Antigen: Jurnal Kesehatan Masyarakat dan Ilmu Gizi
Vol.2, No.3 Agustus 2024
e-ISSN: 3026-6092; p-ISSN: 3026-6084, Hal 39-53
DOI: https://doi.org/10.57213/antigen.v2i3.303
Available online at: https://jurnal.stikeskesosi.ac.id/index.php/Antigen

Prediksi Penyakit Diabetes Mellitus Tipe I dan Tipe II Menggunakan Metode KNN di Klinik Dharma Husada

Richa Nanda Fitria[1*], Wahyu Sugianto[2], Amalia Cemara Nur'aidha[3]
[1-3] Universitas PGRI Yogyakarta, Indonesia
*Email; ichannadafitria@gmal.com**

Abstract*. Diabetes Mellitus (DM) is a metabolic disorder characterized by high blood sugar levels due to insulin deficiency. Factors causing Diabetes Mellitus (DM) are lifestyle which includes diet, lack of exercise, monitoring blood sugar, and medication. Most people do not realize that they have DM and only find out when they experience severe symptoms. To avoid this, the k-Nearest Neighbor (KNN) method can be used to predict the possibility of developing diabetes. The aim of this research is to classify diabetes mellitus using the K-Nearest Neighbor (KNN) method and make people more aware of the risk of disease through healthy lifestyle changes. Data received from the Dharma Husada Clinic is categorized based on researchers' needs, including age, BMI, insulin, skin thickness, glucose, diabetes, genetics, and insulin. This research was carried out in three main steps: dataset input, preprocessing, and evaluation. The first stage is data analysis which begins by entering a dataset to train and test the model, where each data element has certain characteristics (attributes) and classes. Preprocessing steps include training data generation and data cleaning, which includes sanitization, lowercase, normalization, stopwords, stemming, and tokenizing. The final step is evaluating. Evaluation includes building an evaluation model and measuring the level of accuracy, building a predictive model, and saving the model. This research shows that the K-Nearest Neighbor (KNN) method can be used to classify diabetes mellitus (DM), but especially in a small dataset consisting of 245 dates and 8 attributes it is not accurate for patients aged 30 years. . A k value that is too small can cause overfitting, and a k value that is too large can cause underfitting. However, if the amount of data is small, the choice of k can have a large impact.*

Keywords*: Diabetes Mellitus, KNN, Dataset*

Abstrak. Diabetes Mellitus (DM) merupakan adanya kelainan metabolisme yang ditandai dengan kadar gula darah tinggi karena kekurangan insulin. Faktor penyebab terjadinya Diabetes Mellitus (DM) yaitu gaya hidup yang meliputi pola makan, kurangnya olahraga, pemantauan gula darah, dan pengobatan. Kebanyakan orang tidak menyadari bahwa mereka mengidap penyakit DM ini dan baru mengetahui ketika mereka mengalami gejala yang parah. Untuk menghindari hal tersebut, metode k-Nearest Neighbor(KNN) dapat digunakan untuk memprediksi kemungkinan terkena diabetes. Tujuan penelitian ini untuk mengklasifikasikan penyakit diabetes mellitus menggunakan metode K-Nearest Neighboar (KNN) dan membuat masyarakat lebih sadar akan resiko penyakit melalui perubahan gaya hidup yang sehat. Data yang diterima dari Klinik Dharma Husada dikategorikan berdasarkan kebutuhan peneliti, antara lain usia, BMI, insulin, ketebalan kulit, glukosa, diabetes, genetika, dan insulin. Penelitian ini dilakukan dalam tiga langkah utama: input dataset, preprocessing, dan evaluasi. Tahap pertama adalah analisis data yang dimulai dengan memasukkan dataset untuk melatih dan menguji model, di mana setiap elemen data memiliki karakteristik (atribut) dan kelas tertentu. Langkah-langkah preprocessing meliputi pembuatan data pelatihan dan pembersihan data, yang mencakup sanitasi, huruf kecil, normalisasi, stopwords, stemming, dan tokenizing. Langkah terakhir adalah mengevaluasi, Evaluasi meliputi membangun model evaluasi dan mengukur tingkat akurasi, membangun model prediktif, dan menyimpan model. Penelitian ini menunjukkan bahwa metode K-Nearest Neighbor (KNN) dapat digunakan untuk mengklasifikasikan penyakit diabetes melitus (DM), namun terutama pada dataset kecil yang terdiri dari 245 tanggal dan 8 atribut tidak akurat untuk pasien berusia 30 tahun. . Nilai k yang terlalu kecil dapat menyebabkan overfitting, dan nilai k yang terlalu besar dapat menyebabkan underfitting. Namun, jika jumlah datanya kecil, pemilihan k dapat berdampak besar.

Kata kunci: Diabetes Mellitus, K-Nearest Neighbor (KNN), Dataset

1. LATAR BELAKANG

Diabetes Mellitus (DM) merupakan adanya kelainan metabolisme yang ditandai dengan kadar gula darah tinggi karena kekurangan insulin(Hardianto, 2021). Insulin merupakan hormon yang mengatur keseimbangan kadar gula, akibatnya konsentrasi glukosa dalam darah

Received: Juli 12, 2024; Revised: Juli 28, 2024; Accepted: August 04, 2024; Online Available: August 07, 2024;

sehingga terjadi apa yang disebut hiperglikemia(Marzel, 2020). Faktor penyebab terjadinya Diabetes Mellitus (DM) yaitu gaya hidup yang meliputi pola makan, kurangnya olahraga, pemantauan gula darah, dan pengobatan(Irwansyah & Kasim, 2021). Diabetes dikaitkan dengan kerusakan jangka panjang, disfungsi, dan kegagalan beberapa organ tubuh, terutama mata, ginjal, saraf, jantung, dan pembuluh darah(Norma Lalla & Rumatiga, 2022a). Pada tahun 2019, jumlah penderita DM di Indonesia mencapai 10,7 orang. Menjadikannya salah satu negara dengan kategori tertinggi di dunia(Nesyifa & Huriah, 2023).

Menurut Dinas Kesehatan Yogyakarta pada tahun 2019, hasil Riskesdas tahun 2018 menunjukkan bahwa Daerah Istimewa Yogyakarta (DIY) adalah salah satu daerah dengan jumlah penderita Diabetes Mellitus (DM) tertinggi di Indonesia, dengan prevalensi 3,2%. Di dalam DIY, kota Yogyakarta memiliki prevelensi DM tertinggi, yaitu 4,75% yang dua kali lipat dari rata-rata prevelensi DM nasional sebesar 2%(Instituto Nacional de Estadística, 2021)

Kebanyakan orang tidak menyadari bahwa mereka mengidap penyakit DM ini dan baru mengetahui ketika mereka mengalami gejala yang parah. Untuk memprediksi diagnosa Diabetes tersebut, metode k-Nearest Neighbor(KNN) dapat digunakan untuk memprediksi kemungkinan terkena diabetes dan membuat masyarakat lebih sadar akan resiko penyakit melalui perubahan gaya hidup yang sehat(Melinda et al., 2022). Algoritma KNN dapat mengklasifikasikan data berdasarkan kemiripan atau kedekatan data yang diambil dengan data lainnya(Sholeh et al., 2022). Jika diterapkan dengan benar, KNN dapat menjadi alat yang efektif untuk membantu diagnosis dan pengobatan diabetes(Hasanah et al., 2024)

2. KAJIAN TEORITIS

Diabetes Mellitus

Diabetes Mellitus adalah gangguan *metabolic* yang ditandai peningkatan kadar glukosa darah (*Hiperglikemia*) akibat kerusakan pada sekresi insulin dan kerja insulin, kadar glukosa darah setiap hari bervariasi(Norma Lalla & Rumatiga, 2022b). Beberapa penyakit DM juga bersifat simtomatik, dengan gejala penyerta seperti gangguan pendengaran dan atrofi optik(Rochmah, 2019). Diabetes tipe 1 termasuk diabetes yang disebabkan oleh kerusakan sel pankreas akibat proses autoimun atau idiopatik yang menyebabkan berkurangnya atau terhentinya produksi insulin. Diabetes tipe ini biasanya berkembang sebelum usia 20 tahun(Guarango, 2022). DM tipe II paling sering terjadi pada orang lanjut usia, namun kejadiannya meningkat pada anak-anak, remaja, dan dewasa muda. Penyebab DM tipe II erat kaitannya dengan kelebihan berat badan dan obesitas, usia, serta riwayat keluarga. Di antara

faktor makanan, penelitian terbaru menunjukkan hubungan antara tingginya asupan minuman manis dan risikonya(Suyani, 2022)

Algoritma K-Nearest Neighbor (KNN)

K-Nearest Neighbor (K-NN) digunakan untuk mengklasifikasikan objek dengan memeriksa data jarak terdekat. Anda dapat menggunakan jarak untuk menghitung seberapa dekat atau jauh Anda dari data database. Algoritma ini terdiri dari dua tahap yaitu pembelajaran (training) dan klasifikasi atau pengujian (testing)(Silalahi & Simanullang, 2023). Keuntungan dari KNN adalah teknik klasifikasinya yang sangat sederhana, mudah digunakan, memiliki resolusi tinggi (misalnya, tidak ada pemisahan kelas secara linier), dan efisien untuk menghitung area data yang kecil.Hitung data kecil dan miliki beberapa parameter referensi (pengukur jarak dan k)(Pratiwi & Wijayanto, 2019). Salah satu kelemahan algoritma KNN adalah penentuan variabel K. Jika nilai K terlalu besar dan hasil yang digunakan adalah 1 maka hasil klasifikasi akan terlihat berat(Mahalisa & Arminarahmah, 2022).

A. Evaluasi Model

Dalam evaluasi klasifikasi, ada empat kemungkinan berdasarkan hasil klasifikasi data. Jika datanya positif dan diprediksi positif maka dianggap positif benar. Jika data positif diprediksi negatif, maka dianggap negatif palsu. Data negatif dihitung sebagai data negatif sejati jika diprediksi negatif, palsu. Data negatif dihitung, sebagai data negatif sejati jika diprediksi negatif, dan positif palsu jika diprediksi negatif.

Tabel 1. Confusion Matriks

Actual	Prediction	
	Positif	Negatif
Positif	True Positive (TP)	True Negative (TN)
Negatif	False Positive (FP)	False Negative (FN)

a. Precision

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban sistem precision dapat dihitung dengan persamaan **2.1** sebagai berikut :

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \qquad \textbf{(2.1)}$$

b. Recall

Recall adalah salah satu perhitungan keakuratan prediksi yang digunakan sebagai ukuran tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi, recall dapat dihitung melalui persamaan **2.2** sebagai berikut :

$$Recall = \frac{\text{TP}}{TP+FN} \qquad \textbf{(2.2)}$$

c. Accuracy

Accuracy adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual. Jika nilai akurasi tinggi maka sebuah sistem akan semakin bagus dalam melakukan prediksi, akurasi dapat dihitung dengan persamaan **2.3** sebagai berikut ini :

$$Accuracy = \frac{\text{Prediksi data benar}}{\text{Total data}} \quad (2.3)$$

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

TP (True Positif) : Jumlah data positif yang akan diklasifikasikan benar oleh sistem.

TN (True Negatif) : Jumlah data negatif yang diklasifikasikan benar oleh sistem.

FP (False Positif) : Jumlah data positif yang diklasifikasikan salah oleh sistem.

FN (False Negatif) : Jumlah data negatif yang diklasifikasikan salah oleh sistem.

(Suryati et al., 2023)

d. F1-Score

Pembobotan dari perbandingan rata-rata sebuah precision dan recall. Adapun persamaan 2.4 untuk menghitung nilai *f1-score* adalah sebagai berikut :

$$\boldsymbol{F}1 - score = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall+precision}} \textbf{(2.4)}$$

(Mandita & Lintang Ragadanu Arizona, 2023)

3. METODE PENELITIAN

1. Data dan Sumber Data

Karena data yang digunakan dalam penelitian ini merupakan data sekunder yang telah tersedia, maka peneliti tidak terjun langsung ke lapangan untuk mengumpulkan data individu pasien. Data yang diterima dari Klinik Dharma Husada 245 data, dan 8 atribut.

2. Metode Pengolahan Data

Karena data yang digunakan dalam penelitian ini sudah tersedia data sekunder, maka alat yang digunakan untuk menjelaskan penelitian ini adalah sebagai berikut :

1. Data

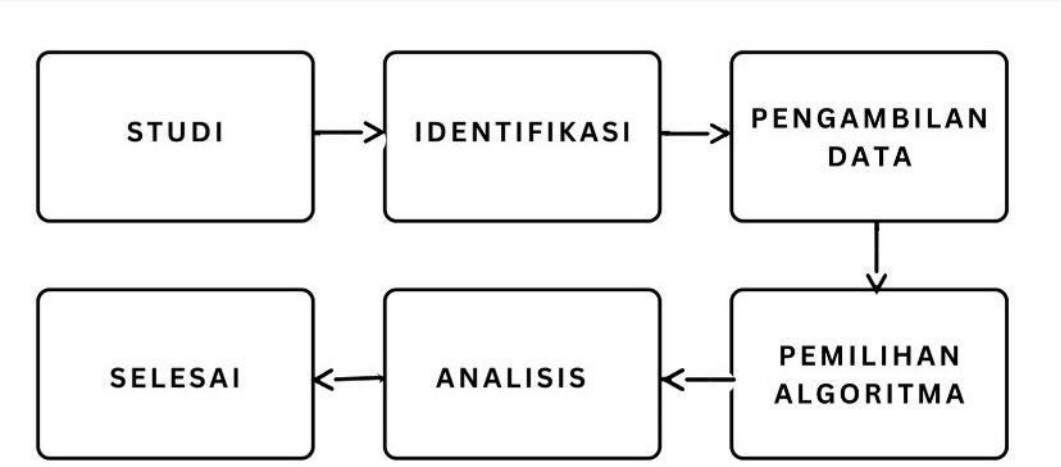
   Data ini merupakan data sekunder pasien diabetes di Klinik Dharma Husada

2. Laptop

   Laptop komputer digunakan untuk menjalankan aplikasi yang diperlukan selama proses penelitian

3. Metode/Design Penelitian

Berikut adalah gambar diagram alur penelitian ini :



Gambar 1. Diagram Alur Penelitian

4. Analisis Data

Penelitian ini dilakukan dalam tiga langkah utama: input dataset, preprocessing, dan evaluasi. Tahap pertama adalah analisis data yang dimulai dengan memasukkan dataset untuk melatih dan menguji model, di mana setiap elemen data memiliki karakteristik (atribut) dan kelas tertentu. Langkah-langkah pada fase ini mencakup memutuskan perpustakaan yang akan digunakan, memuat dataset, melakukan standarisasi data, dan memisahkan data pelatihan serta pengujian. Setelah dataset dimuat, preprocessing dilakukan untuk mempersiapkan data bagi algoritma KNN. Langkah-langkah preprocessing meliputi pembuatan data pelatihan dan pembersihan data, yang mencakup sanitasi, huruf kecil, normalisasi, stopwords, stemming, dan tokenizing. Preprocessing menyesuaikan data dengan format pemrosesannya. Langkah terakhir adalah mengevaluasi hasil model KNN yang telah dilatih untuk mengukur performa model dalam membuat prediksi berdasarkan data pengujian yang belum pernah digunakan sebelumnya. Evaluasi meliputi membangun model evaluasi dan mengukur tingkat akurasi, membangun model prediktif, dan menyimpan model.

4. HASIL DAN PEMBAHASAN

Hasil Penelitian

A. Deskripsi Atribut

Berikut adalah parameter data sekunder yang diperoleh dari rumah sakit :

Tabel 3. Deskripsi Atribut Dataset

	Parameter	Keterangan
1.	Glukosa	Tingkat glukosa darah yang diukur pada pasien
2.	Ketebalan Kulit	Ukuran kelipatan kulit atau ketebalan lipatan lemak tertentu pada tubuh pasien
3.	Insulin	Tingkat hormon insulin yang diukur pada tubuh pasien
4.	BMI	Body Mass Index/Indeks Massa Tubuh
5.	Diabetes	Status atau kondisi pasien terkait dengan diabetes
6.	Nama	Nama Pasi en
7.	Usia	Usia pasien
8.	Diagnosis	Diagnosa yang sudah ditentukan, terkena diabetes atau non diabetes

B. Menginput Dataset

Dalam tahap menginput dataset. ini ada beberapa tahapan yang akan dilakukan, yaitu :

1. Menentukan Library yang Digunakan

Berikut merupakan tahapan import librari (mengimpor pustaka) yang ditampilkan pada *google collab* :

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, precision_s
```

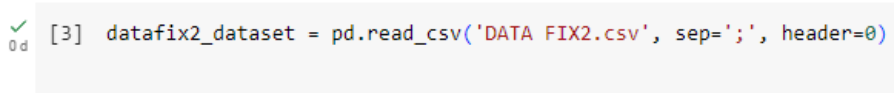
Gambar 2. Impor Library

Jika kode untuk mengimpor library berhasil dieksekusi, pustaka dan modul penting untuk analisis data dan pembangunan model machine learning di Python dengan Visual Studio Code (VsCode) telah dimuat. NumPy dan Pandas membantu mengelola dan memanipulasi data, dengan Pandas memudahkan pembacaan dan pengelolaan data tabular. Modul pre-processing dari scikit-learn, seperti StandardScaler, memfasilitasi standarisasi data untuk meningkatkan kinerja model. Fungsi train_test_split membagi dataset menjadi data latih dan data uji untuk

menguji model. Dengan KneighborsClassifier dari scikit-learn, Anda dapat membangun model K-Nearest Neighbor (K-NN), dan accuracy_score mengukur akurasi model pada data uji. Evaluasi model menggunakan matriks seperti accuracy_score, classification_report, confusion_matrix, precision_score, dan recall_score memberikan pemahaman tentang performa model dalam mengklasifikasikan data. Kombinasi ini memfasilitasi pemrosesan data, pelatihan, dan evaluasi model klasifikasi dengan KNN secara efektif.

2. Load Dataset

Berikut adalah hasil kodingan yang berhasil pada saat meload dataset :



Gambar 3. Load Dataset

Jika kode untuk mengimpor library berhasil dieksekusi, pustaka dan modul penting untuk analisis data dan pembangunan model machine learning di Python dengan Visual Studio Code (VsCode) telah dimuat. NumPy dan Pandas membantu mengelola dan memanipulasi data, dengan Pandas memudahkan pembacaan dan pengelolaan data tabular. Modul pre-processing dari scikit-learn, seperti StandardScaler, memfasilitasi standarisasi data untuk meningkatkan kinerja model. Fungsi train_test_split membagi dataset menjadi data latih dan data uji untuk menguji model. Dengan KneighborsClassifier dari scikit-learn, Anda dapat membangun model K-Nearest Neighbor (K-NN), dan accuracy_score mengukur akurasi model pada data uji. Evaluasi model menggunakan matriks seperti accuracy_score, classification_report, confusion_matrix, precision_score, dan recall_score memberikan pemahaman tentang performa model dalam mengklasifikasikan data. Kombinasi ini memfasilitasi pemrosesan data, pelatihan, dan evaluasi model klasifikasi dengan KNN secara efektif.

Jika proses memuat dataset berhasil, berarti Pandas telah memuat data dari file CSV 'DATA FIX.csv' ke dalam DataFrame Python bernama ‘datafix_dataset’. Variabel ini menyimpan seluruh data, termasuk header dari baris pertama, dengan ‘;’ sebagai delimiter antar kolom dan baris pertama sebagai nama kolom. Setelah dataset dimuat, Anda dapat menjelajahi data menggunakan fungsi Pandas seperti ‘datafix_dataset.head()’ untuk melihat lima baris pertama, ‘datafix_dataset.describe()’ untuk statistik deskriptif, atau ‘datafix_dataset.info()’ untuk informasi tipe data. Hasil dari ‘datafix_dataset.head()’ memberikan gambaran awal tentang struktur data dan memastikan data dimuat dengan benar.

Anda kemudian dapat melanjutkan dengan pembersihan data dan persiapan sebelum analisis atau pembangunan model machine learning.

index	Glukosa	Ketebalan Kulit	Insulin	BMI	Diabetes	Genetik	Usia	Diagnosis
0	101	50	15	36	24.2	0.526	26	0
1	88	66	21	23	24.4	0.342	30	0
2	176	90	34	300	33.7	0.467	28	1
3	150	66	42	342	34.7	0.718	22	0
4	73	50	10	0	23.0	0.248	21	0

Gambar 4. Tampilan Dataset

```
datafix2_dataset['Diagnosis'].value_counts()
```

```
Diagnosis
0    140
1     93
2     12
Name: count, dtype: int64
```

Gambar 5. Hasil dari Kolom Diagnosis

Pada gambar 5, kode `datafix_dataset['Diagnosis'].value_counts()` menghitung dan menampilkan jumlah kemunculan setiap nilai dalam kolom ‘Diagnosis’ dari Dataset. Ini menunjukkan seberapa sering masing-masing kategori muncul, membantu memahami distribusi data dan mempersiapkan analisis atau pembangunan model machine learning, serta membuat visualisasi distribusi kategori dalam dataset.

```
     Glukosa  Ketebalan Kulit  Insulin  BMI  Diabetes Genetik  Usia
0        101              50       15   36      24.2   0.526    26
1         88              66       21   23      24.4   0.342    30
2        176              90       34  300      33.7   0.467    28
3        150              66       42  342      34.7   0.718    22
4         73              50       10    0      23.0   0.248    21
..       ...             ...      ...  ...       ...     ...   ...
240      110              74       29  125      32.4   0.698    27
241      103              60       33  192      24.0   0.966    30
242      138              76        0    0      33.2    0.42    29
243       57              80       37    0      32.8   0.096    21
244      106              64       35  119      30.5     1.4    24

[245 rows x 7 columns]
```

```
print(y)
```

```
0      0
1      0
2      1
3      0
4      0
      ..
240    0
241    2
242    0
243    1
244    0
```

Gambar 6. Pemisahan Data dan Label

Pada **Gambar 6** merupakan hasil dari proses pemisahan data input (fitur) dari label atau terget yang ingin diprediski atau dianalisis dalam konteks pengembangan model

machine learning. Dengan memisahkan data dan label, dapat melanjutkan *pre-processing* data, pembangunan model, dan evaluasi performa model dengan lebih struktur dan efektif. Dan menampilkan bahwa data tersebut memiliki 245 baris dan 8 kolom.

3. Standarisasi Data

StandardScaler mengubah distribusi data sehingga memiliki mean nol dan varians satu, memastikan semua fitur numerik dalam dataset memiliki skala seragam. Ini penting untuk algoritma machine learning seperti k-Nearest Neighbor (KNN), karena standarisasi meningkatkan akurasi dan konsistensi hasil prediksi model.

```
[ ]  scaler = StandardScaler()
```

Gambar 7. Codingan Standarisasi Data

4. Memisahkan Data Latih dan Data Testing

Proses ini penting untuk menguji dan melatih model machine learning dengan menggunakan data yang terpisah, sehingga hasil evaluasi model lebih objektif dan generalisasi model lebih baik terhadap data baru.

```
Ukuran keseluruhan dataset (X): (245, 7)
Ukuran data latih (X_train): (196, 7)
Ukuran data uji (X_test): (49, 7)
```

Gambar 8. Hasil dari Pemisahan Data Latih dan Data Testing

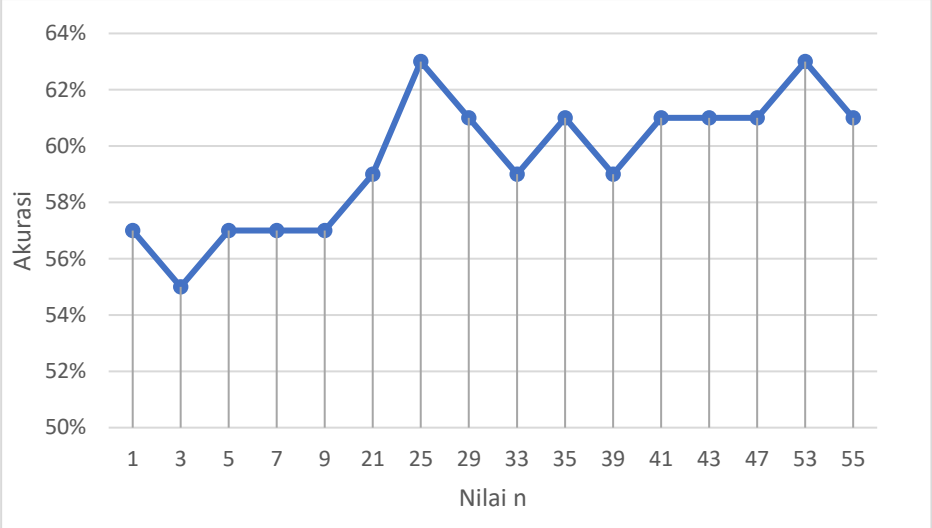
Pada **gambar 8** merupakan informasi tentang ukuran (*shape*) dari beberapa dataset yang digunakan dalam analisis atau pengembangan model machine learning.

5. Membuat Data Latih Menggunakan Algoritma KNN dan Pemilihan Nilai n

Algoritma KNN tidak menghasilkan data latih, melainkan memanfaatkan dataset yang sudah ada. Dataset ini perlu diproses terlebih dahulu, seperti normalisasi atau penanganan nilai hilang, sebelum dibagi menjadi data latih dan data uji. Pemilihan nilai \(k \) (jumlah tetangga terdekat) sangat penting dalam algoritma KNN karena mempengaruhi kinerja model. Nilai \(k \) yang terlalu kecil dapat menyebabkan overfitting, di mana model menjadi sangat sensitif terhadap noise dalam data pelatihan dan bekerja buruk pada data uji. Sebaliknya, nilai \(k \) yang terlalu besar dapat menyebabkan underfitting, di mana model menjadi terlalu umum dan gagal menangkap pola penting dalam data.

6. Membuat Model Evaluasi

Berikut hasil akurasi dari penentuan nilai n :



Gambar 9 Grafik Akurasi

Dilihat **pada gambar 9** bahwa akurasi paling tinggi ditunjukkan pada nilai n = 53. Berikut hasil *confusion matrix* dengan nilai n = 25.

```
Metrics:
Accuracy: 0.6326530612244898
Precision: 0.40952380952380957
Recall: 0.41416040100250623
F1-score: 0.405002405002405

Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.82      0.73        28
           1       0.57      0.42      0.48        19
           2       0.00      0.00      0.00         2

    accuracy                           0.63        49
   macro avg       0.41      0.41      0.41        49
weighted avg       0.60      0.63      0.61        49


Confusion Matrix:
[[23  5  0]
 [11  8  0]
 [ 1  1  0]]
```

Gambar 10. Confusion Matrix nilai n = 25

Gambar 10. menunjukkan evaluasi kinerja model dengan akurasi 63.27%, artinya model membuat prediksi yang benar 63.27% dari waktu. Precision model sebesar 40.95% menunjukkan bahwa dari semua prediksi positif, hanya 40.95% yang benar. Recall sebesar 41.42% mengindikasikan model hanya mengidentifikasi 41.42% dari kasus positif yang sebenarnya. F1-score sebesar 40.50% menggabungkan precision dan recall untuk mengukur

kinerja keseluruhan. Laporan klasifikasi lebih rinci menunjukkan kinerja model pada setiap kelas. Untuk kelas 0, precision 66%, recall 82%, dan F1-score 73%, menunjukkan performa yang cukup baik. Pada kelas 1, precision 57%, recall 42%, dan F1-score 48%, menunjukkan performa yang kurang baik. Untuk kelas 2, precision, recall, dan F1-score semuanya 0%, menunjukkan model gagal mengenali kelas ini. Secara keseluruhan, akurasi model adalah 63%. Rata-rata precision dan recall untuk semua kelas (macro avg) masing-masing adalah 41%. Precision dan recall tertimbang (weighted avg) adalah 60% dan 63%, dengan F1-score tertimbang sebesar 61%. Model berkinerja baik pada kelas 0, tetapi memerlukan perbaikan signifikan pada kelas 1 dan terutama kelas 2.

Tabel 4. Perhitungan Confusion Matriks

	Prediksi		
Aktual	Non Diabetes	Diabetes Tipe I	Diabetes Tipe II
Non Diabetes	23	5	0
Diabetes Tipe I	11	8	0
Diabetes Tipe II	1	1	0

Evaluasi yang menunjukkan bagaimana prediksi model klasifikasi dibandingkan dengan nilai sebenarnya dari data yang diuji. Dalam matriks ini, setiap baris mewakili jumlah instance dalam kelas aktual, sedangkan setiap kolom mewakili jumlah instance dalam kelas prediksi. Untuk kelas 0 sebenarnya, model memprediksi dengan benar sebanyak 23 kali, tetapi salah memprediksi sebagai kelas 1 sebanyak 5 kali. Untuk kelas 1 sebenarnya, model memprediksi dengan benar sebanyak 8 kali, tetapi salah memprediksi sebagai kelas 0 sebanyak 11 kali. Untuk kelas 2 sebenarnya, model gagal memprediksi dengan benar, dengan 1 instance salah diprediksi sebagai kelas 0 dan 1 *instance* salah diprediksi sebagai kelas 1. Secara keseluruhan, model menunjukkan kinerja yang baik pada kelas 0, dengan banyak prediksi yang benar dan beberapa kesalahan. Namun, model mengalami kesulitan signifikan dalam memprediksi kelas 1 dan terutama kelas 2, di mana tidak ada prediksi yang benar untuk kelas 2. Hasil ini menunjukkan perlunya peningkatan model agar lebih efektif

dalam mengenali dan memprediksi *instance* dari kelas 1 dan kelas 2, untuk mencapai kinerja yang lebih seimbang dan akurat.

Pembahasan

Klasifikasi diabetes tipe I dan tipe II didasarkan pada beberapa parameter utama. Diabetes tipe I terjadi karena kerusakan autoimun pada sel beta pankreas, memerlukan terapi insulin seumur hidup, dan biasanya dimulai pada usia muda. Sebaliknya, diabetes tipe II terjadi akibat kombinasi kurangnya produksi insulin dan resistensi insulin, sering berkembang pada usia dewasa dan terkait dengan gaya hidup seperti kelebihan berat badan. Gejala diabetes tipe I muncul tiba-tiba dan berat, sedangkan gejala diabetes tipe II berkembang lebih lambat dan bisa lebih ringan. Diagnosis diabetes tipe I dilakukan melalui tes darah dan pemeriksaan autoantibodi, sedangkan diabetes tipe II didiagnosis dengan tes darah dan penilaian resistensi insulin. Perawatan diabetes tipe I memerlukan terapi insulin, sedangkan diabetes tipe II dapat dikelola dengan perubahan gaya hidup dan obat-obatan oral.

Dalam penelitian ini, klasifikasi diabetes tipe I dan tipe II menggunakan parameter seperti umur, BMI, insulin, ketebalan kulit, glukosa, diabetes, dan faktor genetik. Parameter-parameter ini membantu membedakan antara kedua tipe diabetes dan memberikan pemahaman lebih dalam tentang faktor-faktor yang mempengaruhi masing-masing jenis diabetes. Metode K-Nearest Neighbors (KNN) pada dataset besar memberikan akurasi yang lebih tinggi tetapi memerlukan waktu komputasi lebih lama. Pada dataset kecil, KNN lebih cepat tetapi mungkin kurang akurat. Akurasi 63% yang diperoleh dalam penelitian ini menunjukkan kinerja model yang moderat dan perlu dievaluasi lebih lanjut. Akurasi KNN sangat bergantung pada kualitas data, pemilihan parameter k, dan skala fitur. Penelitian ini menghadapi tantangan signifikan karena menggunakan dataset kecil dengan 245 data dan 8 atribut untuk mengklasifikasikan Diabetes Mellitus (DM) pada pasien usia 15 hingga 30 tahun. Ukuran dataset yang kecil dapat mempengaruhi akurasi dan stabilitas model, sehingga diperlukan penelitian lebih lanjut dengan data yang lebih besar dan lebih banyak atribut untuk meningkatkan akurasi dan keandalan model.

5. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian ini, metode K-Nearest Neighbors (KNN) dapat digunakan untuk mengklasifikasikan penyakit Diabetes Mellitus (DM). Namun, dengan dataset yang terdiri dari 245 data dan 8 atribut, model KNN menunjukkan performa yang kurang memadai.

Hasil evaluasi menunjukkan akurasi sebesar 63%, dengan precision, recall, dan F1-score masing-masing sebesar 40%, 41%, dan 40%. Nilai-nilai ini menunjukkan bahwa meskipun KNN dapat diimplementasikan, akurasinya tidak memadai, terutama untuk pasien berusia 15 hingga 30 tahun.

Ukuran dataset yang kecil dan jumlah atribut yang terbatas berkontribusi pada keterbatasan ini. Model KNN kesulitan membedakan antara pasien yang sehat dan yang menderita DM secara akurat, serta berisiko mengalami overfitting dan kurangnya generalisasi. Oleh karena itu, penelitian lebih lanjut dengan dataset yang lebih besar dan atribut yang lebih banyak diperlukan untuk meningkatkan performa dan akurasi model KNN dalam klasifikasi DM.

B. Saran

Berdasarkan hasil penelitian ini, terdapat beberapa saran untuk meningkatkan efektivitas klasifikasi penyakit Diabetes Mellitus (DM) menggunakan metode K-Nearest Neighbors (KNN). Pertama, mengumpulkan dataset yang lebih besar akan sangat bermanfaat karena data yang lebih banyak memungkinkan model untuk belajar dari lebih banyak contoh dan menangkap pola yang lebih kompleks, sehingga mengurangi risiko overfitting. Selain itu, menambah jumlah atribut yang digunakan dalam model dapat memberikan informasi yang lebih lengkap tentang faktor-faktor yang mempengaruhi diagnosis DM, meningkatkan kemampuan model untuk membuat keputusan yang lebih akurat. Penting juga untuk melakukan pencarian parameter yang lebih mendalam, seperti menentukan nilai k yang optimal, melalui teknik validasi silang atau grid search untuk meningkatkan performa model. Standarisasi atau normalisasi data sebelum penerapan KNN juga harus dipertimbangkan untuk memastikan bahwa semua fitur berkontribusi secara setara dalam perhitungan jarak, yang dapat meningkatkan akurasi model. Selain itu, mempertimbangkan penggunaan metode klasifikasi lain yang mungkin lebih cocok untuk data kecil, seperti pohon keputusan atau Support Vector Machines (SVM), bisa memberikan hasil yang lebih baik. Menggunakan metrik evaluasi tambahan seperti precision, recall, dan F1-score juga dapat memberikan gambaran yang lebih menyeluruh tentang kinerja model dan membantu dalam memahami area yang perlu diperbaiki. Terakhir, memastikan kualitas data yang tinggi dengan membersihkan data dari noise dan outlier serta melakukan preprocessing yang tepat akan meningkatkan performa model secara keseluruhan. Dengan menerapkan saran-saran ini, diharapkan akurasi dan keandalan model KNN dalam klasifikasi DM dapat meningkat secara signifikan.

DAFTAR REFERENSI

Admojo, F. T., & Sulistya, Y. I. (2022). Analisis Performa Algoritma Stochastic Gradient Descent (SGD) Dalam Mengklasifikasi Tahu Berformalin. *Indonesian Journal of Data and Science*, *3*(1), 1–8. https://doi.org/10.56705/ijodas.v3i1.42

Guarango, P. M. (2022). No Title *8.5.2017*, Analisis Pengelolaan Pola Makan Yang Berpengaruh Terhadap Gula Darah Penderita DM, 2003–2005.

Hardianto, D. (2021). Telaah Komprehensif Diabetes Melitus: Klasifikasi, Gejala, Diagnosis, Pencegahan, Dan Pengobatan. *Jurnal Bioteknologi & Biosains Indonesia (JBBI)*, *7*(2), 304–317. https://doi.org/10.29122/jbbi.v7i2.4209

Hasanah, L. U., Natasya, R. P., & Utami, V. D. (2024). Penerapan Algoritma K-Nearest Neighbor (Knn) Untuk Diagnosis Penyakit Diabetes Melitus. *Jurnal Ilmu Komputer Revolusioner*, *8*(1), 86–89.

Irwansyah, I., & Kasim, I. S. (2021). Indentifikasi Keterkaitan Lifestyle Dengan Risiko Diabetes Melitus. *Jurnal Ilmiah Kesehatan Sandi Husada*, *10*(1), 62–69. https://doi.org/10.35816/jiskh.v10i1.511

Mahalisa, G., & Arminarahmah, N. (2022). Diabetes Classification Analysis Using the Euclidean Distance Method Based on the K-Nearest Neighbors Algorithm. *JTKSI (Jurnal Teknologi Komputer Dan Sistem Informasi)*, *5*(3), 178. https://doi.org/10.56327/jtksi.v5i3.1249

Marzel, R. (2020). Terapi pada DM Tipe 1. *Jurnal Penelitian Perawat Profesional*, *3*(1), 51–62. https://doi.org/10.37287/jppp.v3i1.297

Melinda, Khasanah, S., & Susanto, A. (2022). Gambaran Kadar Gula Darah Penderita Diabetes Mellitus Peserta Prolanis di Puskesmas 1 Sumbang Kabupaten Banyumas. *Jurnal Inovasi Penelitian*, *3*(6), 6657–6670.

Nesyifa, N., & Huriah, T. (2023). Studi Kasus Penerapan Senam Kaki DM DAN Edukasi Rokok Dalam Peningkatan Sirkulassi Dan Pengetahuan Klien Diabetes Mellitus Tipe 2 Dan Perokok Aktif. *Nursinng Sciene Journal*, *4*(1), 79–86.

Norma Lalla, N. S., & Rumatiga, J. (2022a). Ketikdakstabilan Kadar Glukosa Darah Pada Pasien Diabetes Melitus Tipe II. *Jurnal Ilmiah Kesehatan Sandi Husada*, 473–479. https://doi.org/10.35816/jiskh.v11i2.816

Norma Lalla, N. S., & Rumatiga, J. (2022b). Ketikdakstabilan Kadar Glukosa Darah Pada Pasien Diabetes Melitus Tipe II. *Jurnal Ilmiah Kesehatan Sandi Husada*, *December*, 473–479. https://doi.org/10.35816/jiskh.v11i2.816

Pratiwi, I. A. A. S., & Wijayanto, A. W. (2019). Klasifikasi Indeks Pembangunan Manusia dengan Metode K-Nearest Neighbor dan Support Vector Machine di Pulau Jawa. *Jurnal Ilmu Komputer*, *15*(1), 8–21. https://ojs.unud.ac.id/index.php/jik/article/download/68565/44248

Rochmah, N. S. M. F. L. (2019). *Tatalaksana Poliuria pada Anak dalam Praktek Sehari-Hari*. 1–113.

Sholeh, M., Andayati, D., & Rachmawati, R. Y. (2022). Data Mining Model Klasifikasi Menggunakan Algoritma K-Nearest Neighbor Dengan Normalisasi Untuk Prediksi Penyakit Diabetes. *TeIKa*, *12*(02), 77–87. https://doi.org/10.36342/teika.v12i02.2911

Sidik, A. D. W. M., Himawan Kusumah, I., Suryana, A., Edwinanto, Artiyasa, M., & Pradiftha Junfithrana, A. (2020). Gambaran Umum Metode Klasifikasi Data Mining. *FIDELITY : Jurnal Teknik Elektro*, *2*(2), 34–38. https://doi.org/10.52005/fidelity.v2i2.111

Silalahi, A. P., & Simanullang, H. G. (2023). Supervised Learning Metode K-Nearest Neighbor Untuk Prediksi Diabetes Pada Wanita. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, *7*(1), 144–149. https://doi.org/10.46880/jmika.vol7no1.pp144-149

Suryati, E., Styawati, & Aldino, A. A. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Dan Sistem Informasi*, *4*(1), 96–106. https://doi.org/10.33365/jtsi.v4i1.2445

Suyani, S. (2022). Faktor-Faktor Yang Berhubungan Dengan Kejadian Bblr. *JKM (Jurnal Kesehatan Masyarakat) Cendekia Utama*, *10*(2), 199. https://doi.org/10.31596/jkm.v10i2.1069