurasi sebesar 3,92 poin persentase dibandingkan *Decision Tree* dan 17,65 poin persentase dibandingkan KNN. Temuan tersebut memperlihatkan bahwa metode *ensemble* yang digunakan oleh *Random Forest* lebih sesuai untuk menangkap pola kompleks dalam dataset gejala Diabetes Melitus dibandingkan pendekatan berbasis tetangga terdekat maupun satu pohon keputusan.

Pembahasan

Hasil penelitian menunjukkan bahwa *Random Forest* merupakan algoritma dengan performa paling optimal untuk klasifikasi Diabetes Melitus pada dataset yang digunakan. Model ini memperoleh akurasi sebesar 92,16% dan AUC sebesar 0,9768, serta menghasilkan kesalahan klasifikasi paling rendah berdasarkan *confusion matrix*. Keunggulan tersebut menunjukkan bahwa penggabungan sejumlah *Decision Tree* mampu menghasilkan prediksi yang lebih stabil dibandingkan penggunaan satu pohon keputusan. Temuan ini juga sejalan dengan hasil penelitian terdahulu dalam dokumen penelitian yang menunjukkan bahwa algoritma *Random Forest* memiliki potensi untuk digunakan dalam prediksi Diabetes Melitus.

Tabel 4. Ringkasan Hasil Evaluasi dan Interpretasi Model

Model	Hasil Evaluasi	Interpretasi
KNN	Akurasi 74,51% dan AUC 0,8393	Menunjukkan kinerja klasifikasi terendah dibandingkan model lainnya, dengan kemampuan diskriminasi kelas yang relatif lebih rendah.
Decision Tree	Akurasi 88,24% dan AUC 0,8464	Memiliki kinerja lebih baik dibandingkan KNN serta menghasilkan jumlah *False Negative* yang lebih rendah.
Random Forest	Akurasi 92,16% dan AUC 0,9768	Menunjukkan kinerja terbaik berdasarkan akurasi, AUC, dan jumlah kesalahan klasifikasi pada data uji sehingga dipilih sebagai model dengan performa paling optimal.

Dari perspektif penerapan, hasil tersebut menunjukkan bahwa *Random Forest* berpotensi digunakan sebagai model pendukung untuk klasifikasi awal Diabetes Melitus berdasarkan atribut gejala yang tersedia dalam dataset. Meskipun demikian, hasil penelitian tidak dapat diinterpretasikan sebagai pengganti diagnosis klinis karena model dibangun berdasarkan dataset sekunder dan atribut yang tersedia di dalamnya. Penggunaan model pada lingkungan klinis memerlukan validasi lebih lanjut menggunakan dataset yang lebih besar, beragam, dan berasal dari populasi yang relevan. Selain itu, pengujian menggunakan *cross-validation* dan validasi eksternal diperlukan untuk memastikan bahwa performa model dapat digeneralisasikan pada data pasien yang berbeda. Secara keseluruhan, hasil penelitian menjawab tujuan perbandingan tiga algoritma *machine learning* untuk klasifikasi Diabetes Melitus. *Random Forest* menunjukkan performa paling unggul dengan akurasi 92,16% dan AUC 0,9768, sehingga menjadi model terbaik di antara algoritma yang diuji pada dataset penelitian ini. Hasil tersebut memperkuat bahwa pemilihan algoritma perlu mempertimbangkan lebih dari satu metrik evaluasi agar kemampuan model dapat dinilai secara komprehensif, khususnya dalam konteks klasifikasi penyakit yang memiliki konsekuensi penting terhadap kesalahan prediksi.

KESIMPULAN

Penelitian ini berhasil menerapkan dan membandingkan tiga algoritma *Machine Learning*, yaitu *K-Nearest Neighbors* (KNN), *Decision Tree*, dan *Random Forest*, untuk melakukan klasifikasi penyakit Diabetes Melitus berdasarkan 17 atribut gejala pada dataset yang digunakan. Hasil evaluasi menunjukkan bahwa *Random Forest* memberikan kinerja terbaik dengan akurasi sebesar 92,16%, *F1-Score* sebesar 94,44%, dan nilai *Area Under the Curve* (AUC) sebesar 0,9768, lebih tinggi dibandingkan *Decision Tree* yang memperoleh akurasi 88,24% dan AUC 0,8464 serta KNN dengan akurasi 74,51% dan AUC 0,8393. Berdasarkan perbandingan tersebut, *Random Forest* memiliki kemampuan klasifikasi dan diskriminasi kelas positif dan negatif yang paling baik sehingga menjadi model yang paling optimal pada dataset penelitian ini dalam mendukung prediksi Diabetes Melitus.

e-ISSN : 2985-802X
p-ISSN : 2987-0534

Meskipun demikian, hasil penelitian ini masih terbatas pada karakteristik dan ukuran dataset yang digunakan sehingga pengembangan penelitian selanjutnya disarankan menggunakan dataset yang lebih besar dan beragam serta membandingkan lebih banyak algoritma untuk memperoleh model prediksi yang lebih robust dan dapat digeneralisasikan.

REFERENSI

Marlim, Y. N., Suryati, L., & Agustina, N. (2022). Deteksi Dini Penyakit Diabetes Menggunakan Machine Learning Dengan Algoritma Logistic Regression. Jurnal Nasional Teknik Elektro Dan Teknologi Informasi, 11(2).

International Diabetes Federation. (2021). *IDF diabetes atlas* (10th ed.). https://idf.org/news-and-resources/resources/idf-diabetes-atlas-2021/

World Health Organization. (2022). *Diabetes*. https://www.who.int/news-room/fact-sheets/detail/diabetes

Yasin, S. Q. F., Widodo, A. W., & Indriati (2025). Klasifikasi Tingkat Obesitas Berdasarkan Pola Hidup dan Kebiasaan Konsumsi Makanan menggunakan meotde K-Nearest Neighbor. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *9*(3).

Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32 https://doi.org/10.1023/A:1010933404324

Triyono, A., Trianto, R. B., & Arum, D. M. P. (2021). Early Detection Of Diabetes Mellitus Using Random Forest Algorithm. *Julia: Jurnal Ilmu Komputer An Nuur*, *1*(01), 25-31. https://doi.org/10.35720/julia.v1i01.13

Sriyanto, S., & Supriyatna, A. R. (2023). Prediksi Penyakit Diabetes Menggunakan Algoritma Random Forest. *TEKNIKA*, *17*(1), 163-â. https://doi.org/10.5281/zenodo.8051410

Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm: Random forest. In *Lecture notes in computer science* (Vol. 7473, pp. 246–252). Springer. https://doi.org/10.1007/978-3-642-34062-8_32

Bell, J. (2020). *Machine learning: Hands-on for developers and technical professionals* (2nd ed.). Wiley.

Dewi, L. A. (2023). *Klasifikasi Machine Learning Untuk Mendeteksi Penyakit Jantung Dengan Algoritma K-Nn, Decision Tree dan Random Forest* (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).

Ramadhan, A. S. (2022). Decision Tree Algoritma Beserta Contohnya Pada Data Mining. Binus University School of Information Systems. *binus ac. id*.

Sarosa, M., Nailul Muna, S. S. T., Mila Kusumawardani, S. T., Suyono, A., Azis, I. Y. M., & MPd, S. (2022). *Pemrograman Python dalam Contoh dan Penerapan*. Media Nusa Creative (MNC Publishing).

Heryadi, Y., & Wahyuno, T. (2020). *Machine learning: Konsep dan implementasi*. Gava Media.

Alberti, K. G., & Zimmet, P. Z. (1998). Definition, diagnosis and classification of diabetes mellitus and its complications. Part 1: diagnosis and classification of diabetes mellitus provisional report of a WHO consultation. *Diabetic medicine : a journal of the British Diabetic Association*, *15*(7), 539–553. https://doi.org/10.1002/(SICI)1096-9136(199807)15:7<539::AID-DIA668>3.0.CO;2-S

World Health Organization. (2023). *Diabetes*. https://www.who.int/news-room/fact-sheets/detail/diabetes

Perkumpulan Endokrinologi Indonesia. (2021). *Konsensus pengelolaan dan pencegahan diabetes melitus tipe 2 di Indonesia*. PB PERKENI.

Setiawan, B. (2021). *Penerapan Algoritma You Only Look Once (Yolo) Untuk Deteksi Tanaman Miana Berbasis Android* (Doctoral dissertation, Universitas Muhammadiyah Ponorogo). https://eprints.umpo.ac.id/id/eprint/7775

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Perkumpulan Endokrinologi Indonesia. (2021). *Profil Perkumpulan Endokrinologi Indonesia*. https://pbperkeni.or.id/

Id, I. D. (2021). *Machine learning: Teori, studi kasus dan implementasi menggunakan Python* (Vol.

e-ISSN : 2985-802X
p-ISSN : 2987-0534

1). Unri Press.

Rozzi Kesuma Dinata, D., & Novia Hasdyna, H. (2020). Machine learning.

Begum, S., Chakraborty, D., & Sarkar, R. (2016). Data classification using feature selection and kNN machine learning approach. In *Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks* (pp. 811–814). IEEE. https://doi.org/10.1109/CICN.2015.165

Depari, D. H., Widiastiwi, Y., & Santoni, M. M. (2022). Perbandingan Model Decision Tree, Naive Bayes dan Random Forest untuk Prediksi Klasifikasi Penyakit Jantung. *Informatik: Jurnal Ilmu Komputer*, *18*(3), 239-248. https://doi.org/10.52958/iftk.v18i3.4694

Silitonga, Y. R., Munawar, I., & MMSI, M. C. (2019). Analisis Dan Penerapan Datamining Untuk Mendeteksi Berita Palsu (Fake News) Pada Social Media Dengan Memanfaatkan Modul Scikit Learn. *Digilib Esa Unggul*.

Wibisono, S., Hadikurniawati, W., Yulianton, H., Lestariningsih, E., & Cahyono, T. D. (2024). Optimalisasi Model Klasifikasi Diabetes Menggunakan Ensemble Learning Adaboost, Gradient Boosting, dan XGBoost. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 184-193.