Journal of Information Technology and Computer Science (INTECOMS)
Volume 7 Nomor 4, Tahun 2024
e-ISSN : 2614-1574
p-ISSN : 2621-3249

PENANGANAN OUTLIER PADA METODE ALGORITMA K- NEAREST NEIGHBORS (KNN) DENGAN METODE KERNEL DENSITY ESTIMATION PADA KASUS PENYAKIT DIABETES

HANDLING OUTLIERS IN THE K-NEAREST NEIGHBORS (KNN) ALGORITHM USING KERNEL DENSITY ESTIMATION IN DIABETES CASES

Adam Razaki[1], Yulison Herry Chrisnanto[2], Melina[3]
[1,2,3]Program Studi Informatika, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani, Cimahi
adamrazaki20@if.unjani.ac.id[1]

ABSTRACT

Diabetes is a global health challenge that requires early detection. The presence of outliers in the data can affect the accuracy of diabetes classification models. This research is an experimental quantitative study that aims to improve the accuracy of diabetes classification by handling outliers using a combination of K-Nearest Neighbors (KNN) and Kernel Density Estimation (KDE) algorithms. The research subject used 2,000 samples of diabetes data sourced from the Kaggle web. The data is processed through preprocessing, transformation, normalization, outlier detection with KDE, outlier imputation, KNN classification, and evaluation using confusion matrix. The results of this study show that the use of KDE and outlier imputation can improve model performance on all evaluation metrics. The best model was obtained in KNN with K=3 and 90%:10% dataset ratio, increasing the accuracy from 90% to 92%. %. This research has the potential to improve the accuracy of early detection of diabetes which has implications for more appropriate diabetes treatment. In addition, this research can also be applied to other medical cases that require accurate classification so as to make a significant contribution in the health sector.

***Keywords**: Diabetes, K-Nearest Neighbors, Kernel Density Estimation, Classification, Outlier.*

ABSTRAK

Penyakit diabetes merupakan tantangan kesehatan global yang memerlukan deteksi dini. Keberadaan *outlier* pada data dapat memengaruhi akurasi model klasifikasi diabetes. Penelitian ini merupakan suatu penelitian kuantitatif eksperimental yang bertujuan meningkatkan akurasi klasifikasi penyakit diabetes dengan menangani *outlier* menggunakan kombinasi algoritma K-*Nearest Neighbors* (KNN) dan *Kernel Density Estimation* (KDE). Subjek penelitian menggunakan 2.000 sampel data diabetes yang bersumber dari web Kaggle. Data diproses melalui *preprocessing*, transformasi, normalisasi, deteksi *outlier* dengan KDE, imputasi *outlier*, klasifikasi KNN, dan evaluasi menggunakan confusion matrix. Hasil penelitian ini menunjukkan bahwa penggunaan KDE dan imputasi *outlier* dapat meningkatkan performa model pada semua metrik evaluasi. Model terbaik diperoleh pada KNN dengan $K = 3$ dan rasio $dataset$ 90%:10%, meningkatkan akurasi dari 90% menjadi 92%. %. Penelitian ini berpotensi meningkatkan akurasi deteksi dini penyakit diabetes yang berimplikasi pada penanganan penyakit diabetes yang lebih tepat. Selain itu, penelitian ini juga dapat diterapkan pada kasus medis lain yang memerlukan klasifikasi akurat sehingga memberikan kontribusi signifikan dalam bidang kesehatan.

Kata Kunci: Diabetes, K-Nearest Neighbors, Kernel Density Estimation, Klasifikasi, Outlier.

PENDAHULUAN

Diabetes Melitus (DM) merupakan penyakit kronis dengan gejala adanya peningkatan kadar gula darah (*glukosa*) diatas nilai normal. *Diabetes* terjadi ketika tubuh tidak dapat menyerap glukosa ke dalam sel dan menjadikannya sebgai energi.Hal ini menyebabkan penumpukkan gula ekstra dalam aliran tubuh. DM yang tidak diagnosis dan dikontrol dengan baik dapat menyebabkan kerusakan pada berbagai organ dan jaringan tubuh seperti seperti ginjal, jantung, mata, dan saraf (Azizah, Firdaus, Suyaningsih, & Indrayatna, 2023). Beberapa faktor dapat menyebabkan *diabetes*, termasuk kelelahan, *obesitas*, *diabetes melitus*, usia, dan pilihan gaya hidup, serta kebiasaan makan yang tidak baik (Lestari, Nadhiroh, & Novia, 2021). DM merupakan salah satu penyebab kematian tertinggi di dunia karena obatnya belum ditemukan serta

penyakit ini dapat menyebabkan komplikasi pada penyakit semakin parah, seperti retinopati, amputasi anggota tubuh, penyakit *kardiovaskular*, dan penyakit saraf, jika tidak diobati. Oleh karena itu, penting untuk dapat memantau dan mengetahui perkembangan penyakit *diabetes* (Hennebelle et al., 2024). Peran penting ilmu kedokteran dalam mendiagnosis penyakit DM sangat dibutuhkan. Namun, ketidak seimbangan antara jumlah dokter dan jumlah pasien *diabetes* menjadi keterbatasan sehingga alternatif memanfaatkan teknologi seperti *Machine Learning* (ML) dengan algoritma *K-Nearest Neighbors* (KNN) yang dapat membantu dalam mengetahui perkembangan dan prediksi penyakit *diabetes* (Azizah et al., 2023). Data mining merupakan bagian dari *Artificial Intelligence* (AI), ML, *statistics*, dan data *base systems*. Mendiagnosis suatu penyakit tidak selalu mudah karena perlu dilakukaan serangkaian tes sebelum diagnosis. AI dapat berperan efektif dangan menggunakan ML dalam mempelajari pola prediksi dari sejumlah besar data layanan Kesehatan (Rabie & Saleh, 2024).

Data mining dapat dijadikan sebagai acuan untuk memprediksi dan mendiagnosa suatu jenis penyakit. Salah satu penyakit *degeneratife* yang dapat diprediksi dengan menggunakan metode *data mining* adalah penyakit diabetes (Faizal Aris, 2019). *Data Mining* melibatkan pengenalan pola dalam data dengan tujuan tertentu, dan dapat digunakan untuk klasifikasi, asosiasi, estimasi, prediksi, klasifikasi, dan pengelompokan (Baharuddin, Azis, & Hasanuddin, 2019). *K-Nearest Neighbors* (KNN) merupakan suatu algoritma pembelajaran terawasi yang mengklasifikasikan *instance* kueri baru dengan mempertimbangkan sebagian besar kategori yang ada di KNN. Kelas yang paling umum di antara tetangga-tetangga tersebut menjadi kelas hasil klasifikasi (Mustafa & Simpen, 2014). Pendekatan ini mengklasifikasikan objek baru sesuai dengan atribut dan titiknya. Ketika sebuah pertanyaan diajukan, KNN akan mencari jumlah *K* titik atau titik-titik pelajaran yang paling dekat dengan pertanyaan tersebut. Nilai prediksi dari pertanyaan tersebut akan ditentukan dengan mengukur tingkat kedekatannya (Lubis, Lubis, & Al-Khowarizmi, 2020). Klasifikasi penyakit *diabetes* merupakan hal yang penting dan menjadi tantangan untuk diagnosis dan interpretasi data *diabetetes* karena data medis pada umumnya bersifat nonlinier, tidak normal, berkorelasi, dan kompleks. Data tersebut memiliki nilai yang hilang atau memiliki *outlier*, yang dapat memperngaruhi performa sistem ML (Maniruzzaman et al., 2018).

Outlier adalah suatu fenomena yang sangat berbeda dari fenomena lain yang dapat menimbulkan mekanisme yang berbeda. Keberadaan *outlier* dalam *dataset* sangat memengaruhi kualitas data dan hasil analisis *data mining*. Identifikasi *outlier* sangat penting karena dapat menunjukkan pola sistem baru yang menghasilkan data, mendeteksi kejadian yang tidak sah dalam *dataset*, dan juga dapat mengubah data (Abdul Wahid & Annavarapu Chandra Sekhara Rao, 2020). *Outlier* selain dapat mempengaruhi pengambilan kesimpulan atau keputusan penelitian, data *outlier* juga dapat menyebabkan data tidak berdistribusi normal (Sihombing, Suryadiningrat, Sunarjo, & Yuda, 2023), Mengklasifikasikan *outlier* adalah masalah utama lainnya dalam ML karena sampel data jarang mengikuti pola yang jelas. Dalam kumpulan data medis, sampel tersebut mungkin menunjukkan individu atau kelompok yang berperilaku sangat berbeda mayoritas dalam kelas yang sama. Misalnya, dalam tugas kelas biner melibatkan kelompok yang sehat dan tidak sehat. Hal ini disebabkan karena sampel menunjukkan karakteristik yang umumnya diasosiasikan dengan kelompok tidak sehat seperti indeks massa tubuh (BMI) yang tinggi. Dinamika ini berpotensi

mengganggu mekanisme pembelajaran suatu algoritma, yang pada akhirnya menyebabkan kesalahan klasifikasi (Nnamoko & Korkontzelos, 2020).

Salah satu metode paling umum untuk analisis kepadatan adalah metode *density kernel* (Thompson et al., 2022), Biasanya digunakan dalam analisis spasial untuk menentukan distribusi kepadatan variabel atau kejadian di suatu wilayah geografis (Vestal, Carlson, & Ghosh, 2021). Hasil analisis kepadatan kernel biasanya ditampilkan dalam bentuk peta panas atau permukaan kontinu yang menunjukkan tingkat kepadatan tinggi atau rendah. Peta panas ini membantu menemukan pola dan konsentrasi, menemukan pusat kepadatan penting, dan mengetahui distribusi spasial dari fenomena yang penulis pelajari (Vestal et al., 2021).

Beberapa penelitian terdahulu yang mengkaji tentang prediksi kemungkinan penderita penyakit *diabetes melitus* yaitu penelitian yang dilakukan menggunakan algoritma (KNN) pada dataset penderita penyakit diabetes. Penelitian (Abdul Wahid & Annavarapu Chandra Sekhara Rao, 2020) mengkaji tentang metodologi deteksi *outlier* dalam berbagai bidang, termasuk jaringan sensor nirkabel industri, asuransi kesehatan, asuransi otomotif, rekayasa perangkat lunak, deteksi kecurangan keuangan, dan proses manufaktur, dengan metode deteksi *outlier* berbasis kerapatan *(density-based outlier detection)* dengan memanfaatkan tiga kategori tetangga terdekat, yaitu tetangga terdekat (KNN), hasil penelitiannya menunjukkan bahwa mendeteksi *outlier* berdasarkan kepadatan lokal, yang berkinerja baik dalam mendeteksi *outlier*, terutama pada dataset dengan pola kepadatan rendah dan pola kepadatan lokal. Penelitian (Argina, 2020) mengkaji tentang metode klasifikasi KNN pada dataset pasien diabetes untuk menghitung akurasi, *presisi*, *recall*, dan *F-Measure* berdasarkan nilai *K* yang berbeda. Hasil penelitiannya menunjukkan bahwa akurasi tertinggi adalah 39% pada $K = 3$, presisi tertinggi adalah 65% pada $K = 3$ dan $K = 5$, *recall* tertinggi adalah 36% pada $K = 3$, dan *F-Measure* tertinggi adalah 46% pada $K = 3$. Penelitian (Nur Ikhromr, Sugiyarto, Faddillah, & Sudarsono, 2023) mengkaji tentang memprediksi penyakit *diabetes* dengan metode algoritma KNN. Hasil penelitiannya menunjukkan bahwa evaluasi menggunakan 2000 set data pasien diabetes *KNN* menghasilkan akurasi sebesar 99% sedangkan Naives Bayes menghasilkan akurasi sebesar 75%. Oleh karena itu, dapat disimpulkan bahwa KNN lebih mendukung untuk mencari data dalam jumlah besar.

Berdasarkan uraian dan penelitian terdahulu, dapat diketahui bahwa telah ada penelitian-penelitian yang menunjukkan efektivitas KNN dalam klasifikasi diabetes tetapi masih ada kesenjangan dalam integrasi metode deteksi *outlier* dengan metode KNN untuk meningkatkan akurasi prediksi penyakit diabetes. Penelitian (Abdul Wahid & Annavarapu Chandra Sekhara Rao, 2020) mendemonstrasikan keunggulan metode deteksi *outlier* tetapi belum menerapkannya secara spesifik pada kasus diabetes. Selanjutnya, Penelitian (Argina, 2020) dan (Nur Ikhromr et al., 2023) menunjukkan variasi performa KNN dalam klasifikasi diabetes, tetapi belum mengeksplorasi bagaimana pengaruh penanganan outlier terhadap akurasi model. Oleh karena itu, terdapat peluang untuk mengusulkan pendekatan baru yang mengombinasikan KNN dengan penanganan *outlier* sebagai langkah yang strategis untuk meningkatkan ketepatan prediksi dan mengidentifikasi *anomali* atau *outlier*.

TINJAUAN PUSTAKA

Data Mining

Data mining merupakan proses eksplorasi atau ekstraksi data dan informasi berskala besar yang, meskipun sebelumnya tidak diketahui, dapat dipahami dan berguna dari basis data berskala besar. Prosedur ini digunakan

untuk mengurangi dampak dari keputusan bisnis yang penting. Data mining terdiri dari banyak teknik yang bertujuan mencari pola yang tidak terdeteksi dalam keterangan yang dikumpulkan. Melalui cara ini, pengguna dapat memperoleh pengetahuan baru dari basis data yang sebelumnya tidak terjangkau (Zai, 2022).

K-Nearest Neighbors (KNN)

KNN dioperasikan pada kumpulan data untuk atribut yang memiliki nilai kategori dengan memanfaatkan pembelajaran dari data yang telah diklasifikasikan sebelumnya. Setelah KNN dikumpulkan, mayoritas KNN digunakan sebagai estimasi dari sampel uji. Baik itu jarak dekat atau jauh, rata- rata tetangga diambil menggunakan jarak *Euclidean*. Beberapa metode untuk mengimplementasikan metode KNN, (Cahyanti, Rahmayani, & Ainy Husniar, 2020),yaitu:

1. Menentukan parameter *K*
2. Menghitung jarak antara data *training* dan data *testing*
   Perhitungan jarak yang paling umum dipakai pada perhitungan pada algoritma *KNN* adalah menggunakan perhitungan jarak *Euclidean*, dengan menggunakan persamaan (1).

$$euc = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \qquad (1)$$

   dimana:
   p_i : *sample data training*
   q_i : *sample data testing*
   i : variabel data
   n : dimensi data
3. Mengurutkan jarak yang terbentuk
4. Menentukan jarak terdekat sampai urutan *K*
5. Memasangkan kelas yang bersesuaian
   Mencari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi(Cahyanti et al., 2020).

Kernel Density Estimation

Estimasi Densitas Kernel (KDE) adalah metode *non parametrik* untuk mengestimasi bentuk fungsi densitas, yang diberikan sebuah sampel dari distribusi. Untuk sebuah kumpulan data $X \subseteq Rd$ dan sebuah fungsi kernel $Kh : Rd \times Rd \rightarrow [0, 1]$, estimasi densitas kernel dari vektor kueri $y \in Rd$ berdasakan persamaan (2).

$$f(x, h) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

Dimana
k: *kernel function*
h: *bandwidth function*
n: jumlah sampel

KDE menggunakan fungsi kernel untuk menghaluskan data dan mengurangi efek *noise* dan variasi kecil pada data. *Bandwidth* mengontrol seberapa halus atau kasar perkiraan yang dihasilkan. semakin besar rentangnya maka semakin kasar perkiraan yang dihasilkan, dan sebaliknya (Delacourt & Scott, 2019).

Outlier

Pencilan merupakan data yang berada jauh atau terpencil dari data yang ada di populasi, biasanya disebut *outlier*. Kehadiran *outlier* memiliki implikasi penting untuk analisis data. Keberadaan *outlier* juga dapat menyebabkan bentuk distribusi data menjadi tidak normal, menciptakan bias dalam taksiran parameter serta berpengaruh terhadap hasil signifikansi pengujian parameter (Maulita Barus, 2023).

Confusion Matrix

Evaluasi dilakukan dengan metode konfusi matriks, dan nilai kinerja yang digunakan adalah *presisi* dan *precesion*, *recall*, dan skor F1. Akurasi menggambarkan tingkat efektivitas metode yang digunakan dalam proses klasifikasi(Muhaimin, Hariyadi, & Imamudin, 2024). *Recall* nantinya menunjukkan padanan rasio prediksi benar positif dengan keseluruhan jumlah data yang benar positif. Sedangkan, *precision* menunjukkan padanan dari rasio prediksi benar positif dengan keseluruhan hasil

yang diprediksi positif. *F1-score* kemudian melakukan perhitungan dengan membandingkan rata-rata nilai *recall* dan *precision* yang dibobotkan (Muhaimin et al., 2024). Confusion Matrix ditunjukkan pada Gambar 1.

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	TP (*True Positive*)	FP (*False Positive*)
	False	FN (*False Negative*)	TN (*True Negative*)

Gambar 1. Confusion Matrix

Persamaan yang digunakan untuk mengukur akurasi didefinisikan pada persamaan (1).

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} * 100\% \quad (1)$$

Persamaan yang digunakan untuk mendapatkan nilai *precision* didefinisikan pada persamaan (2).

$$Precision = \frac{TP}{TP+FP} * 100\% \quad (2)$$

Persamaan yang digunakan untuk mendapatkan nilai *recall* didefinisikan pada persamaan (3).

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (3)$$

Persamaan yang digunakan untuk mendapatkan nilai F1-score didefinisikan pada persamaan (4).

$$\text{F1} - \text{score} = \frac{2*precision* recall}{precision+recall} * 100\% \quad (4)$$

Hipotesis Penelitian

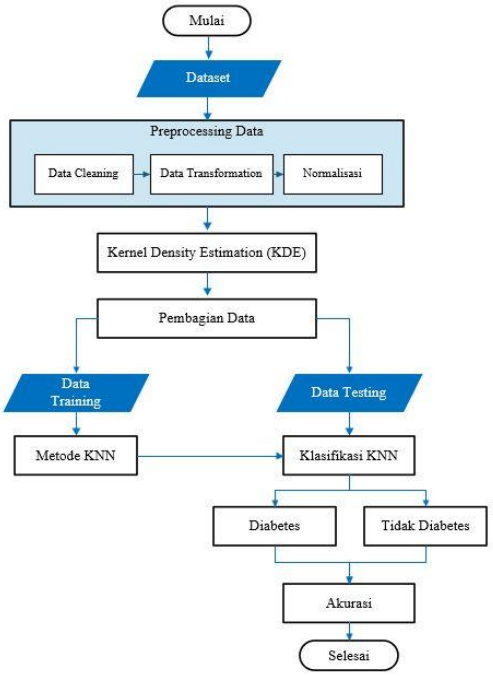
Penelitian ini menggunakan hipotesis:

H_0: Penanganan *outlier* dengan metode KDE dan KNN tidak dapat menghasilkan peningkatan yang signifikan dalam akurasi klasifikasi penyakit diabetes apabila dibandingkan dengan penggunaan KNN tanpa penanganan *outlier*.

H_1 : Penanganan *outlier* menggunakan metode KDE dan algoritma KNN menghasilkan peningkatan yang signifikan dalam akurasi klasifikasi penyakit diabetes dibandingkan dengan penggunaan KNN tanpa penanganan *outlier*.

METODE

Metode penelitian ditunjukkan pada Gambar 2.



Gambar 2. Metode Penelitian

Gambar 2 merupakan metode yang diterapkan dalam penelitian ini, yaitu menggunakan kombinasi algoritma KNN dan KDE. Data latih yang memiliki luaran *class* atau *label* yang telah ditentukan digunakan untuk melatih model algoritma KNN. Proses pelatihan ini dilakukan dengan menggunakan teknik *supervised learning*, dimana model belajar dari data latih yang telah diberi label. Selanjutnya, model yang telah dilatih digunakan untuk memprediksi hasil klasifikasi dari data uji yang akan dimasukkan berikutnya. Penggunaan metode *supervised learning* menjadi penting dalam mengoptimalkan pembentukan model klasifikasi dengan memanfaatkan data penjualan yang telah diberi label untuk memperoleh informasi yang lebih akurat.

Pengumpulan Data

Pengumpulan data adalah tahap pertama dalam penelitian ini. Penelitian ini menggunakan dataset Diabetes yang tersedia untuk umum dapat diakses oleh publik. *Dataset* diperoleh bersumber dari Kaggle (Nur kharisa umami, 2021).

Preprocessing Data

Tahap *preprocessing/cleaning* merupakan proses memeriksa kembali data dan membersihkannya, terutama data *redundant*. Data yang sedang dianalisis harus bebas dari anomali dan kosongan. Oleh maka itu, pencilan dan data kosong perlu diidentifikasi dan diatasi untuk memastikan bahwa algoritma menghasilkan hasil yang konsisten dan relevan, dengan tahapan sebagai berikut: (Widaningsih & Yusuf, 2022).

a) *Missing Value*

Pada tahap ini, data disiapkan dengan menghilangkan nilai yang hilang (*missing value*) atau data yang tidak konsisten.

b) Normalisasi

Pada tahap ini, normalisasi data sangat penting untuk mengurangi data dengan penyebaran nilai yang besar sehingga dapat dinormalisasi dengan membuat rentang nilai yang lebih kecil.

Proses normalisasi ini menggunakan metode *min-max normalization*, yang mengurangi nilai rent ke rentang 0-1. Normalisasi data diberikan pada persamaan (1) (Melina, Napitupulu, Sambas, Murniati, & Adimurti Kusumaningtyas, 2022):

$$x' = \frac{x - b}{a - b} \qquad (1)$$

dimana

x': data yang telah dinormalisasikan

x : data asli

a : data dengan nilai terbesar dari data asli

b : data dengan nilai terkecil dari data asli.

c) Transformasi

Transformasi adalah serangkaian petunjuk yang digunakan agar mengubah *input* ke *output* yang dituju dengan mengikuti siklus *input*-proses-*output*. Proses ini melibatkan transformasi data ke format yang diinginkan, yang kemudian ditentukan oleh algoritma klasifikasi yang ingin diterapkan. Normalisasi digunakan pada data numerik untuk menstandarkan perbedaan skala yang mungkin berdampak pada hasil yang diperoleh.

Pembagian Data

Untuk membuat prediksi menggunakan model, diperlukan *dataset* yang akan dibagi menjadi dua bagian-yaitu *data training* dan *data testing*. Data latih digunakan untuk membuat model regresi, yang kemudian disesuaikan dengan menggunakan data uji, dalam penelitian ini *split data* dilakukan tiga kali dengan pembagian data 90% : 10%, 80% : 20% dan 70% : 30%.

Evaluasi

Setelah menghitung akurasi, langkah berikutnya adalah mengevaluasi kinerja algoritma yang digunakan. Evaluasi ini mencakup penggunaan *Confusion Matrix* dan menampilkan nilai-nilai pada *Classification Report*, seperti *Precision*, *Recall*, dan *F1-Score*. Setelah seluruh tahapan eksperimen selesai, langkah berikutnya adalah menentukan hasil pengujian mana yang memiliki akurasi terbaik: apakah eksperimen *K-Nearest Neighbors* tanpa KDE atau dengan KDE. Penelitian ini juga menganalisis *K* mana yang memiliki nilai akurasi tertinggi dan *split data* mana saja yang berpengaruh terhadap label yang ada pada *dataset.*

HASIL DAN PEMBAHASAN

Berdasarkan penelitian ini, dilakukan perbandingan antara dua metode dalam klasifikasi menggunakan *K-Nearest Neighbors* tanpa KDE, dan dengan KDE pada *dataset* penyakit *Diabetes*.

Deskripsi *Dataset*

Penelitian ini menggunakan *dataset* dari *Kaggle* terkait dengan Penyakit *Diabetes*. Berikut adalah beberapa informasi tentang *dataset*:

- Jumlah Data : Terdapat 2.000 data dalam *dataset*.
- Atribut Fitur: Terdapat 8 atribut fitur dan 1 *class lable* yang mencakup berbagai informasi seperti *Pregnancies, Glucose, Blood Pressure, Skin*

Thickness, Insulin, BMI, Diabetes Pedigree Function, Age, dan Outcome.

Preprocessing

Pada tahap ini. data diproses terlebih dahulu sebelum digunakan dalam model klasifikasi. Proses ini melibatkan penanganan nilai yang hilang, transformasi, dan normalisasi fitur *numerik*. Selain itu, data dibagi menjadi set pelatihan dan pengujian untuk validasi model, seperti yang ditunjukkan pada Gambar 3.

```
# Split data
def evaluate_knn(X, y, test_size, n_neighbors):
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=42)\
# Define test sizes
test_sizes = [0.1, 0.2, 0.3]  # 90/10, 80/20, 70/30
```

Gambar 3. Split Data

Gambar 1.2 menunjukkan pembagian dataset menjadi dua bagian utama yaitu data *training* (pelatihan) dan data *test* (pengujian). *Split data* dilakukan tiga kali dengan pembagian data 90% : 10%, 80% : 20% dan 70% : 30%.

K-Nearest Neighbors* dan *Kernel Density Estimation

Hasil pengujian pertama dilakukan menggunakan *dataset* dengan perbandingan 90%:10% dan dengan variasi nilai *K* = 1, 3, 5, dan 7, yang ditunjukkan pada Gambar 4.

```
# Fit KDE model for outlier detection
kde = KernelDensity(kernel='gaussian', bandwidth=0.5).fit(X_scaled)log_density = kde.score_samples(X_scaled)
# Detect outliers (e.g., density in the lowest 5%)
threshold = np.percentile(log_density, 5)
outliers = log_density < threshold
# Print detected outliers
print(f"Detected outliers: {np.sum(outliers)}")
```

```
# Impute outliers with median
df_scaled = pd.DataFrame(X_scaled, columns=df.columns[:-1])
        df_scaled['Outcome'] = df['Outcome']
        df_imputed = df_scaled.copy()
        for col in df.columns[:-1]:
median = df_scaled.loc[~outliers, col].median()
df_imputed.loc[outliers, col] = median
# Prepare data for training
X_imputed = df_imputed.drop('Outcome', axis=1)y_imputed = df_imputed['Outcome']
def evaluate_knn(X, y, test_size, n_neighbors):
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=42)
knn = KNeighborsClassifier(n_neighbors=n_neighbors)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
f1 = f1_score(y_test, y_pred, average='macro')
report = classification_report(y_test, y_pred)
return accuracy, precision, recall, f1, report

# Define test sizes and K values
test_sizes = [0.1, 0.2, 0.3]  # 90/10, 80/20, 70/30
k_values = [1, 3, 5, 7]
# Evaluate for each combination of test size and K
results = []
for test_size in test_sizes:
for k in k_values:
accuracy, precision, recall, f1, report = evaluate_knn(X_imputed, y_imputed, test_size, k)
            result = {
                'test_size': test_size,
                'k': k,
```

```
                    'accuracy': accuracy,
                    'precision': precision,
                    'recall': recall,
                    'f1': f1,
                    'report': report
                }
results.append(result)
print(f"Test size: {test_size}, K: {k}")
print(f"Accuracy: {accuracy:.2f}, Precision: {precision:.2f}, Recall: {recall:.2f}, F1: {f1:.2f}")
print(report)
print('-' * 50)
```

Gambar 4. Algoritma K-Nearest Neighbors dan Kernel Density Estimation

Gambar 4 menggambarkan penerapan algoritma *K-Nearest Neighbors* dengan *Kernel Density Estimation* yang dilatih menggunakan data latih, dan dievaluasi dengan *classification report* untuk melihat kinerjanya pada data uji.

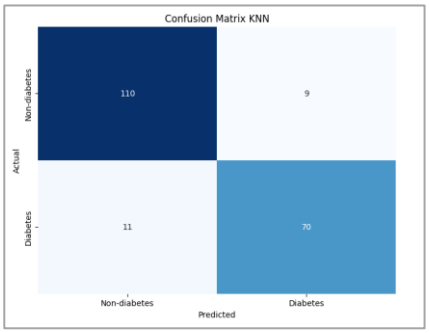
Hasil pengujian pertama dilakukan menggunakan dataset dengan perbandingan 90%:10% dan dengan variasi nilai K = 1, 3, 5, dan 7. Performa model dapat dilihat pada Tabel 1.

Tabel 1. Hasil Klasifikasi K-Nearest Neighbors dengan Kernel Density Estimation

Performa	KNN				KNN dan Imputasi Outlier			
	1	3	5	7	1	3	5	7
Akurasi	0.99	0.90	0.78	0.81	0.98	**0.92**	0.79	0.81
Presisi	0.99	0.90	0.77	0.81	0.98	**0.91**	0.78	0.81
Recall	0.99	0.89	0.76	0.78	0.98	**0.91**	0.78	0.80
F1-Score	0.99	0.90	0.77	0.79	0.98	**0.91**	0.78	0.80

Berdasarkan Tabel 1, *dataset* dengan rasio 90% : 10% menunjukan bahwa nilai akurasi meningkat dari sekitar 90% tanpa *imputasi outlier* menjadi 92% dengan *imputasi outlier*. Terlihat bahwa penggunaan imputasi *outlier* secara signifikan meningkatkan kinerja model pada semua metrik evaluasi, termasuk akurasi, *presisi*, *recall*, dan *F1-Score*. KNN dan *imputasi outlier* secara konsisten memberikan hasil yang lebih baik dibandingkan KNN tanpa *imputasi outlier*, terutama pada nilai *K* yang lebih rendah $(K = 3)$.

Confusion matrix adalah alat yang sangat berguna dalam evaluasi kinerja model klasifikasi. Matrix ini memberikan informasi tentang jumlah prediksi yang benar dan salah yang dilakukan oleh model, dengan rincian yang ditunjukkan pada Gambar 1.4.



Gambar 5. Confusion Matrix KNN

Gambar 5 merupakan performa model dalam memprediksi penyakit diabetes dengan Confusion Matrix KNN, yaitu:

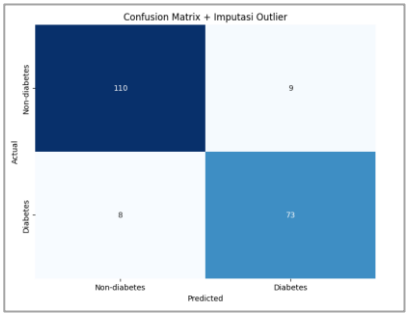
***True Negative* (TN)**: 110
- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

***False Positive* (FP)**: 9
- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

***False Negative* (FN)**: 11
- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

***True Positive* (TP)**: 70
- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Selanjutnya performa model dalam memprediksi penyakit diabetes menggunakan Confusion Matrix KNN dengan Imputasi Median, ditunjukkan pada Gambar 1.5.



Gambar 6. Confusion Matrix KNN dan Imputasi Median

Gambar 6 merupakan Confusion Matrix KNN dengan Imputasi Median, dengan data sebagai berikut:

True Negative **(TN)**: 110
- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive **(FP)**: 9
- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

False Negative **(FN)**: 8
- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive **(TP)**: 73
- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Memahami distribusi ini berarti dapat dihitung beberapa metrik kinerja penting seperti akurasi, *presisi, recall, dan F1-score* yang membantu dalam mengevaluasi sejauh mana model klasifikasi berhasil dalam tugasnya. Laporan klasifikasi *(classification report)* untuk model *K-Nearest Neighbors (KNN)* yang digunakan dalam klasifikasi diabetes memberikan beberapa metrik kinerja penting untuk setiap kelas (Negatif dan Positif) yang diberikan pada Tabel 2 dan Tabel 3.

Tabel 2. Clasification Report KNN

	Presisi	Recall	F1-Score	Support
Negatif	0.91	0.92	0.92	119
Positif	0.98	0.86	0.87	81

Tabel 3. Clasification Report KNN dan Imputasi Outlier

	Presisi	Recall	F1-Score	Support
Negatif	0.93	0.92	0.93	119
Positif	0.89	0.90	0.90	81

Presisi: Nilai presisi untuk kelas Negatif pada Tabel 1 adalah 0.91, sedangkan untuk kelas Positif adalah 0.98.
Recall: Nilai *recall* untuk kelas Negatif pada Tabel 1adalah 0.92, sedangkan untuk kelas Positif adalah 0.86.
F1-Score: Nilai *F1-score* untuk kelas Negatif pada Tabel 1adalah 0.92, sedangkan untuk kelas *Positif* adalah 0.87.
Support: Jumlah aktual sampel dari setiap kelas dalam *dataset* yang digunakan. *Support* untuk kelas Negatif dan Positif pada kedua tabel adalah 119 dan 81, masing-masing, yaitu. :

- **Performa Kelas Negatif**:
  Tabel 1 menunjukkan *presisi* 0.91, *recall* 0.92, dan F1-score 0.92.
  Tabel 2 menunjukkan *presisi* 0.93, *recall* 0.92, dan F1-score 0.93.
  Performa kelas Negatif sedikit meningkat pada evaluasi kedua (Tabel 2) dibandingkan dengan yang pertama (Tabel 1).
- **Performa Kelas Positif**:
  Tabel 1 menunjukkan presisi 0.98, *recall* 0.86, dan F1-score 0.87.
  Tabel 2 menunjukkan presisi 0.89, *recall* 0.90, dan F1-score 0.90.
  Pada evaluasi kedua (Tabel 2), *recall* dan F1-score untuk kelas Positif meningkat dibandingkan dengan evaluasi pertama (Tabel 1), meskipun presisinya sedikit menurun.

Secara keseluruhan, evaluasi ketiga pada Tabel 3 menunjukkan peningkatan keseimbangan antara *presisi* dan *recall*, terutama untuk kelas Positif. Ini bisa mengindikasikan peningkatan performa model dalam mendeteksi kasus diabetes dengan benar.

Hasil pengujian kedua dilakukan menggunakan *dataset* dengan perbandingan 80%:20% dan dengan variasi nilai $K = 1, 3, 5,$ dan 7. Performa model dapat dilihat pada Tabel.

Tabel 4. Hasil Klasifikasi K-Nearest Neighbors dengan Kernel Density Estimation

Performa	KNN				KNN dan Imputasi Outlier			
	1	3	5	7	1	3	5	7
Akurasi	0.98	0.84	0.82	0.81	0.96	**0.84**	0.82	0.81
Presisi	0.98	0.84	0.81	0.80	0.96	**0.83**	0.81	0.80
Recall	0.98	0.82	0.80	0.78	0.95	**0.82**	0.81	0.79
F1-Score	0.98	0.83	0.81	0.79	0.95	**0.83**	0.81	0.79

Berdasarkan Tabel 4, pada dataset dengan rasio 80%:20%, dapat disimpulkan bahwa penggunaan imputasi *outlier* sedikit mengurangi performa model KNN dalam hal akurasi, *presisi*, *recall*, dan *F1-Score*. Meskipun perbedaannya tidak signifikan, KNN tanpa imputasi *outlier* menunjukkan keunggulan dalam kemampuan umumnya untuk memprediksi kelas dengan benar.

Hasil pengujian ketiga dilakukan menggunakan *dataset* dengan perbandingan 70%:30% dan dengan variasi nilai *K = 1, 3, 5,* dan *7*. Performa model dapat dilihat pada Tabel 5.

Tabel 5. Hasil Klasifikasi K-Nearest Neighbors Dengan Kernel Density Estimation

Performa	KNN				KNN dan Imputasi Outlier			
	1	3	5	7	1	3	5	7
Akurasi	0.95	0.83	0.80	0.79	**0.94**	0.81	0.79	0.78
Presisi	0.94	0.81	0.78	0.77	**0.93**	0.81	0.78	0.76
Recall	0.94	0.81	0.78	0.76	**0.94**	0.78	0.75	0.73
F1-Score	0.94	0.81	0.78	0.76	**0.93**	0.79	0.76	0.74

Berdasarkan Tabel 5, pada *dataset* dengan rasio 70%:30%, terlihat bahwa nilai akurasi pada $K = 1$ untuk KNN tanpa imputasi *outlier* sedikit lebih tinggi (0.95) dibandingkan dengan KNN dengan *imputasi outlier* (0.94). Penggunaan *imputasi outlier* tidak secara signifikan meningkatkan kinerja model pada semua metrik evaluasi (akurasi, *presisi*, *recall*, dan *F1-Score*). Namun, perbedaan kinerja antara KNN dengan dan tanpa imputasi *outlier* cenderung kecil, dan performa kedua model menjadi lebih seimbang pada nilai *K* yang lebih tinggi. Hasil penelitian ini menunjukkan bahwa penanganan *outlier* dengan KDE yang dikombinasikan dengan KNN berhasil meningkatkan performa model pada semua metrik evaluasi. Model terbaik diperoleh pada KNN dengan rasio pembagian data training dan testing 90%:10%.

Berdasarkan hasil penelitian yang telah dilakukan dalam penelitian ini dapat dinyatakan bahwa hasil penelitian ini sejalan dengan penelitian terdahulu (Abdul Wahid & Annavarapu Chandra Sekhara Rao, 2020) yang membuktikan keefektifan metode deteksi outlier berbasis densitas dalam berbagai bidang. Penerapan metode ini pada kasus diabetes membuktikan bahwa pendekatan ini efektif dalam konteks medis, khususnya untuk penyakit diabetes. Hasil penelitian ini juga memperkuat penelitian terdahulu (Argina, 2020), yang menunjukkan variasi dari performa KNN pada *dataset* diabetes yang mencapai akurasi tertinggi pada 39%. Sedangkan hasil yang diperoleh dari penelitian ini menunjukkan terjadinya peningkatan yang lebih signifikan dengan akurasi mencapai 92%. Hal ini mengindikasikan bahwa penanganan *outlier* dengan KDE memberikan kontribusi dalam meningkatkan akurasi KNN. Selanjutnya, pada penelitian terdahulu (Nur Ikhromr et al., 2023), yang memperoleh akurasi KNN sebesar 99% pada dataset diabetes. Sedangkan hasil penelitian ini mencapai akurasi pada 92%, yang lebih rendah. Namun, fokus penelitian ini adalah pada penanganan *outlier*, yang mungkin tidak dipertimbangkan dalam (Nur Ikhromr et al., 2023). Perbedaan ini juga bisa disebabkan oleh adanya variasi dalam karakteristik *dataset* atau metode *preprocessing* yang digunakan. Secara keseluruhan, hasil penelitian ini memperkuat temuan penelitian terdahulu tentang efektivitas KNN dalam klasifikasi diabetes

SIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa peningkatan akurasi yang diperoleh dalam penelitian ini menunjukkan kombinasi algoritma KNN dan KDE dapat menjadi pendekatan baru untuk meningkatkan kualitas prediksi dalam bidang kesehatan, khususnya untuk penyakit diabetes.

DAFTAR PUSTAKA

Covariance Determinant (MMCD). In *IJM: Indonesian Journal of Multidisciplinary* (Vol. 1). Retrieved from https://journal.csspublishing/index.php/ijm

Melina Universitas Jenderal Achmad Yani, M., Napitupulu, H., Sambas, A.,

Murniati, A., & Adimurti Kusumaningtyas, V. (n.d.). *Artificial Neural Network-Based Machine Learning Approach to Stock Market Prediction Model on the Indonesia Stock Exchange During the COVID-19*. Retrieved from https://www.researchgate.net/publication/362983602

Muhaimin, A., Hariyadi, M. A., & Imamudin, M. (2024). Klasifikasi Prestasi Akademik Siswa Berdasarkan Nilai Rapor dan Kedisiplinan dengan Metode K-Nearest Neighbor. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI*, *7*(1), 193–202.

Mustafa, M. S., & Simpen, W. (n.d.). *Perancangan Aplikasi Prediksi Kelulusan Tepat Waktu Bagi Mahasiswa Baru Dengan Teknik Data Mining (Studi Kasus: Data Akademik Mahasiswa STMIK Dipanegara Makassar)*.

Nnamoko, N., & Korkontzelos, I. (2020). Efficient treatment of outliers and class imbalance for diabetes prediction. *Artificial Intelligence in Medicine*, *104*. https://doi.org/10.1016/j.artmed.2020.101815

Nur Ikhromr, F., Sugiyarto, I., Faddillah, U., & Sudarsono, B. (2023). IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI PENYAKIT DIABETES MENGGUNAKAN ALGORITMA NAIVES BAYES DAN K-NEAREST NEIGHBOR IMPLEMENTATION OF DATA MINING TO PREDICT DIABETES DISEASE USING NAIVES BAYES AND K-NEAREST NEIGHBOR ALGORITHMS. *Journal of Information Technology and Computer Science (INTECOMS)*, *6*(1).

Nur kharisa umami. (2021). Kaggle. Retrieved June 27, 2024, from Kaggle website: https://www.kaggle.com/code/nurkharisaumami/klasifikasi-penyakit-diabetes/input

Rabie, A. H., & Saleh, A. I. (2024). Diseases diagnosis based on artificial intelligence and ensemble classification. *Artificial Intelligence in Medicine*, *148*, 102753. https://doi.org/10.1016/J.ARTMED.2023.102753

Sihombing, P. R., Suryadiningrat, S., Sunarjo, D. A., & Yuda, Y. P. A. C. (2023). Identifikasi Data Outlier (Pencilan) dan Kenormalan Data Pada Data Univariat serta Alternatif Penyelesaiannya. *Jurnal Ekonomi Dan Statistik Indonesia*, *2*(3), 307–316. https://doi.org/10.11594/jesi.02.03.07

Sistem Komputer dan Sistem Informasi, J., Studi Teknologi Komputasi dan Informatika Stmik Bina Bangsa Kendari, P., Aris, F., Program Studi Sistem Komputer, D., Studi Sistem Komputer, P., & Bina Bangsa Kendari, S. (2019). *Router Research Penerapan Data Mining untuk Identifikasi Penyakit Diabetes Melitus dengan Menggunakan Metode Klasifikasi* (Vol. 1). Retrieved from http://ejournal.stipwunaraha.ac.id/index.php/router

Thompson, A. E., Walden, J. P., Chase, A. S. Z., Hutson, S. R., Marken, D. B., Cap, B., … Chase, D. Z. (2022). Ancient Lowland Maya neighborhoods: Average Nearest Neighbor analysis and kernel density models, environments, and urban scale. *PLoS ONE*, *17*(11 November). https://doi.org/10.1371/journal.pone.0275916

Vestal, B. E., Carlson, N. E., & Ghosh, D. (2021). Filtering spatial point patterns using kernel densities. *Spatial Statistics*, *41*. https://doi.org/10.1016/j.spasta.2020.100487

Widaningsih, S., & Yusuf, S. (2022). Penerapan Data Mining Untuk Memprediksi Siswa Berprestasi Dengan Menggunakan Algoritma K Nearest Neighbor. *Jurnal Teknik Informatika Dan Sistem Informasi*, *9*(3). Retrieved from http://jurnal.mdp.ac.id

Zai, C., & Komputer, T. (n.d.). IMPLEMENTASI DATA MINING SEBAGAI PENGOLAHAN DATA. In *Portaldata.org* (Vol. 2).