

# PENERAPAN ALGORITMA *K-NEAREST NEIGHBOR* (K-NN) UNTUK KLASIFIKASI PENYAKIT DIABETES MELITUS STUDI KASUS : WARGA DESA JATITENGAH

**Happy Andrian Dwi Fasnua<sup>1)</sup>, Haris Yuana<sup>2)</sup>, M. Taofik Chulkamdi<sup>3)</sup>**

<sup>1, 3)</sup> Program Studi Teknik Informatika, Universitas Islam Balitar

<sup>2)</sup> Program Studi Sistem Komputer, Universitas Islam Balitar

<sup>1, 2, 3)</sup> Jalan Majapahit No. 4, Kec. Sananwetan, Kota Blitar, Jawa Timur, Indonesia, kode pos: 66131  
e-mail: [happyfasnuari@gmail.com](mailto:happyfasnuari@gmail.com)<sup>1)</sup>, [harisyuana2020@gmail.com](mailto:harisyuana2020@gmail.com)<sup>2)</sup>, [chulkamdi@gmail.com](mailto:chulkamdi@gmail.com)<sup>3)</sup>

**Abstrak :** *Diabetes adalah penyakit yang ditandai dengan tingginya kadar gula dalam darah yang menyebabkan penyakit ini sangat berbahaya.. Jika diabetes tidak dikontrol dengan baik akan memicu kematian. Tingkat kematian akibat diabetes melitus tergolong tinggi disebabkan oleh penderita tidak merasakan gejala penyakit diabetes atau tidak memahami ciri-ciri penyakit diabetes. Untuk menentukan seseorang menderita diabetes melitus diperlukan beberapa uji kesehatan agar hasil diagnosa dapat terjamin keasliannya dan proses uji klinis tersebut tentunya memakan waktu yang lama. Berdasarkan permasalahan tersebut, maka dibuat sebuah program untuk klasifikasi penyakit diabetes melitus menggunakan algoritma K-Nearest Neighbor (KNN). Algoritma KNN merupakan metode untuk melakukan klasifikasi objek baru berdasarkan data training yang memiliki jarak terdekat (nearest neighbor) dengan objek tersebut. Penelitian ini menggunakan 8 variabel yaitu mudah haus, berat badan turun meskipun konsumsi makanan teratur, tekanan darah tinggi, terdapat riwayat diabetes dalam keluarga, luka yang sulit sembuh, sering buang air di malam hari, hasil cek gula darah dan usia. Data yang digunakan sebanyak 108 data training dan 27 data testing menghasilkan akurasi 93% pada K=9, presisi 100%, recall 60% dan F1-Score 75%. Dengan tingkat akurasi sebesar 93% maka penelitian ini dinilai telah berhasil menerapkan metode KNN untuk melakukan klasifikasi terhadap penyakit diabetes melitus.*

**Kata Kunci—** *Diabetes melitus, Klasifikasi, K-Nearest Neighbor (KNN)*

**Abstract :** *Diabetes is a disease characterized by high levels of sugar in the blood which causes this disease to be very dangerous. If diabetes is not controlled properly it will lead to death. The death rate due to diabetes mellitus is relatively high because the patient does not feel the symptoms of diabetes or does not understand the characteristics of diabetes. To determine a person suffering from diabetes mellitus, several medical tests are needed so that the diagnostic results can be guaranteed authenticity and the clinical trial process certainly takes a long time. long. Based on these problems, a program for the classification of diabetes mellitus was made using the K-Nearest Neighbor (KNN) algorithm. The KNN algorithm is a method for classifying new objects based on training data that has the closest neighbor to the object. This study uses 8 variables, namely easy thirst, weight loss despite regular food consumption, high blood pressure, there is a history of diabetes in the family, wounds that are difficult to heal, frequent urination at night, results of blood sugar checks and age. The data used are 108 training data and 27 testing data resulting in 93% accuracy at K=9, 100% precision, 60% recall and 75% F1-Score. With an accuracy rate of 93%, this study is considered to have succeeded in applying the KNN method to classify diabetes mellitus.*

**Keywords—** *Diabetes mellitus, Classification, K-Nearest Neighbor (KNN)*

## I. PENDAHULUAN

**D**IABETES Melitus adalah salah satu penyakit yang tergolong dalam penyakit tidak menular. Diabetes juga termasuk penyakit yang disebabkan oleh pankreas yang tidak dapat memproduksi dengan baik seperti hormon yang mengatur gula darah atau glukosa [1]. Adanya penyakit diabetes melitus di Indonesia membuat semua orang perlu waspada dan harus menjaga kesehatan dengan melakukan pola hidup yang sehat, termasuk masyarakat di Desa Jatitengah. Pemerintah Desa Jatitengah bekerja sama dengan Puskesmas Kecamatan Selopuro untuk berupaya mencegah penyakit diabetes melitus sejak dini dengan

sering mengadakan tes kesehatan untuk mendeteksi penyakit yang ada dalam masyarakat. Dalam tes kesehatan tersebut menghasilkan data terkait gejala-gejala yang muncul, seperti: mudah haus, riwayat keluarga dengan penyakit diabetes melitus, turunnya berat badan tanpa sebab yang jelas, sering buang air kecil di malam hari, luka yang sulit untuk sembuh, tekanan darah tinggi dan gula darah tinggi. Dari berbagai gejala-gejala tersebut, perlu dilakukan klasifikasi sesuai dengan penyakit yang diderita agar pengobatan dapat dilakukan secara maksimal.

Klasifikasi yang dilakukan sebelumnya mengacu pada hasil uji klinis yang dilakukan oleh pihak Puskesmas Selopuro. Tentunya membutuhkan waktu yang lama dalam melakukan klasifikasi karena klasifikasi dilakukan satu persatu atau tidak bisa melakukan klasifikasi sekaligus dalam banyak data yang ada. Hal tersebut menjadi masalah bagi Pemerintah Desa Jatitengah karena dengan pengklasifikasian yang lama membuat Pemerintah Desa Jatitengah tidak bisa segera melakukan pencegahan atau pengobatan penyakit diabetes melitus. Penelitian ini dilakukan untuk mendapatkan hasil pengklasifikasian Warga Desa Jatitengah yang terindikasi penyakit diabetes melitus (positif) dan tidak terindikasi penyakit diabetes melitus (negatif). Hasil klasifikasi akan digunakan oleh Pemerintah Desa Jatitengah dalam melakukan pengobatan lebih lanjut bagi warga yang terindikasi penyakit diabetes melitus.

Untuk pengklasifikasiannya dapat dengan dilakukan penerapan *data mining* untuk menemukan dan melakukan pemilahan data. *Data mining* merupakan metode analisis data dengan menambang data untuk mendapatkan data dengan cara yang berbeda dari sebelumnya [2]. Salah satu algoritma dalam *data mining* yang digunakan adalah Algoritma *K-Nearest Neighbor* atau KNN. Algoritma KNN adalah metode untuk melakukan klasifikasi objek baru berdasarkan *data training* yang memiliki jarak terdekat (*nearest neighbor*) dengan objek tersebut.

Pada penelitian sebelumnya, Sahar meneliti tentang “Analisis Perbandingan Metode *K-Nearest Neighbor* dan *Naïve Bayes Classifier* pada *Dataset* Penyakit Jantung”. Penelitian ini melakukan perbandingan antara metode KNN dan *Naïve Bayes* untuk melihat metode yang lebih baik untuk melakukan klasifikasi pada dataset penyakit jantung. Berdasarkan hasil pengujian, tingkat akurasi pada metode *K-Nearest Neighbor* sebesar 67%, presisi 65%, *recall* 73%, dan *f-measure* 96% pada nilai  $K=250$  dengan metode jarak *Manhattan*, sedangkan pada metode *Naïve Bayes Classifier* tingkat akurasi yang didapatkan sebesar 58%, presisi 90%, *recall* 55%, dan *f-measure* 68%. Sehingga dapat disimpulkan bahwa performa metode klasifikasi terbaik pada dataset penyakit jantung yaitu metode KNN [3]. Penelitian lain yang dilakukan oleh Abdul Azis dan Danar Putra yang meneliti tentang “Analisa Perbandingan Algoritma *Euclidean* Dan *Manhattan Distance* Dalam Identifikasi Wajah”, penelitian ini menggunakan metode *Local Binary Patterns* (LBP) dalam ekstraksi fitur, sedangkan untuk klasifikasi membandingkan akurasi *euclidean distance* dan *Manhattan distance* yang digunakan dalam tahap pengujian yang memanfaatkan *data training* dan *data testing*. *Euclidean distance* mendapatkan hasil akurasi tertinggi dengan nilai 100% dan *manhattan distance* mendapatkan hasil akurasi sebesar 85%. Dari penelitian ini dapat disimpulkan bahwa metode perhitungan jarak euclidean dalam metode KNN terbaik pada dataset identifikasi wajah [4].

Berdasarkan penelitian tersebut, peneliti menggunakan penerapan *data mining* menggunakan metode *K-Nearest Neighbor* dan perbandingan perhitungan jarak yang tepat. Peneliti bermaksud menggunakan metode KNN dengan perhitungan jarak *euclidean* untuk memecahkan permasalahan yang telah dijabarkan sebelumnya. Sehingga peneliti bermaksud untuk membuat penelitian dengan judul “Implementasi *Data Mining* Untuk Klasifikasi Penyakit Diabetes Melitus Menggunakan Algoritma *K-Nearest Neighbor* (KNN)” yang bertujuan untuk mengklasifikasikan Penyakit Diabetes Melitus dengan menggunakan KNN. Selain itu, penelitian ini juga berguna sebagai acuan untuk program kesehatan bagi pemerintah desa dengan mudah.

## II. TINJAUAN PUSTAKA

### A. Diabetes Melitus

Diabetes Melitus adalah penyakit yang disebabkan oleh kelebihan zat gula atau glukosa dalam darah. Penyakit diabetes melitus terjadi tubuh manusia tidak dapat menggunakan insulin dengan baik atau bisa terjadi ketika pankreas tidak menghasilkan cukup insulin [1]. Berikut ini adalah ukuran seseorang dikatakan menderita diabetes melitus [5]:

1. Seseorang dikatakan normal jika hasil cek gula darah mencapai  $< 140$  mg/dl
2. Jika hasil cek gula darah mencapai  $140 - < 200$  mg/dl, maka disebut sebagai Toleransi Glukosa Terganggu (TGT).
3. Seseorang dikatakan diabetes melitus jika hasil cek gula darah mencapai  $> 200$  mg/dl

### B. Klasifikasi

Klasifikasi adalah proses kategorisasi data dengan memasukkan data tersebut ke dalam kategori tertentu dari kategori yang tersedia [2]. Teknik dalam klasifikasi ialah melakukan kategorisasi *data training* untuk membuat suatu model untuk melakukan kategorisasi pada *data testing* yang tersedia.

### C. Data Mining

*Data Mining* berasal dari dua kata bahasa inggris yaitu “*data*” yang berarti data dan “*mining*” yang berarti menambang. Sehingga *data mining* dapat diartikan sebagai proses penambangan data. *Data mining* merupakan metode untuk mendapatkan inti dari suatu ilmu atau pengetahuan [6]. *Data mining* menghasilkan teknik pengenalan pola data yang dapat memberikan perbedaan hasil dari data lain, sehingga dapat digunakan di masa yang akan datang [7].

### D. Algoritma *K-Nearest Neighbor* (KNN)

*K-Nearest Neighbor* (KNN) adalah metode klasifikasi terhadap objek baru berdasarkan *data training* yang memiliki jarak tetangga terdekat (*nearest neighbor*) dengan objek baru tersebut [8]. Dekat atau jauhnya *neighbor* biasanya dihitung berdasarkan jarak *Euclidean*. Berikut ini adalah langkah-langkah algoritma KNN:

1. Menentukan nilai K. Nilai K dapat dihitung menggunakan persamaan 1 berikut ini :

$$k = \sqrt{N} \quad (1)$$

N merupakan banyaknya sampel pada *data training*

2. Melakukan perhitungan nilai jarak (*euclidean distance*) terhadap masing-masing objek data yang diberikan. Rumus untuk menghitung *euclidean distance* dapat dilihat pada persamaan 2.

$$d_i = \sqrt{(x_{ki} - x_{kj})^2 + (x_{ki} - x_{kj})^2 + \dots + (x_{ki} - x_{kj})^2} \quad (2)$$

Keterangan :

$d_i$  = jarak *euclidean*

$x_{ki}$  = *data training* ke-1

$x_{kj}$  = *data testing* ke-1

3. Melakukan pengelompokkan data sesuai dengan perhitungan jarak (*Euclidean distance*)
4. Melakukan pengelompokkan data sesuai dengan nilai tetangga terdekat (*nearest neighbor*) atau berdasarkan data yang mempunyai jarak *Euclidean* terkecil.
5. Memilih nilai mayoritas dari tetangga terdekat sebagai hasil klasifikasi.

### E. Bahasa Pemrograman Python

Dalam *website docs.python.org*, Python adalah bahasa pemrograman yang dapat digunakan untuk analisis data, mudah digunakan dan berorientasi objek. Python dapat digunakan dalam beberapa sistem operasi termasuk Linux dan macOS, dan Windows.

### F. Confusion Matrix

Confusion Matrix mempresentasikan prediksi dengan membandingkan nilai asli dengan nilai hasil prediksi model untuk menghasilkan penilaian seperti berikut ini [9]:

#### 1. Akurasi

Akurasi merupakan ketepatan antara nilai prediksi dengan nilai aktual. Rumus untuk menghitung nilai akurasi dapat dilihat pada persamaan 3.

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3)$$

#### 2. Presisi

Presisi merupakan tingkat keberhasilan model dalam memberikan jawaban dengan tepat kepada pengguna. Rumus untuk menghitung nilai presisi dapat dilihat pada persamaan 4.

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (4)$$

#### 3. Recall

Recall merupakan tingkat keberhasilan model dalam menemukan kembali informasi dengan benar. Rumus untuk menghitung nilai recall dapat dilihat pada persamaan 5.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

#### 4. F1-Score atau F-Measure.

F1-Score merupakan hasil perbandingan antara nilai presisi dan recall. Rumus untuk menghitung nilai f1-Score dapat dilihat pada persamaan 6.

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

Keterangan:

TP : True Positive

TN : True Negative

FP : False Positive

FN : False Negative

## III. METODE PENELITIAN

### A. Tahapan Awal

Penelitian ini dilakukan di Desa Jatitengah Kecamatan Selopuro Kabupaten Blitar. Penelitian ini dimulai dengan mengidentifikasi masalah yang ada kemudian menyusun rumusan masalah, menentukan tema dan judul penelitian, melakukan studi literatur dan melakukan wawancara serta observasi ke tempat penelitian.

### B. Tahap Pengumpulan Data

Dalam tahap pengumpulan data, peneliti menggunakan 3 teknik dalam proses pengumpulan data yaitu wawancara, observasi dan studi literatur. Wawancara digunakan untuk mendapatkan informasi lebih lanjut terkait dengan penyakit diabetes melitus. Observasi digunakan untuk mendapatkan *dataset* penderita penyakit diabetes melitus dan gejala-gejala yang menyertai. Sedangkan studi literatur digunakan untuk

mendapatkan teori serta materi yang mendukung terkait penyakit diabetes melitus dan Algoritma *K-Nearest Neighbor* (KNN).

### C. Tahapan Data Mining

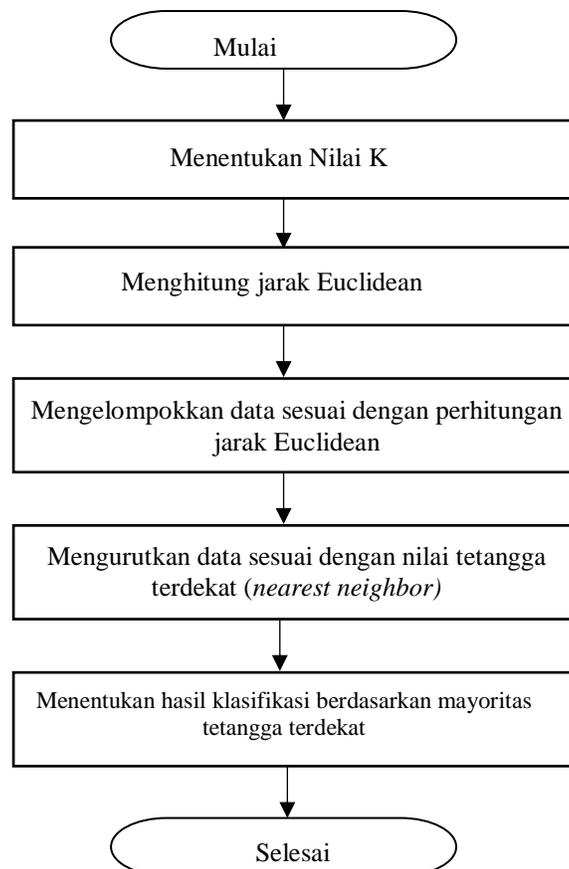
Dalam penelitian ini, tahapan *data mining* yang dilakukan dalam pengambilan data yaitu proses pembersihan data yang digunakan untuk menghilangkan data yang tidak diperlukan seperti data duplikat dan data yang tidak lengkap, tahap integrasi data untuk menggabungkan berbagai sumber data untuk menghasilkan data yang benar dan terhindar dari kesalahan, tahap transformasi data untuk mengubah data ke bentuk yang sesuai dengan *mining* yang dipilih. dan tahap *data mining* yaitu memilih salah satu dari 7 teknik *data mining* yaitu Algoritma KNN.

### D. Teknik Data Mining

Teknik *data mining* yang digunakan dalam penelitian ini adalah Algoritma *K-Nearest Neighbor* (KNN). Algoritma KNN merupakan metode untuk melakukan klasifikasi objek baru berdasarkan *data training* yang memiliki jarak terdekat (*nearest neighbor*) dengan objek tersebut. Berikut ini adalah langkah-langkah dalam melakukan klasifikasi menggunakan Algoritma KNN:

1. Menentukan nilai K
2. Melakukan perhitungan nilai jarak (*euclidean distance*)
3. Melakukan pengelompokan data sesuai dengan perhitungan jarak (*euclidean distance*)
4. Melakukan pengurutan data sesuai dengan nilai tetangga terdekat (*nearest neighbor*) atau berdasarkan data yang mempunyai jarak *euclidean* terkecil.
5. Memilih nilai mayoritas dari tetangga terdekat sebagai hasil klasifikasi.

Alur diagram Algoritma KNN dapat dilihat dalam Gambar 1.



Gambar 1. Alur Diagram Metode KNN

### E. Pengujian Algoritma

Pengujian Algoritma *K-Nearest Neighbor* (KNN) dilakukan untuk mengetahui tingkat keberhasilan algoritma KNN dalam melakukan klasifikasi terhadap penyakit diabetes melitus. Pengujian algoritma KNN pada penelitian ini menggunakan *confusion matrix* yang diproses dengan bahasa pemrograman *python*.

## IV. HASIL DAN PEMBAHASAN

### A. Hasil

Pada penelitian ini, peneliti menerapkan *data mining* untuk melakukan klasifikasi penyakit diabetes melitus menggunakan algoritma *K-Nearest Neighbor* (KNN) berdasarkan pada 8 gejala, yakni: usia, mudah haus (G1), berat badan turun meskipun konsumsi makanan teratur (G2), tekanan darah tinggi (G3), memiliki riwayat penyakit diabetes (G4), luka yang sulit sembuh (G5), sering buang air di malam hari (G6) dan hasil cek gula darah (G7). Dalam proses *data cleansing*, data duplikat dan data yang tidak konsisten dihilangkan sehingga penulis berhasil mendapatkan data dengan total 135 data, dengan rincian yaitu 81 *data testing* dan 54 *data training*.

*Data testing* dan *data training* yang telah dikumpulkan dilakukan proses normalisasi data dengan cara menghitung nilai maksimal dan minimal dari *data testing* dan *data training*, normalisasi data dilakukan di setiap gejala yang ada. Normalisasi data adalah proses penskalaan nilai agar data memiliki jarak yang dekat antara data satu dengan yang lainnya [10].

Tabel 1. *Data Training* Sebelum Normalisasi

No.	Nama	Usia	G1	G2	G3	G4	G5	G6	G7	Ket.
1.	Mdyh	46	0.4	0.5	0.6	0	0.4	0.5	325	Positif
2.	Byn	55	0	0.5	0.7	0.8	0.4	0	253	Positif
3.	Krm	56	0.5	0.5	0.7	0.8	0	0.5	288	Positif
4.	Msrnd	60	0	0.5	0	0.8	0.4	0.5	342	Positif
5.	Fzi	58	0	0.5	0.7	0.8	0.4	0.5	324	Positif
...	...	...	...	...	...	...	...	...	...	...
81.	Fzl	31	0	0	0.6	0	0	0	133	Negatif

Tabel 2. *Data Testing* Sebelum Normalisasi

No.	Nama	Usia	G1	G2	G3	G4	G5	G6	G7	Ket.
1.	Fqrtl	30	0	0.3	0	0	0.4	0	69	?
2.	Rtn	32	0.2	0	0	0.8	0	0	106	?
3.	Ktn	60	0	0.6	0.7	0.8	0	0.5	283	?
4.	Hst	39	0	0.3	0	0.8	0	0	93	?
5.	Mji	37	0.2	0.3	0	0	0	0.5	91	?
...	...	...	...	...	...	...	...	...	...	...
54.	Mrth	51	0	0.6	0.6	0	0	0.5	225	?

Tabel 3. *Data Training* Sesudah Normalisasi

No.	Nama	Usia	G1	G2	G3	G4	G5	G6	G7
1.	Mdyh	0.533333	0.8	0.833333	0.857143	0	1	1	0.681818
2.	Byn	0.833333	0	0.833333	1	1	1	0	0.477273
3.	Krm	0.866667	1	0.833333	1	1	0	1	0.576705
4.	Msrnd	1	0	0.833333	0	1	1	1	0.730114
5.	Fzi	0.933333	0	0.833333	1	1	1	1	0.678977
...	...	...	...	...	...	...	...	...	...
54.	Fzi	0	0	0.6	0	0	0	133	31

Tabel 4. *Data Testing* Sesudah Normalisasi

No.	Nama	Usia	G1	G2	G3	G4	G5	G6	G7
1.	Fqrtl	0	0	0.5	0	0	1	0	0
2.	Rtn	0.066667	0.4	0	0	1	0	0	0.172897
3.	Ktn	1	0	1	1	1	0	1	1
4.	Hst	0.3	0	0.5	0	1	0	0	0.11215
5.	Mji	0.233333	0.4	0.5	0	0	0	1	0.102804
...	...	...	...	...	...	...	...	...	...
54.	Mrth	0.7	0	1	0.857143	0	0	1	0.728972

Berikut ini adalah tahap pengolahan data menggunakan Algoritma KNN dengan data yang telah siap diolah:

1. *Menentukan nilai K*

Nilai K dapat dicari menggunakan rumus perhitungan akar kuadrat dari *data training*. Karena jumlah *data training* sebanyak 81 data maka nilai K ditentukan sebanyak 9 yang dapat dilihat pada Gambar 2.

$$K = \sqrt{81} = 9$$

Gambar 2. Penentuan Nilai K

2. *Melakukan perhitungan nilai jarak Euclidean pada setiap objek data*

Perhitungan jarak dilakukan dengan menghitung akar kuadrat dari pengurangan *data training* dan *data testing* yang dijumlahkan sebanyak *data training* yang ada.

Tabel 5. Perhitungan Jarak Menggunakan *Euclidean Distance*

Data Ke-	1	2	3	4	5	6	...	54
1.	1.798645	2.250759	1.734129	2.205623	1.559187	2.237806	...	1.302989
2.	1.74165	1.880124	1.52604	1.590242	2.184339	1.87125	...	1.768987
3.	2.488938	1.964053	1.106677	1.909983	1.759774	1.952469	...	1.448844
4.	1.908973	2.008991	1.44935	1.727134	1.803443	1.992997	...	1.688927
5.	2.107898	2.20492	1.065494	1.957937	2.023138	2.191775	...	1.450907
...	...	...	...	...	...	...	...	...
81.	1.415768	1.377367	2.168114	1.434012	1.478452	1.380315	...	1.672013

3. *Melakukan pengelompokkan data sesuai dengan perhitungan jarak (Euclidean distance).*

Pengelompokkan data dilakukan dengan mengelompokkan data dengan 9 tetangga terdekat (*nearest neighbor*) berdasarkan hasil perhitungan jarak Euclidean.

Tabel 6. Pengelompokkan data berdasarkan perhitungan *euclidean distance*

Data Ke-	1	2	3	4	5	6	...	54
1.	Negatif	Negatif	Positif	Negatif	Negatif	Negatif	...	Positif
2.	Negatif	Negatif	Positif	Negatif	Negatif	Negatif	...	Positif
3.	Negatif	Negatif	Positif	Negatif	Negatif	Negatif	...	Positif
4.	Negatif	Negatif	Positif	Negatif	Negatif	Negatif	...	Positif
5.	Negatif	Negatif	Positif	Negatif	Negatif	Negatif	...	Positif
...	...	...	...	...	...	...	...	...
81.	Negatif	Negatif	Positif	Negatif	Negatif	Negatif	...	Negatif

4. *Melakukan pengurutan data*

Dalam proses pengurutan, data akan diurutkan sesuai dengan nilai tetangga terdekat (*nearest neighbor*) atau berdasarkan data yang mempunyai jarak *Euclidean* terkecil, seperti yang ditunjukkan pada Tabel 7.

Tabel 7. Pengurutan data berdasarkan nilai tetangga terdekat

Data Ke-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	... 54
Positif	0	0	9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	... 7
Negatif	9	9	0	9	9	9	9	9	9	9	9	9	9	9	8	7	9	9	... 2

5. Melakukan pemilahan data yang sering muncul dari tetangga terdekat sebagai hasil klasifikasi.

Proses terakhir dalam perhitungan menggunakan algoritma KNN adalah memilih nilai yang sering muncul dari mayoritas tetangga terdekat sebagai hasil klasifikasi. Dari data testing sebanyak 54 data didapatkan hasil klasifikasi sebanyak 4 orang “Positif” diabetes melitus dan 50 orang “Negatif” diabetes melitus.

Tabel 8. Hasil Klasifikasi.

Data Ke-	3	34	39	54
Positif	9	6	9	7
Negatif	0	3	0	2

Setelah data diolah menggunakan Algoritma KNN, tahap selanjutnya adalah melakukan pengujian algoritma menggunakan *confusion matrix* menggunakan bahasa pemrograman *python*. Berdasarkan perhitungan *confusion matrix* yang telah dilakukan, di dapatkan hasil akurasi sebesar 0.93 atau 93%, Presisi sebesar 1.00 atau 100%, *recall* sebesar 0.6 atau 60% dan *f1-Score* sebesar 0.75 atau 75%. Berikut ini tabel *confusion matrix* untuk membuktikan perhitungan pada Gambar 3.

	precision	recall	f1-score	support
NEGATIF	0.92	1.00	0.96	22
POSITIF	1.00	0.60	0.75	5
accuracy			0.93	27

Gambar 3. Hasil Perhitungan Akurasi, Presisi, Recall dan F1-Score

Predicted	NEGATIF	POSITIF	All
Actual			
NEGATIF	22	0	22
POSITIF	2	3	5
All	24	3	27

Gambar 4. Tabel Confusion Matrix

Berikut ini perhitungan manual *confusion matrix* berdasarkan Gambar 4[11] :

1. Akurasi

$$Akurasi = \frac{(TP + TN)}{(TP + FP + FN + TN)} = \frac{(3 + 22)}{(3 + 0 + 2 + 22)} = 0.93 * 100\% = 93\%$$

2. Presisi

$$Presisi = \frac{TP}{TP + FP} = \frac{3}{3 + 0} = 1.0 * 100\% = 100\%$$

3. Recall

$$Recall = \frac{TP}{TP + FN} = \frac{3}{3 + 2} = 0.60 * 100\% = 60\%$$

#### 4. *F1-Score* atau *F-Measure*

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} = \frac{2 \times 1.0 \times 0.60}{1.0 + 0.60} = 0.75 * 100\% = 75\%$$

Berdasarkan dari perhitungan di atas dapat dijelaskan bahwa akurasi sebesar 93% berarti model yang dibangun memiliki ketepatan antara nilai prediksi dengan nilai aktual sebesar 93%. Untuk nilai presisi sebesar 100% menandakan bahwa model yang dibuat dapat memberikan jawaban kepada pengguna dengan tepat. Nilai *Recall* sebesar 60% berarti tingkat model yang dibuat berhasil menemukan kembali informasi dengan benar sebesar 60%. Sedangkan nilai *F1-Score* sebesar 75% mengindikasikan bahwa model yang dibuat memiliki presisi dan *recall* yang baik karena nilai terbaik *F1-Score* adalah 1.0 (100 %) dan nilai terburuknya adalah 0.

#### B. Pembahasan

Pada penelitian ini, peneliti menemukan 2 hal yang dapat mempengaruhi hasil dari Algoritma KNN antara lain:

##### 1. Pembagian *data training* dan *data testing*

Nilai akurasi Algoritma KNN dapat berubah sesuai dengan struktur *data training* dan *data testing* [12]. Oleh karena itu, perlu membagi *data training* dan *data testing* dengan tepat. Pada penelitian ini, peneliti mencoba membagi *data testing* sebanyak 25% dari jumlah data sebanyak 153 mendapatkan hasil akurasi sebesar 91%. Sedangkan jika pembagian *data testing* sebanyak 20% mendapatkan hasil akurasi sebesar 93%. Oleh karena itu, peneliti membagi *data testing* sebanyak 20% agar mendapatkan nilai akurasi yang tinggi yaitu sebesar 93%.

##### 2. Penentuan nilai K

Data merupakan hal terpenting dalam menentukan nilai K yang terbaik [13], karena nilai K juga dapat mempengaruhi nilai akurasi Algoritma KNN. Berdasarkan hasil perhitungan *confusion matrix* yang dihitung menggunakan bahasa pemrograman *python*, peneliti melakukan pengujian terhadap 101 *data training* dan 34 *data testing*. Model ini mendapatkan nilai K terbaik dengan rata-rata *error* sebesar 0.07 pada K=4 sampai dengan K=27, dengan menentukan nilai K=9 menghasilkan nilai akurasi sebesar 93%. Dengan K=28, model mendapatkan nilai akurasi sebesar 89%, sedangkan jika K=35 nilai akurasi model sebesar 85%. Dengan demikian peneliti menentukan nilai K=9 agar hasil akurasi model mendapatkan hasil yang tinggi.

Dari percobaan yang telah dilakukan untuk memperbaiki hasil akurasi Algoritma KNN dapat ditarik kesimpulan bahwa model KNN terbaik diperoleh dari percobaan dengan membagi *data testing* sebanyak 20% menggunakan nilai K=9.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Penerapan Algoritma *K-Nearest Neighbor* (KNN) dalam menentukan klasifikasi penyakit diabetes melitus adalah melakukan pengolahan data menggunakan algoritma KNN menggunakan 8 variabel gejala seperti: usia, mudah haus, berat badan turun meskipun konsumsi makanan teratur, tekanan darah tinggi, memiliki riwayat penyakit diabetes, luka yang sulit sembuh, sering buang air di malam hari dan hasil cek gula darah. Penelitian ini menggunakan 81 *data training* dan 54 *data testing* yang sudah mengalami proses pembersihan data kemudian dilakukan proses normalisasi serta perhitungan jarak tetangga terdekat menggunakan metode *Euclidean distance* dengan nilai K=9 mendapatkan hasil klasifikasi 4 orang positif dan 50 orang negatif diabetes melitus. Pengujian Algoritma *K-Nearest Neighbor* (KNN) dalam menentukan klasifikasi penyakit diabetes melitus menggunakan *confusion matrix* mendapatkan hasil akurasi yang baik sebesar 93%, presisi sebesar 100%, *Recall* sebesar 60% dan *F1-Score* sebesar 75%. Dengan tingkat akurasi sebesar 93% maka penelitian ini dinilai telah berhasil menerapkan metode KNN untuk melakukan klasifikasi penyakit diabetes melitus.

## B. Saran

Dalam penelitian ini perlu adanya perbaikan untuk mendapatkan hasil yang baik dengan cara memperbanyak *dataset* yang digunakan agar algoritma KNN dapat bekerja lebih maksimal. Dapat juga dengan melakukan pengujian algoritma KNN menggunakan metode pengujian yang lain, seperti metode *cross validation* untuk menunjukkan validasi akurasi yang lebih terpercaya. Selain itu, dapat pula dikembangkan menjadi suatu aplikasi android sehingga dapat bermanfaat bagi yang membutuhkan. Dapat juga dilakukan dengan menggunakan metode *data mining* yang lain seperti: algoritma C4.5, *K-Means Clustering*, *Naïve Bayes* untuk mendapatkan perbandingan hasil klasifikasi Algoritma *K-Nearest Neighbor*.

## DAFTAR PUSTAKA

- [1] Khairani, “Hari Diabetes Sedunia Tahun 2018,” *Pus. Data dan Inf. Kementrian Kesehat. RI*, pp. 1–8, 2019.
- [2] D. P. Utomo and M. Mesran, “Analisis Komparasi Metode Klasifikasi *Data Mining* dan Reduksi Atribut Pada Data Set Penyakit Jantung,” *J. Media Inform. Budidarma*, vol. 4, no. 2, p. 437, 2020, doi: 10.30865/mib.v4i2.2080.
- [3] S. Sahar, “Analisis Perbandingan Metode *K-Nearest Neighbor* dan *Naïve Bayes Clasiffier* Pada Dataset Penyakit Jantung,” *Indones. J. Data Sci.*, vol. 1, no. 3, pp. 79–86, 2020, doi: 10.33096/ijodas.v1i3.20.
- [4] Abdul Azis, Danar Putra Pamungkas, and Ahmad Bagus Setiawan, “Analisa Perbandingan Algoritma *Euclidean* Dan *Manhattan Distance* Dalam Identifikasi Wajah,” *Semin. Nas. Inov. Teknol.*, pp. 219–224, 2021.
- [5] D. Garnita, “Faktor Risiko Diabetes Melitus Di Indonesia,” *Fkm Ui*, p. 118, 2012.
- [6] Y. Yahya and W. Puspita Hidayanti, “Penerapan Algoritma *K-Nearest Neighbor* Untuk Klasifikasi Efektivitas Penjualan Vape (Rokok Elektrik) pada ‘Lombok Vape On,’” *Infotek J. Inform. dan Teknol.*, vol. 3, no. 2, pp. 104–114, 2020, doi: 10.29408/jit.v3i2.2279.
- [7] J. Leskovec and J. D. Ullman, “Mining of Massive Datasets,” 2014.
- [8] M. M. Baharuddin, H. Azis, and T. Hasanuddin, “Analisis Performa Metode *K-Nearest Neighbor* Untuk Identifikasi Jenis Kaca,” *Ilk. J. Ilm.*, vol. 11, no. 3, pp. 269–274, 2019, doi: 10.33096/ilkom.v11i3.489.269-274.
- [9] I. Pratiwi, “Analisis Performa Metode *K- Nearest Neighbor* ( KNN ) dan *Crossvalidation* pada Data Penyakit Cardiovascular,” vol. 2, no. 1, pp. 21–28, 2021.
- [10] P. E. Amilia R., “Klasifikasi Diagnosa Penyakit Demam Berdarah Dengue Pada Anak Menggunakan Metode *K-Nearest Neighbor* Studi Kasus Rumah Sakit Pku Muhammadiyah Ujung Pangkah Gresik,” *Pap. Knowl. . Towar. a Media Hist. Doc.*, vol. 2, no. 2, pp. 1–10, 2014.
- [11] R. Wahyudi, M. Orisa, and N. Vendyansyah, “Penerapan Algoritma *K-Nearest Neighbors* Pada Klasifikasi,” *J. Mhs. Tek. Inform.*, vol. 5, no. 2, 2021.
- [12] W. Musu, A. Ibrahim, and Heriadi, “Pengaruh Komposisi *Data Training* dan *Testing* terhadap Akurasi Algoritma C4 . 5,” *Pros. Semin. Ilm. Sist. Inf. Dan Teknol. Inf.*, vol. X, no. 1, pp. 186–195, 2021.
- [13] I. A. A. Angreni, S. A. Adisasmita, and M. I. Ramli, “Terhadap Tingkat Akurasi Identifikasi Kerusakan Jalan,” vol. 7, no. 2, pp. 63–70, 2018.