

# Comparison of Data Normalization Techniques on KNN Classification Performance for Pima Indians Diabetes Dataset

Yohanes Dimas Pratama<sup>1\*</sup>, Abu Salam<sup>2\*\*</sup>

\* Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

[111202113254@mhs.dinus.ac.id](mailto:111202113254@mhs.dinus.ac.id)<sup>1</sup>, [abu.salam@dsn.dinus.ac.id](mailto:abu.salam@dsn.dinus.ac.id)<sup>2</sup>

## Article Info

### Article history:

Received 2025-03-27

Revised 2025-04-12

Accepted 2025-04-14

### Keyword:

*Data Normalization,  
K-Nearest Neighbors,  
Diabetes Classification,  
Min-Max Scaling,  
Z-Score Scaling.*

## ABSTRACT

This study analyzes the comparison of data normalization techniques in the K-Nearest Neighbors (KNN) model for diabetes classification using the Pima Indians Diabetes dataset. The three normalization techniques evaluated are Min-Max Scaling, Z-Score Scaling, and Decimal Scaling. After preprocessing, such as handling missing values and removing duplicates, as well as feature selection using the Random Forest method, the features removed include SkinThickness, Insulin, Pregnancies, and BloodPressure. The evaluation was carried out using accuracy, precision, recall, F1-Score, specificity, and ROC AUC metrics. The results show that Min-Max Scaling provides a significant improvement in all metrics, with the highest accuracy of 0.8117 and ROC AUC of 0.8050. Z-Score Scaling provides good results, but not as good as Min-Max Scaling. Decimal Scaling shows the lowest performance. Statistical tests using Paired T-Test show significant differences between Min-Max Scaling and without normalization on all metrics (P-Value <0.05), while Z-Score Scaling and Decimal Scaling are only significant on some metrics, with P-Values of 0.08363 and 0.43839 respectively for accuracy and ROC AUC. Overall, Min-Max Scaling proved to be the best normalization method for improving KNN performance in diabetes classification.



This is an open access article under the [CC-BY-SA](#) license.

## I. PENDAHULUAN

Diabetes merupakan salah satu penyakit tidak menular yang semakin meningkat prevalensinya di seluruh dunia, termasuk di Indonesia. Menurut data dari Organisasi Kesehatan Dunia (WHO), jumlah penderita diabetes meningkat dari 108 juta pada tahun 1980 menjadi 422 juta pada tahun 2014. Diperkirakan jumlah penderita diabetes akan mencapai 578 juta pada tahun 2030 dan 700 juta pada tahun 2045 [1]. Diabetes terjadi ketika tubuh tidak bisa mengatur kadar gula darah dengan baik, yang menyebabkan kadar gula darah terlalu tinggi [2][3]. Diabetes dapat menyebabkan berbagai komplikasi kesehatan serius, seperti kerusakan pada jantung, ginjal, mata, dan sistem saraf [3][4]. Oleh karena itu, deteksi dini dan klasifikasi risiko diabetes sangat penting untuk mencegah terjadinya komplikasi lebih lanjut. Dalam hal ini, teknologi informasi dan pembelajaran mesin dapat berperan penting dalam mempermudah dan mempercepat diagnosis serta klasifikasi penyakit diabetes.

Salah satu algoritma pembelajaran mesin yang umum digunakan dalam klasifikasi adalah K-Nearest Neighbors (KNN). KNN merupakan metode klasifikasi yang bekerja berdasarkan kedekatan jarak antara titik data yang akan diklasifikasikan dengan data yang sudah terlabel [5]. KNN mengandalkan perhitungan jarak seperti Euclidean Distance, yang mengukur jarak terpendek antara dua titik, dengan menghitung rumus akar kuadrat dari jumlah kuadrat selisih nilai pada setiap dimensi [6]. Meskipun sederhana dan mudah diimplementasikan, tantangan utama dalam penggunaan KNN adalah sensitivitasnya terhadap data yang belum dilakukan normalisasi. Jika data memiliki skala atau satuan yang berbeda, hal ini dapat menyebabkan ketidakseimbangan dalam perhitungan jarak dan menurunkan kinerja model [7]. Karena itu, normalisasi data menjadi langkah penting sebelum diterapkan pada algoritma KNN. Normalisasi bertujuan untuk mengubah nilai fitur ke dalam skala yang seragam, sehingga perhitungan jarak antar data dapat dilakukan secara akurat tanpa dipengaruhi perbedaan skala

[8]. Terdapat tiga teknik normalisasi yang relevan dengan perhitungan Euclidean Distance, yaitu Min-Max Scaling, Z-Score Scaling, dan Decimal Scaling [9].

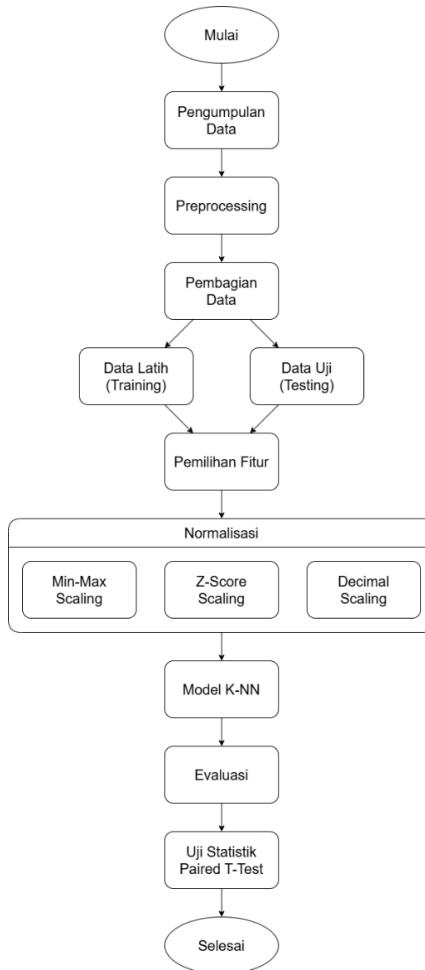
Beberapa penelitian sebelumnya telah membandingkan teknik normalisasi terhadap kinerja berbagai algoritma klasifikasi, yang berdampak signifikan pada akurasi klasifikasi di berbagai dataset. Sebagai contoh, penelitian Muasir Pagan et al. [10] membandingkan teknik normalisasi data yaitu Min-Max Scaling, Z-Score Scaling, dan Decimal Scaling terhadap kinerja algoritma KNN menggunakan sepuluh dataset. Hasil penelitian menunjukkan bahwa Z-Score Scaling dan Decimal Scaling memberikan kinerja yang lebih baik dibandingkan Min-Max Scaling, dengan Z-Score Scaling secara konsisten menghasilkan akurasi, precision, recall, dan F1-score yang lebih tinggi di sebagian besar dataset. Temuan ini menyoroti pentingnya pemilihan teknik normalisasi yang tepat berdasarkan karakteristik dataset. Selanjutnya, penelitian Alshdaifat et al. [11] juga mengevaluasi dampak teknik normalisasi, yaitu Min-Max Scaling, Z-Score Scaling, dan Decimal Scaling terhadap kinerja algoritma klasifikasi seperti SVM dan ANN, dengan temuan yang juga relevan untuk KNN. Hasil penelitian pada 18 dataset menunjukkan bahwa Z-Score Scaling sering kali memberikan hasil terbaik karena kemampuannya menangani outlier, sementara Decimal Scaling kurang efektif dalam meningkatkan akurasi model secara keseluruhan. Selain itu, penelitian Saichon Sinsomboonthong [12] membandingkan kinerja delapan teknik normalisasi dalam klasifikasi menggunakan algoritma ANN pada enam dataset. Hasil penelitian menunjukkan bahwa Min-Max Scaling umumnya memberikan akurasi tertinggi dan MSE terendah. Namun, pada beberapa dataset, seperti White Wine Quality dan Pima Indians Diabetes, Adjusted-2 Min-Max Scaling memberikan hasil yang lebih baik. Teknik normalisasi Decimal Scaling, juga menunjukkan hasil kompetitif pada dataset tertentu, meskipun tidak mengungguli Min-Max Scaling.

Penelitian ini bertujuan untuk menganalisis perbandingan berbagai teknik normalisasi terhadap kinerja model KNN dalam klasifikasi diabetes pada dataset Pima Indians Diabetes. Teknik normalisasi diterapkan untuk mengatasi perbedaan rentang nilai antar fitur dan mengevaluasi dampaknya pada kinerja model [13]. Selain itu, pemilihan fitur juga dilakukan untuk memastikan hanya fitur relevan yang digunakan, sehingga dapat mengurangi overfitting [14]. Dengan membandingkan teknik normalisasi yang berbeda, diharapkan dapat dibangun model yang lebih akurat dan efisien, mendukung deteksi dini diabetes, serta membantu tenaga medis dalam pengambilan keputusan yang lebih tepat.

## II. METODE

Penelitian ini dimulai dengan pengumpulan data dari platform Kaggle, diikuti dengan tahapan preprocessing yang mencakup pemeriksaan tipe data, missing values, data duplikat, skala fitur numerik, dan nilai unik pada fitur kategorikal. Dataset kemudian dibagi menjadi data latih dan data uji. Selanjutnya, dilakukan pemilihan fitur untuk

menentukan fitur yang paling relevan dengan model. Data kemudian dilakukan normalisasi menggunakan tiga teknik, yaitu Min-Max Scaling, Z-Score Scaling, dan Decimal Scaling. Setelah normalisasi, model KNN dilatih dan dievaluasi kinerjanya. Hasil evaluasi kinerja masing-masing teknik normalisasi dibandingkan dengan kondisi tanpa normalisasi melalui uji statistik Paired T-Test untuk menguji signifikansi. Diagram alur penelitian ini disajikan pada Gambar 1.



Gambar 1. Alur Penelitian

### A. Pengumpulan Data

Pada tahap pengumpulan data, penelitian ini menggunakan dataset Pima Indians Diabetes yang diambil dari Kaggle. Dataset ini berasal dari National Institute of Diabetes and Digestive and Kidney Diseases dengan tujuan untuk mengklasifikasi secara diagnostik apakah seorang pasien mengidap diabetes atau tidak berdasarkan berbagai fitur pengukur medis [15]. Dataset ini mencakup 768 baris data, di mana setiap baris mewakili seorang pasien. Setiap baris terdiri dari 8 fitur independen yang digunakan untuk mengklasifikasi kemungkinan diabetes, serta 1 fitur target yang menunjukkan hasil diagnosis diabetes. Fitur independen tersebut meliputi Pregnancies, Glucose, BloodPressure, SkinThickness,

Insulin, BMI, DiabetesPedigreeFunction, dan Age, sedangkan fitur target adalahnya Outcome [16]. Semua pasien dalam dataset ini adalah perempuan berusia minimal 21 tahun dengan latar belakang etnis Pima Indian [17]. Penjelasan lebih detail mengenai fitur dalam dataset disajikan pada Tabel 1.

TABEL 1  
FITUR DATASET PIMA INDIANS DIABETES

Fitur	Deskripsi
Pregnancies	Jumlah kehamilan yang pernah dialami oleh pasien.
Glucose	Konsentrasi glukosa plasma dalam darah pasien.
BloodPressure	Tekanan darah diastolik yaitu tekanan darah saat relaksasi jantung.
SkinThickness	Ketebalan lipatan kulit trisep yang diukur di lengan atas.
Insulin	Kadar insulin yang terdapat dalam darah pasien.
BMI	Indeks massa tubuh yang menunjukkan perbandingan berat badan pasien relatif terhadap tinggi badan.
Diabetes Pedigree Function	Riwayat penyakit diabetes dalam keluarga sebagai indikator hubungan keturunan pasien terkait diabetes.
Age	Usia pasien.
Outcome	Variabel target yang menunjukkan apakah pasien didiagnosis mengidap diabetes (1) atau tidak (0).

### B. Preprocessing

Preprocessing adalah tahapan penting dalam pengolahan dataset yang bertujuan untuk meningkatkan kualitas data sebelum digunakan dalam pemodelan. Proses ini dilakukan agar informasi yang diekstraksi lebih akurat, sehingga dapat berkontribusi pada peningkatan performa model [18]. Pada penelitian ini, preprocessing mencakup beberapa langkah utama yang dimulai dengan memeriksa tipe data agar setiap fitur memiliki format yang sesuai dan konsisten. Hal ini penting untuk mencegah kesalahan dalam pemrosesan data, terutama saat menerapkan algoritma pembelajaran mesin seperti KNN (K-Nearest Neighbors), yang hanya bisa menghitung jarak antar data jika data berformat numerik [19]. Selanjutnya, dilakukan identifikasi dan penanganan missing values pada setiap fitur. Dalam penelitian ini, data yang memiliki missing values akan dihapus untuk memastikan bahwa hanya data yang lengkap yang digunakan dalam analisis. Pendekatan ini dipilih karena jumlah data yang hilang relatif kecil, sehingga penghapusan tidak berdampak signifikan terhadap keseluruhan dataset [20]. Tahap berikutnya adalah deteksi dan penghapusan data duplikat, karena data yang terduplikasi dapat membuat model terlalu fokus pada pola tertentu, sehingga mengurangi kemampuan generalisasi dan menurunkan akurasi klasifikasi [21]. Setelah itu, fitur dalam dataset diklasifikasikan berdasarkan jenisnya menjadi fitur numerik dan fitur kategorikal. Untuk fitur numerik, dilakukan analisis skala untuk memahami distribusi dan rentang nilainya. Sementara itu, untuk fitur kategorikal,

dilakukan identifikasi terhadap nilai unik yang terdapat di dalamnya untuk mengetahui karakteristik tipe data [22].

### C. Pembagian Data

Pada tahap pembagian data, dataset yang telah melalui proses preprocessing akan dibagi menjadi dua bagian, yaitu data latih dan data uji. Pembagian ini dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji. Data latih akan digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi performa model yang telah dilatih. Pembagian ini bertujuan untuk memungkinkan model mempelajari pola dari data latih dan menguji kemampuannya dalam mengklasifikasi hasil pada data uji yang belum pernah dilihat sebelumnya, sehingga dapat dinilai kemampuan generalisasi model [23].

### D. Pemilihan Fitur

Setelah pembagian dataset menjadi data pelatihan dan pengujian, langkah selanjutnya adalah melakukan pemilihan fitur menggunakan metode Random Forest. Pemilihan fitur adalah proses penting untuk memilih fitur-fitur yang paling relevan dan signifikan dalam model, sehingga dapat meningkatkan akurasi serta efisiensi komputasi [24]. Random Forest, yang merupakan algoritma berbasis pohon keputusan, dapat digunakan untuk menentukan pentingnya setiap fitur independen dalam mengklasifikasi fitur target [25]. Metode Random Forest ini menggunakan teknik Ensemble Method yang menggabungkan banyak pohon keputusan yang dilatih dengan subset data yang berbeda, kemudian menggabungkan prediksi dari setiap pohon untuk menghasilkan keputusan akhir [26]. Pemilihan fitur akan dilakukan pada dua versi data, yaitu data yang sudah ternormalisasi dan data yang tidak dilakukan normalisasi.

Pada tahap ini, Random Forest akan mengevaluasi kontribusi relatif setiap fitur dengan menghitung feature importance berdasarkan seberapa besar kontribusi fitur dalam mengurangi ketidakpastian (impurity) dalam pohon keputusan. Fitur dengan nilai importance tinggi dianggap lebih berpengaruh dalam klasifikasi dan akan dipertahankan, sementara fitur dengan nilai importance rendah dapat dihapus untuk menyederhanakan model dan meningkatkan performa. Deteksi feature importance hanya dilakukan pada data latih untuk mencegah kebocoran data (data leakage). Penghapusan fitur yang tidak relevan harus diterapkan secara konsisten pada kedua dataset, baik data latih maupun data uji, agar struktur data tetap selaras selama pelatihan dan evaluasi model [27]. Proses ini membantu mengurangi kompleksitas model dan mencegah overfitting, dan dengan demikian dapat meningkatkan performa model dalam mengklasifikasi data yang belum pernah dilihat sebelumnya.

### E. Normalisasi

Setelah tahap pemilihan fitur, langkah berikutnya adalah melakukan normalisasi data. Normalisasi dilakukan untuk menyelaraskan skala fitur sehingga tidak ada fitur yang mendominasi perhitungan dalam algoritma berbasis jarak seperti K-Nearest Neighbors (KNN). Dalam algoritma ini,

perhitungan jarak, terutama Euclidean Distance, sangat bergantung pada skala data, sehingga perbedaan rentang nilai antar fitur dapat menyebabkan distorsi dalam proses klasifikasi [28]. Oleh karena itu, normalisasi menjadi tahap krusial untuk memastikan setiap fitur memiliki bobot yang seimbang dalam analisis.

Pada penelitian ini, normalisasi diterapkan menggunakan teknik yang relevan dengan perhitungan Euclidean Distance, yaitu Min-Max Scaling, Z-Score Scaling, dan Decimal Scaling. Setiap teknik memiliki karakteristik dan manfaat spesifik dalam mengubah distribusi data agar lebih optimal untuk kinerja model KNN. Penjelasan lebih lanjut mengenai masing-masing teknik akan dijelaskan sebagai berikut:

### 1. Min-Max Scaling

Min-Max Scaling digunakan untuk menormalisasi data dengan mengubah rentang nilai fitur ke dalam skala 0 hingga 1. Proses ini dilakukan dengan merumuskan ulang setiap nilai berdasarkan nilai minimum dan maksimum dalam dataset, sehingga distribusi data tetap terjaga, tetapi dalam skala yang lebih seragam [29]. Metode ini sering dipilih ketika data memiliki nilai yang bervariasi secara signifikan antar fitur, yang dapat memengaruhi performa model. Pada dataset Pima Indians Diabetes, Min-Max Scaling membantu memastikan bahwa setiap fitur akan berada dalam skala yang sama, memungkinkan model untuk memproses data secara lebih efektif tanpa terpengaruh oleh perbedaan skala antar fitur. Rumus Min-Max Scaling adalah sebagai berikut:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

### 2. Z-Score Scaling

Z-Score Scaling atau Standard Scaling digunakan untuk menstandarisasi data dengan menyesuaikan nilai fitur sehingga memiliki rata-rata 0 dan standar deviasi 1. Metode ini menghitung sejauh mana suatu nilai menyimpang dari rata-rata dalam satuan standar deviasi, sehingga memungkinkan perbandingan antar fitur dengan skala yang berbeda [30]. Penggunaan Z-Score Scaling sangat tepat ketika fitur-fitur dalam dataset memiliki distribusi yang berbeda dan perlu distandarisasi untuk memudahkan perbandingan antar fitur. Dengan menggunakan Z-Score Scaling pada dataset Pima Indians Diabetes, nilai-nilai dalam fitur dapat distandarisasi, sehingga memastikan bahwa fitur dengan variabilitas tinggi tidak mendominasi proses analisis. Rumus Z-Score Scaling adalah sebagai berikut:

$$z = \frac{x - \mu}{\sigma}$$

### 3. Decimal Scaling

Decimal Scaling digunakan untuk menormalisasi data dengan membagi setiap nilai dengan pangkat sepuluh yang sesuai, sehingga semua nilai berada dalam rentang yang lebih kecil. Faktor pembagi ditentukan berdasarkan jumlah digit terbesar dalam dataset, menjaga skala data tetap proporsional tanpa mengubah distribusi relatif antar nilai [12]. Decimal

Scaling digunakan untuk mengurangi rentang nilai fitur besar pada data tanpa mempengaruhi distribusi data secara signifikan. Pada dataset Pima Indians Diabetes, metode ini efektif untuk menurunkan rentang nilai fitur besar, sehingga memudahkan model memproses data lebih cepat dan efisien tanpa mengubah distribusi data asli. Meskipun jarang digunakan dalam penelitian terkini, Decimal Scaling tetap efektif untuk menjaga data tetap proporsional dan mudah diproses, terutama pada fitur dengan nilai besar. Rumus Decimal Scaling adalah sebagai berikut:

$$x^* = \frac{x}{10^j}$$

### F. Model KNN (K-Nearest Neighbors)

Setelah proses pemilihan fitur, tahap berikutnya adalah melatih model KNN. Pelatihan model akan dilakukan pada dua versi data, yaitu data yang telah dinormalisasi dengan berbagai metode yang telah diterapkan sebelumnya dan data yang tidak dilakukan normalisasi. Model KNN akan diuji dengan berbagai nilai K, yaitu 1, 3, 5, 7, 9, dan 11, yang dipilih secara manual, untuk mengoptimalkan kinerja model dengan menganalisis performa secara menyeluruh pada setiap kombinasi normalisasi dan nilai K. Dalam algoritma KNN, klasifikasi dilakukan berdasarkan kedekatan suatu data dengan sejumlah K tetangga terdekatnya dalam ruang fitur [5]. Oleh karena itu, pemilihan nilai K menjadi faktor krusial dalam kinerja model. Nilai K yang terlalu kecil dapat menyebabkan model terlalu sensitif terhadap data latih (overfitting), sedangkan nilai K yang terlalu besar dapat menyebabkan model menjadi terlalu umum (underfitting) [31][32]. Selain itu, metode jarak yang digunakan adalah Euclidean Distance, yang merupakan metode default pada KNN. Euclidean Distance adalah jarak terpendek antara dua titik dalam ruang multidimensi, yang dihitung menggunakan rumus akar kuadrat dari jumlah kuadrat selisih nilai pada setiap dimensi. Semakin kecil nilai Euclidean Distance, maka semakin dekat jarak antara kedua titik tersebut [6].

### G. Evaluasi

Pada tahap evaluasi, kinerja model KNN yang telah dilatih akan diuji menggunakan berbagai metrik evaluasi guna menilai efektivitasnya dalam mengidentifikasi pasien dengan diabetes. Fokus utama evaluasi adalah untuk mengidentifikasi teknik normalisasi yang paling optimal dalam meningkatkan kinerja model klasifikasi, dengan menganalisis berbagai variasi nilai K untuk mengamati dampak dari masing-masing teknik normalisasi terhadap hasil yang diperoleh.

Evaluasi dilakukan dengan mempertimbangkan empat komponen utama dalam analisis klasifikasi, yaitu True Positive (TP), yang menunjukkan jumlah kasus di mana model benar dalam mengklasifikasi pasien mengidap diabetes, True Negative (TN), yang mengindikasikan jumlah kasus ketika model dengan benar mengklasifikasi pasien tidak mengidap diabetes, False Positive (FP), yang terjadi saat model salah mengklasifikasi pasien mengidap diabetes padahal sebenarnya tidak, serta False Negative (FN), yang

terjadi ketika model salah mengklasifikasi pasien tidak mengidap diabetes padahal sebenarnya mengidap [33]. Keempat komponen ini menjadi dasar dalam menghitung berbagai metrik evaluasi yang digunakan untuk mengukur kinerja model KNN secara menyeluruh. Hasil evaluasi ini mencakup berbagai metrik yang digunakan untuk menilai performa model yang akan dijelaskan sebagai berikut [34]:

#### 1. Akurasi

Metrik ini mengukur sejauh mana model dapat melakukan klasifikasi yang benar dibandingkan dengan total klasifikasi. Akurasi memberikan gambaran umum tentang performa model, tetapi kurang berguna pada data yang tidak seimbang. Akurasi dihitung dengan rumus:

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

#### 2. Precision

Precision mengukur akurasi klasifikasi positif yang dilakukan oleh model. Metrik ini digunakan untuk mengetahui seberapa banyak klasifikasi positif yang akurat atau sesuai dengan kondisi yang sebenarnya. Precision dihitung dengan rumus:

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

#### 3. Recall

Recall mengukur kemampuan model dalam menemukan semua kasus positif yang sebenarnya. Metrik ini penting ketika sangat penting untuk mendeteksi sebanyak mungkin kasus positif. Recall dihitung dengan rumus:

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

#### 4. F1-Score

F1-Score adalah rata-rata harmonik antara precision dan recall. Metrik ini penting untuk memberikan gambaran keseimbangan antara kemampuan model dalam mendeteksi kelas positif dan negatif, terutama pada dataset yang tidak seimbang. F1-score dihitung dengan rumus:

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

#### 5. Specificity

Specificity mengukur kemampuan model dalam mengidentifikasi data negatif dengan benar. Metrik ini penting untuk memastikan model tidak salah mengklasifikasikan data negatif sebagai positif. Specificity dihitung dengan rumus:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

#### 6. ROC AUC

ROC-AUC mengukur kemampuan model untuk membedakan antara dua kelas (positif dan negatif) di seluruh rentang nilai threshold yang mungkin. Nilai AUC yang lebih tinggi menunjukkan bahwa model memiliki kemampuan yang

lebih baik dalam membedakan antara kelas positif dan negatif, bahkan ketika threshold berubah.

#### H. Uji Statistik

Uji statistik dilakukan dengan menggunakan metode Paired T-Test untuk menguji perbedaan kinerja antara setiap teknik normalisasi dengan kondisi tanpa normalisasi secara berpasangan. Nilai T-Statistic dihitung berdasarkan selisih performa antara teknik normalisasi dengan tanpa normalisasi, yang kemudian dibandingkan dengan distribusi T. Dari T-Statistic, dihitung P-Value untuk menentukan apakah perbedaan yang diamati terjadi secara kebetulan. Jika P-Value  $\leq 0,05$ , perbedaan performa dianggap signifikan secara statistik, menunjukkan bahwa teknik normalisasi berdampak nyata terhadap performa model [35].

### III. HASIL DAN PEMBAHASAN

Pada bab ini, akan dijelaskan hasil yang diperoleh dari setiap tahapan yang telah dijelaskan pada bagian metode penelitian. Berikut adalah penjelasan mengenai hasil yang didapat dari proses yang telah dilakukan:

#### A. Preprocessing Data

Pada tahap awal preprocessing, dilakukan pemeriksaan terhadap tipe data untuk memastikan bahwa setiap fitur dalam dataset diharuskan memiliki format yang sesuai. Dalam KNN, perhitungan jarak, seperti menggunakan Euclidean Distance, bergantung pada angka yang dapat dihitung secara matematis. Oleh karena itu, jika dataset berisi data non-numerik, algoritma ini tidak dapat melakukan perhitungan jarak yang benar, sehingga dapat mempengaruhi akurasi hasil klasifikasi. Dengan memastikan bahwa semua fitur dalam dataset berbentuk numerik, algoritma KNN dapat bekerja secara optimal dan menghasilkan klasifikasi yang lebih akurat.

TABEL 2  
TIPE DATA PADA FITUR DATASET

Fitur	Tipe Data
Pregnancies	int64
Glucose	int64
BloodPressure	int64
SkinThickness	int64
Insulin	int64
BMI	float64
DiabetesPedigreeFunction	float64
Age	int64
Outcome	int64

Dari Tabel 2, terlihat bahwa sebagian besar fitur memiliki tipe data int64, kecuali BMI dan DiabetesPedigreeFunction yang bertipe float64. Meskipun terdapat perbedaan antara int64 dan float64, keduanya termasuk kategori numerik dan dapat diproses oleh algoritma KNN tanpa konversi tipe data tambahan. Selanjutnya, pada tahap preprocessing, dilakukan pemeriksaan terhadap missing values dan data duplikat. Data dengan missing values akan dihapus untuk memastikan hanya data lengkap yang digunakan dalam analisis. Begitu juga, data

duplicat akan dihapus untuk mencegah pengaruhnya terhadap hasil analisis, yang bisa menyebabkan model terlalu fokus pada pola tertentu atau menghasilkan hasil yang tidak akurat.

TABEL 3  
MISSING VALUES DAN DATA DUPLIKAT PADA FITUR DATASET

Fitur	Missing Values	Data Duplikat
Pregnancies	0	0
Glucose	0	0
BloodPressure	0	0
SkinThickness	0	0
Insulin	0	0
BMI	0	0
DiabetesPedigreeFunction	0	0
Age	0	0
Outcome	0	0

Dari Tabel 3, dapat dilihat bahwa tidak ada fitur yang memiliki missing values, dan juga tidak ditemukan data duplikat pada dataset ini. Hal ini mengindikasikan bahwa dataset telah diproses dengan baik sebelumnya. Dengan demikian, dataset sudah dalam kondisi yang optimal, dengan data yang lengkap, akurat, dan unik. Oleh karena itu, tidak perlu dilakukan penghapusan data atau penanganan lebih lanjut terkait missing values dan data duplikat. Langkah selanjutnya dalam preprocessing adalah mengklasifikasikan fitur dalam dataset berdasarkan jenis datanya, yaitu fitur numerik dan kategorikal. Fitur numerik mencakup variabel yang memiliki nilai numerik yang dapat dihitung dan digunakan dalam perhitungan matematis. Sementara itu, fitur kategorikal berisi variabel yang mengelompokkan data ke

dalam kategori atau kelas tertentu, yang bersifat diskrit dan tidak memiliki makna kuantitatif.

TABEL 4  
KLASIFIKASI FITUR BERDASARKAN JENIS DATA

Jenis Data	Fitur
Numerik	Pregnancies
	Glucose
	BloodPressure
	SkinThickness
	Insulin
	BMI
	DiabetesPedigreeFunction
	Age
Kategorikal	Outcome

Dari Tabel 4, kita dapat melihat bahwa seluruh fitur kecuali Outcome termasuk dalam kategori numerik. Outcome dianggap sebagai fitur kategorikal karena berisi informasi kelas atau hasil diagnosis diabetes. Setelah pengklasifikasian ini, selanjutnya akan dilakukan analisis lebih lanjut terhadap fitur numerik, seperti memahami distribusi dan rentang nilainya. Untuk fitur kategorikal, langkah berikutnya adalah memeriksa nilai unik yang terdapat dalam fitur tersebut untuk mengetahui karakteristik tipe data. Analisis tentang fitur numerik dan kategorikal akan dijelaskan sebagai berikut:

#### 1. Fitur Numerik

Distribusi dan rentang nilai fitur numerik dilakukan pemeriksaan menggunakan metode statistik deskriptif. Statistik ini memberikan informasi mengenai rata-rata (mean), standar deviasi (std), nilai minimum (min), kuartil (25%, 50%, 75%), dan nilai maksimum (max) untuk setiap fitur numerik.

TABEL 5  
STATISTIK DESKRIPТИF PADA FITUR NUMERIK

Fitur	Count	Mean	Std	Min	25%	50%	75%	Max
Pregnancies	768	3.845	3.37	0	1	3	6	17
Glucose	768	120.895	31.973	0	99	117	140.25	199
BloodPressure	768	69.105	19.356	0	62	72	80	122
SkinThickness	768	20.536	15.952	0	0	23	40	99
Insulin	768	79.799	115.244	0	0	30.5	127.25	846
BMI	768	31.993	7.884	0	27.3	32	36.6	67.1
DiabetesPedigreeFunction	768	0.472	0.331	0.078	0.244	0.372	0.626	2.42
Age	768	33.241	11.76	21	24	29	41	81

Berdasarkan analisis statistik deskriptif pada fitur numerik yang ditampilkan pada Tabel 5, terlihat adanya variasi signifikan dalam rentang nilai beberapa fitur. Misalnya, fitur Glucose memiliki nilai antara 0 hingga 199, dengan rata-rata 120.90 dan deviasi standar 31.97. Fitur Age memiliki rentang 21 hingga 81, dengan rata-rata 33.24 dan deviasi standar 11.76. Sementara itu, fitur Pregnancies memiliki nilai maksimum 17 dan rata-rata 3.85, sedangkan Insulin menunjukkan deviasi standar tinggi, mencapai 115.24, yang mencerminkan variasi besar dalam data. Fitur seperti

SkinThickness memiliki nilai minimum 0 dan rata-rata cukup rendah (20.54), yang dapat mempengaruhi model karena perbedaan skala antar fitur. BMI juga menunjukkan variabilitas besar, dengan rentang 0 hingga 67.1, rata-rata 31.99, dan deviasi standar 7.88. Perbedaan skala antar fitur dapat mempengaruhi kinerja model, sehingga fitur dengan rentang nilai yang lebih besar cenderung mendominasi proses pembelajaran model. Oleh karena itu normalisasi diperlukan untuk menyelaraskan skala dan memastikan setiap fitur berkontribusi secara seimbang dalam model.

## 2. Fitur Kategorikal

Untuk fitur kategorikal, dilakukan pemeriksaan terhadap nilai unik yang ada. Dapat dilihat pada Tabel 6 menunjukkan bahwa fitur Outcome hanya memiliki dua nilai unik, yaitu [1, 0].

TABEL 6  
NILAI UNIK PADA FITUR KATEGORIKAL

Fitur	Nilai Unik
Outcome	1
	0

Fitur Outcome ini merupakan fitur target dalam dataset yang mengindikasikan apakah seorang pasien mengidap penyakit diabetes (1) atau tidak (0). Fitur ini dapat dikategorikan sebagai fitur kategorikal biner karena hanya terdapat dua nilai unik yang terkandung di dalamnya. Berdasarkan analisis tersebut, maka fitur Outcome tidak memerlukan normalisasi.

## B. Pembagian Data

Pada tahap ini, akan dilakukan pembagian dataset menjadi dua bagian, yaitu data latih dan data uji. Pembagian ini dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji. Data latih digunakan untuk melatih model, sementara data uji digunakan untuk menguji performa model setelah dilatih. Dataset yang telah dibagi dapat dilihat pada Tabel 7. Data latih terdiri dari 614 baris dan 9 kolom, sementara data uji terdiri dari 154 baris dan 9 kolom. Dengan demikian, dataset sudah siap untuk dilakukan pemilihan fitur.

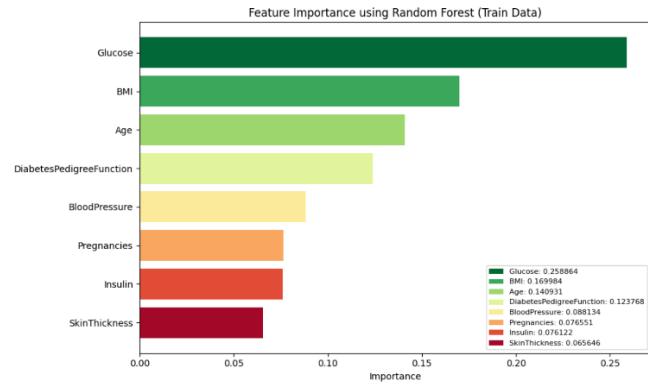
TABEL 7  
PEMBAGIAN DATASET LATIH DAN UJI

Data	Jumlah Baris	Jumlah Kolom
Data Latih	614	9
Data Uji	154	9

## C. Pemilihan Fitur

Setelah membagi dataset menjadi data latih dan data uji, tahap berikutnya adalah melakukan pemilihan fitur untuk mengidentifikasi fitur independen yang memiliki kontribusi besar dalam mengklasifikasi fitur target, yaitu kemungkinan seseorang mengidap diabetes. Pemilihan fitur dilakukan dengan metode Random Forest, yang menggunakan teknik Ensemble Method untuk menggabungkan beberapa pohon keputusan guna meningkatkan stabilitas dan akurasi model. Metode ini memungkinkan pengukuran kontribusi masing-masing fitur terhadap hasil klasifikasi.

Proses deteksi feature importance hanya dilakukan pada fitur numerik, karena fitur kategorikal seperti Outcome sudah merupakan hasil klasifikasi dan tidak relevan untuk diukur menggunakan metode ini. Feature importance dihitung berdasarkan data latih, karena model dilatih menggunakan data tersebut. Penghapusan fitur yang tidak relevan harus diterapkan secara konsisten pada kedua dataset agar model yang dievaluasi tidak terpengaruh oleh fitur yang tidak memberikan kontribusi signifikan.



Gambar 2. Feature Importance Plot

Berdasarkan analisis feature importance yang ditunjukkan dalam Gambar 2, beberapa fitur memiliki kontribusi yang lebih besar dibandingkan yang lain. Fitur Glucose, BMI, Age, dan DiabetesPedigreeFunction memiliki nilai penting yang jauh lebih tinggi dibandingkan fitur lainnya. Sebaliknya, fitur-fitur seperti SkinThickness, Insulin, Pregnancies, dan BloodPressure menunjukkan nilai penting yang relatif kecil. Misalnya, SkinThickness (0.065646) dan Insulin (0.076122) memiliki nilai penting yang sangat rendah, yang menunjukkan bahwa keduanya memberikan kontribusi yang terbatas terhadap klasifikasi. Fitur Pregnancies (0.076551) dan BloodPressure (0.088134) juga menunjukkan nilai penting yang rendah, meskipun sedikit lebih tinggi, namun tidak cukup signifikan untuk dipertahankan dalam model.

Sebagian fitur seperti SkinThickness, Insulin, Pregnancies, dan BloodPressure diputuskan untuk dihapus karena memiliki kontribusi yang sangat kecil terhadap model. Menghapus fitur-fitur ini akan meningkatkan efisiensi model, mengurangi risiko overfitting, dan memastikan model hanya berfokus pada fitur yang lebih relevan terhadap klasifikasi diabetes, seperti Glucose, BMI, DiabetesPedigreeFunction, dan Age. Setelah fitur dihapus dari data latih, langkah yang sama harus diterapkan pada data uji untuk evaluasi model yang konsisten dan menghasilkan performa yang lebih optimal. Hasil dari pemilihan fitur ini dapat dilihat pada Tabel 8.

TABEL 8  
FITUR YANG TERSISA SETELAH PEMILIHAN FITUR

Jenis Data	Fitur
Numerik	Glucose
	BMI
	DiabetesPedigreeFunction
	Age
Kategorikal	Outcome

## D. Normalisasi

Setelah pembagian data menjadi data latih dan data uji, data akan dilakukan normalisasi. Data asli dapat dilihat pada Tabel 9 dan 10 yang memberikan gambaran tentang distribusi dan rentang nilai pada setiap fitur dalam data latih dan data uji sebelum proses normalisasi dilakukan. Dengan melihat data asli ini, dapat lebih jelas dipahami adanya perbedaan skala yang signifikan antar fitur.

TABEL 9  
DATA LATIH SEBELUM DILAKUKAN NORMALISASI

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	84	0	0.304	21	0
2	112	28.2	1.282	50	1
3	139	28.7	0.654	22	0
4	161	21.9	0.254	65	0
5	134	46.2	0.238	46	1
...	...	...	...	...	...
614	125	22.5	0.262	21	0

TABEL 10  
DATA UJI SEBELUM DILAKUKAN NORMALISASI

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	98	34	0.43	43	0
2	112	35.7	0.148	21	0
3	108	30.8	0.158	21	0
4	107	24.6	0.856	34	0
5	136	29.9	0.21	50	0
...	...	...	...	...	...
154	74	35.3	0.705	39	0

Berdasarkan analisis statistik deskriptif pada fitur numerik yang disajikan dalam Tabel 5, setiap fitur menunjukkan rentang nilai yang bervariasi. Beberapa fitur memiliki nilai minimum nol, sementara yang lain memiliki rentang yang jauh lebih besar. Perbedaan skala ini dapat memengaruhi proses pembelajaran model karena fitur dengan nilai lebih besar cenderung mendominasi, maka normalisasi diperlukan untuk menyamakan skala dan rentang nilai antar fitur. Selain itu, berdasarkan pemeriksaan nilai unik pada fitur kategorikal di Tabel 6, fitur Outcome hanya memiliki dua nilai, yaitu 0 dan 1. Karena hanya memiliki dua kelas, fitur ini termasuk dalam kategori biner sehingga tidak memerlukan normalisasi. Dengan demikian, normalisasi akan diterapkan pada semua fitur kecuali Outcome. Proses normalisasi akan diterapkan pada data latih dan data uji. Selanjutnya, akan dibahas lebih lanjut mengenai teknik normalisasi yang digunakan:

### 1. Min-Max Scaling

Min-Max Scaling menormalisasi data dengan mengubah nilai fitur sehingga berada dalam rentang yang seragam, yaitu antara 0 hingga 1. Proses ini dilakukan dengan menghitung nilai minimum dan maksimum dari data latih, kemudian menggunakan nilai tersebut untuk menyesuaikan skala fitur ke rentang tertentu, seperti 0 hingga 1. Dengan demikian, fitur yang memiliki rentang nilai besar akan diperkecil, sementara fitur dengan rentang nilai kecil akan diperbesar, tetapi tetap mempertahankan proporsi antar nilai. Pada data uji, transformasi dilakukan menggunakan nilai minimum dan maksimum yang telah dihitung dari data latih tanpa melakukan perhitungan ulang. Hasil normalisasi dari Min-Max Scaling dapat dilihat pada Tabel 11 untuk data latih dan Tabel 12 untuk data uji.

TABEL 11  
DATA LATIH SETELAH DILAKUKAN MIN-MAX SCALING

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	0.422110553	0	0.096498719	0	0
2	0.56281407	0.420268256	0.514090521	0.483333333	1
3	0.698492462	0.427719821	0.245943638	0.016666667	0
4	0.809045226	0.326378539	0.075149445	0.733333333	0
5	0.673366834	0.68852459	0.068317677	0.416666667	1
...	...	...	...	...	...
614	0.628140704	0.335320417	0.078565329	0	0

TABEL 12  
DATA UJI SETELAH DILAKUKAN MIN-MAX SCALING

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	0.492462312	0.506706408	0.15029889	0.366666667	0
2	0.56281407	0.532041729	0.029888984	0	0
3	0.542713568	0.459016393	0.034158839	0	0
4	0.537688442	0.36661699	0.332194705	0.216666667	0
5	0.683417085	0.445603577	0.056362084	0.483333333	0

...	...	...	...	...	...
154	0.371859296	0.526080477	0.267719898	0.3	0

## 2. Z-Score Scaling

Z-Score Scaling menormalisasi data dengan cara menstandarisasi setiap fitur sehingga memiliki rata-rata 0 dan deviasi standar 1. Proses Z-Score Scaling ini menggunakan nilai mean dan standar deviasi yang dihitung dari data latih untuk menstandarkan fitur sehingga memiliki distribusi dengan mean 0 dan standar deviasi 1. Parameter ini kemudian diterapkan pada data

uji tanpa menghitung ulang statistik baru. Dengan cara ini, data pada data uji akan disesuaikan skala fiturnya menggunakan parameter yang diperoleh dari data latih untuk memastikan konsistensi pada kedua dataset, sehingga distribusi data pada data uji menjadi serupa dengan data latih. Hasil normalisasi dari Z-Score Scaling dapat dilihat pada Tabel 13 untuk data latih dan Tabel 14 untuk data uji.

TABEL 13  
DATA LATIH SETELAH DILAKUKAN Z-SCORE SCALING

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	-1.151397924	-4.135255779	-0.49073479	-1.035940379	0
2	-0.276642826	-0.489168806	2.41502991	1.487100846	1
3	0.566871018	-0.424521874	0.549160552	-0.948938958	0
4	1.254178595	-1.303720151	-0.639291267	2.792122169	0
5	0.410664751	1.838120751	-0.68682934	1.139095159	1
...	...	...	...	...	...
614	0.129493469	-1.226143832	-0.615522231	-1.035940379	0

TABEL 14  
DATA UJI SETELAH DILAKUKAN Z-SCORE SCALING

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	-0.714020375	0.260735607	-0.116372467	0.878090895	0
2	-0.276642826	0.480535176	-0.954231	-1.035940379	0
3	-0.40160784	-0.153004759	-0.924519704	-1.035940379	0
4	-0.432849094	-0.954626717	1.149328721	0.095078101	0
5	0.473147258	-0.269369236	-0.770020968	1.487100846	0
...	...	...	...	...	...
154	-1.463810459	0.428817631	0.700688159	0.530085208	0

## 3. Decimal Scaling

Decimal Scaling menyesuaikan skala fitur dengan membagi nilai setiap fitur dengan pangkat sepuluh yang sesuai. Proses ini dilakukan berdasarkan jumlah digit terbesar dalam dataset. Karena skala ini ditentukan oleh distribusi keseluruhan data, jika hanya dihitung dari data latih, maka distribusi data uji bisa berbeda sehingga skala menjadi tidak konsisten. Pembagian ini bertujuan agar

selisih rentang nilai dalam dataset tidak terlalu besar atau terlalu kecil, tetapi tetap mempertahankan proporsi relatif antar data. Teknik ini sederhana karena hanya melibatkan pembagian dengan angka tetap, sehingga distribusi data tetap terjaga dalam rentang lebih kecil dan mudah diterapkan pada dataset dengan nilai yang bervariasi. Hasil normalisasi dari Decimal Scaling dapat dilihat pada Tabel 15 untuk data latih dan Tabel 16 untuk data uji.

TABEL 15  
DATA LATIH SETELAH DILAKUKAN DECIMAL SCALING

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	0.084	0	0.0304	0.21	0
2	0.112	0.282	0.1282	0.5	1
3	0.139	0.287	0.0654	0.22	0
4	0.161	0.219	0.0254	0.65	0
5	0.134	0.462	0.0238	0.46	1
...	...	...	...	...	...
614	0.125	0.225	0.0262	0.21	0

TABEL 16  
DATA UJI SETELAH DILAKUKAN DECIMAL SCALING

No	Glucose	BMI	DiabetesPedigreeFunction	Age	Outcome
1	0.098	0.34	0.043	0.43	0

2	0.112	0.357	0.0148	0.21	0
3	0.108	0.308	0.0158	0.21	0
4	0.107	0.246	0.0856	0.34	0
5	0.136	0.299	0.021	0.5	0
...	...	...	...	...	...
154	0.074	0.353	0.0705	0.39	0

#### E. Model KNN (K-Nearest Neighbors)

Setelah tahap pemilihan fitur, proses selanjutnya adalah pelatihan model K-Nearest Neighbors (KNN) menggunakan dua skenario data, yakni data yang telah melalui proses normalisasi dengan berbagai metode dan data yang tidak dilakukan normalisasi. Model dilatih dengan berbagai nilai parameter K, yaitu 1, 3, 5, 7, 9, dan 11, yang dipilih secara manual, untuk mengidentifikasi pengaruh jumlah tetangga terdekat terhadap proses klasifikasi. Variasi nilai K ini digunakan untuk mengamati serta memahami respons model terhadap perbedaan skala fitur yang digunakan. Selain itu, metode jarak yang digunakan adalah Euclidean Distance, yang merupakan metode default pada KNN untuk mengukur kedekatan antar data.

#### F. Evaluasi

Setelah proses pelatihan selesai, evaluasi kinerja dilakukan terhadap setiap model KNN yang telah dilatih. Evaluasi menggunakan enam metrik utama, yaitu akurasi, precision, recall, F1-score, specificity, dan ROC AUC. Metrik tersebut dipilih karena memiliki kemampuan untuk memberikan gambaran yang komprehensif mengenai kinerja model dalam membedakan kelas target secara akurat, baik pada kelas positif maupun kelas negatif. Pengujian dilakukan pada seluruh nilai parameter K untuk menilai sensitivitas dan konsistensi performa model terhadap perubahan jumlah tetangga. Hasil evaluasi dari seluruh kombinasi parameter K dan teknik normalisasi disajikan pada Tabel 17 untuk dilakukan analisis perbandingan.

TABEL 17  
HASIL EVALUASI PERFORMA MODEL KNN BERDASARKAN TEKNIK NORMALISASI DAN NILAI K

Teknik Normalisasi	Nilai K	Accuracy	Precision	Recall	F1-Score	Specificity	ROC AUC
Tanpa Normalisasi	1	0.7273	0.7030	0.7030	0.7030	0.7879	0.7030
	3	0.7403	0.7179	0.7212	0.7194	0.7879	0.7587
	5	0.7403	0.7188	0.7253	0.7215	0.7778	0.7719
	7	0.7273	0.7065	0.7152	0.7096	0.7576	0.7707
	9	0.7532	0.7318	0.7354	0.7334	0.7980	0.7751
	11	0.7597	0.7390	0.7444	0.7414	0.7980	0.7802
Min-Max Scaling	1	0.7403	0.7179	0.7212	0.7194	0.7879	0.7212
	3	0.7792	0.7596	0.7636	0.7615	0.8182	0.7844
	5	0.7857	0.7666	0.7687	0.7676	0.8283	0.7910
	7	0.8117	0.7946	0.7970	0.7958	0.8485	0.7950
	9	0.7987	0.7811	0.7788	0.7799	0.8485	0.8037
	11	0.7792	0.7599	0.7556	0.7576	0.8384	0.8050
Z-Score Scaling	1	0.7403	0.7172	0.7172	0.7172	0.7980	0.7172
	3	0.7403	0.7188	0.7253	0.7215	0.7778	0.7553
	5	0.7468	0.7252	0.7303	0.7274	0.7879	0.7697
	7	0.7922	0.7737	0.7737	0.7737	0.8384	0.7969
	9	0.7792	0.7599	0.7556	0.7576	0.8384	0.8060
	11	0.7727	0.7529	0.7586	0.7554	0.8081	0.8129
Decimal Scaling	1	0.7208	0.6964	0.6980	0.6972	0.7778	0.6980
	3	0.7013	0.6813	0.6909	0.6840	0.7273	0.7391
	5	0.7208	0.6976	0.7020	0.6995	0.7677	0.7533
	7	0.7338	0.7114	0.7162	0.7135	0.7778	0.7728
	9	0.7403	0.7172	0.7172	0.7172	0.7980	0.7834
	11	0.7273	0.7024	0.6990	0.7006	0.7980	0.7874

Pada model K-Nearest Neighbors (KNN) tanpa normalisasi, performa model cenderung kurang optimal dibanding skenario lainnya. Akurasi tertinggi hanya sebesar 0.7597 pada K = 11, tanpa menunjukkan tren konsisten seiring meningkatnya nilai K. Misalnya, akurasi turun ke 0.7273 pada K = 7, sama seperti saat K = 1. Ini menandakan bahwa tanpa penyamaan skala fitur, model kesulitan menghitung jarak dengan akurat, yang berdampak pada

kemampuan klasifikasinya. Precision tertinggi hanya mencapai 0.7390 dan fluktuatif di beberapa nilai K, mengindikasikan efisiensi klasifikasi kelas positif yang belum optimal. Recall juga tergolong rendah, dengan nilai maksimal 0.7444, menunjukkan banyak data positif yang tak terdeteksi. F1-Score tertinggi berada di angka 0.7414 dengan tren performa yang stagnan. Specificity terbaik tercatat sebesar 0.7980 pada K = 9 dan 11, namun peningkatannya tidak

signifikan. ROC AUC maksimal 0.7802, menandakan keterbatasan model dalam membedakan kelas positif dan negatif. Secara keseluruhan, tanpa normalisasi, KNN kesulitan memproses jarak secara seimbang ketika fitur memiliki skala berbeda, yang akhirnya menurunkan performa klasifikasi.

Normalisasi dengan Min-Max Scaling memberikan peningkatan performa yang signifikan pada seluruh metrik evaluasi KNN. Akurasi tertinggi tercapai pada  $K = 7$  dengan nilai 0.8117, tertinggi di antara semua metode yang diuji. Precision meningkat konsisten hingga 0.7946, menunjukkan klasifikasi positif yang lebih tepat dan penurunan false positive. Recall juga membaik secara signifikan, mencapai 0.7970, menandakan sensitivitas model yang lebih baik terhadap data positif. F1-Score tertinggi berada di angka 0.7958, mencerminkan keseimbangan precision dan recall yang optimal. Specificity pun sangat baik, dengan nilai maksimal 0.8485, menunjukkan kemampuan identifikasi data negatif yang akurat. ROC AUC mencapai 0.8050, memperlihatkan kekuatan model dalam membedakan dua kelas secara menyeluruh. Keunggulan Min-Max Scaling terletak pada kemampuannya menyamakan rentang fitur ke skala 0–1, membuat semua fitur berkontribusi seimbang dalam perhitungan jarak. Alhasil, model menjadi lebih stabil, akurat, dan andal dalam menghadapi data multivariat dengan skala berbeda.

Normalisasi dengan Z-Score Scaling juga menghasilkan performa KNN yang kompetitif, meskipun beberapa metrik evaluasi masih sedikit di bawah Min-Max Scaling. Akurasi tertinggi tercapai pada  $K = 7$  sebesar 0.7922, dengan pola stabil di berbagai nilai  $K$ . Precision maksimal sebesar 0.7737 menunjukkan kemampuan model dalam menekan false positive secara konsisten. Recall pun cukup baik dengan nilai tertinggi 0.7586, menandakan sensitivitas yang memadai terhadap data positif. F1-Score tertinggi sebesar 0.7554 menunjukkan keseimbangan yang solid antara precision dan recall. Specificity mencapai 0.8384, memperlihatkan keakuratan model dalam mengenali data negatif. Yang paling menonjol dari metode ini adalah nilai ROC AUC tertinggi, yaitu 0.8129, menunjukkan efektivitas Z-Score Scaling dalam membedakan dua kelas secara global. Z-Score Scaling bekerja dengan mengurangi nilai fitur dengan rata-rata dan membaginya dengan standar deviasi, sehingga menghasilkan distribusi fitur yang setara. Pendekatan ini tak hanya menyamakan skala, tetapi juga mempertahankan struktur distribusi data, sehingga sangat cocok untuk data berdistribusi normal.

Normalisasi dengan Decimal Scaling menunjukkan performa terendah dibanding teknik normalisasi lainnya

TABEL 18  
HASIL UJI T-STATISTIC PADA SETIAP METRIK EVALUASI

Teknik Normalisasi	T-Statistic					
	Accuracy	Precision	Recall	F1-Score	Specificity	ROC AUC
Min-Max Scaling vs Tanpa Normalisasi	-3.99754	-4.12930	-3.54493	-4.02148	-3.60555	-14.30525
Z-Score Scaling vs Tanpa Normalisasi	-2.15584	-2.21749	-2.35725	-2.25157	-1.78377	-2.47726
Decimal Scaling vs Tanpa Normalisasi	2.52689	2.73333	2.92463	2.77965	0.91287	0.84161

maupun tanpa normalisasi. Akurasi tertinggi hanya mencapai 0.7403 pada  $K = 9$ , dengan tren nilai  $K$  yang tidak stabil. Precision dan recall maksimum sama-sama berada di angka 0.7172, menandakan keterbatasan model dalam mengidentifikasi data positif secara akurat dan sensitif. F1-Score juga stagnan di angka 0.7172, menunjukkan keseimbangan precision dan recall yang belum tercapai. Specificity tertinggi sebesar 0.7980, sama seperti kondisi tanpa normalisasi, sehingga tidak ada peningkatan berarti. ROC AUC maksimal 0.7874, masih lebih rendah dibanding Min-Max Scaling dan Z-Score Scaling. Decimal Scaling menggeser titik desimal berdasarkan nilai maksimum fitur, tanpa mempertimbangkan distribusi atau penyebaran nilai, sehingga skala antar fitur tetap tidak seragam. Akibatnya, perhitungan jarak dalam KNN menjadi bias dan berdampak negatif pada performa model. Oleh karena itu, Decimal Scaling kurang cocok untuk model berbasis jarak seperti KNN, terutama pada data multivariat dengan rentang nilai yang beragam.

#### G. Uji Statistik

Untuk memastikan apakah perbedaan performa yang diamati signifikan atau hanya terjadi kebetulan, maka selanjutnya akan dilakukan uji statistik. Uji statistik dilakukan dengan menggunakan metode Paired T-Test. Metode ini dipilih karena dianggap tepat untuk menguji perbedaan kinerja antara masing-masing teknik normalisasi dengan kondisi tanpa normalisasi secara berpasangan. Metode Paired T-Test memungkinkan untuk melakukan evaluasi apakah terdapat perbedaan signifikan dalam metrik evaluasi utama seperti Accuracy, Precision, Recall, F1-Score, Specificity, dan ROC AUC, baik dalam peningkatan maupun penurunan performa model.

Dalam pengujian ini, nilai T-Statistic dihitung berdasarkan selisih performa antara teknik normalisasi dengan tanpa normalisasi, yang kemudian dibandingkan dengan distribusi T untuk menentukan apakah perbedaan tersebut cukup besar dibandingkan variasi alami dalam data. P-Value dihitung dari nilai T-Statistic, yang merepresentasikan probabilitas bahwa perbedaan yang diamati terjadi secara kebetulan, jika hipotesis nol benar. Jika  $P\text{-Value} \leq 0,05$ , maka perbedaan performa dianggap signifikan secara statistik, artinya teknik normalisasi memang berdampak nyata terhadap performa KNN. Hasil uji statistik ini dapat dilihat secara lengkap pada Tabel 18 dan Tabel 19 yang menyajikan nilai T-Statistic dan P-Value untuk setiap perbandingan.

TABEL 19  
HASIL UJI P-VALUE PADA SETIAP METRIK EVALUASI

Teknik Normalisasi	P-Value					
	Accuracy	Precision	Recall	F1-Score	Specificity	ROC AUC
Min-Max Scaling vs Tanpa Normalisasi	0,01035	0,00909	0,01648	0,01011	0,01545	0,00003
Z-Score Scaling vs Tanpa Normalisasi	0,08363	0,07737	0,06497	0,07413	0,13454	0,05603
Decimal Scaling vs Tanpa Normalisasi	0,05273	0,04112	0,03284	0,03892	0,40318	0,43839

Pada metrik Accuracy, hasil uji menunjukkan bahwa Min-Max Scaling memberikan peningkatan performa yang signifikan secara statistik dibandingkan Tanpa Normalisasi. Hal ini dibuktikan oleh P-Value sebesar 0,01035, yang berada di bawah ambang signifikansi 0,05. Artinya, peningkatan akurasi dari Min-Max Scaling bukan sekadar kebetulan, melainkan signifikan secara matematis. Sebaliknya, Z-Score Scaling menghasilkan P-Value sebesar 0,08363 dan Decimal Scaling sebesar 0,05273. Keduanya melebihi ambang batas signifikansi, sehingga peningkatan performa dari metode tersebut belum bisa dianggap signifikan secara statistik pada metrik Accuracy.

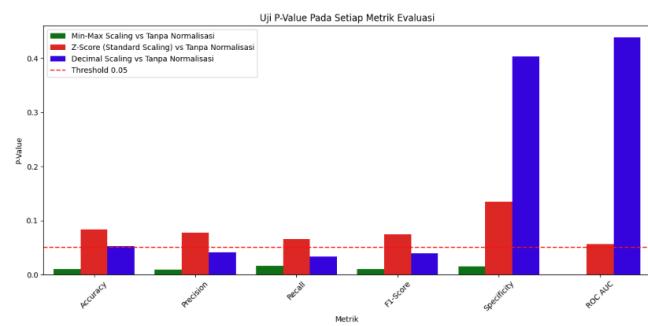
Pada metrik Precision, hasil uji kembali menunjukkan konsistensi. Min-Max Scaling menghasilkan perbedaan signifikan dengan P-Value sebesar 0,00909, menandakan bahwa model ini secara nyata lebih presisi dalam mengklasifikasikan kelas positif dibandingkan tanpa normalisasi. Z-Score Scaling masih belum signifikan dengan P-Value sebesar 0,07737, yang melebihi ambang 0,05. Menariknya, Decimal Scaling justru menunjukkan signifikansi pada metrik Precision dengan P-Value sebesar 0,04112. Ini mengindikasikan bahwa meskipun performa rata-ratanya tidak unggul, dampak Decimal Scaling terhadap precision terdeteksi berbeda secara statistik dari kondisi Tanpa Normalisasi.

Pada metrik Recall, Min-Max Scaling kembali menunjukkan signifikansi dengan P-Value sebesar 0,01648, menandakan peningkatan sensitivitas terhadap kelas positif yang signifikan secara statistik. Z-Score Scaling tetap belum signifikan, dengan P-Value sebesar 0,06497 yang masih di atas ambang 0,05. Menariknya, Decimal Scaling menunjukkan hasil signifikan dengan P-Value sebesar 0,03284. Meski peningkatan absolutnya lebih rendah dibanding Min-Max Scaling, hasil ini menunjukkan bahwa Decimal Scaling tetap berpengaruh dalam meningkatkan recall secara statistik.

Evaluasi pada metrik F1-Score, yang mencerminkan keseimbangan antara precision dan recall, menunjukkan pola serupa. Min-Max Scaling kembali signifikan dengan P-Value sebesar 0,01011. Decimal Scaling juga menunjukkan perbedaan signifikan dengan P-Value sebesar 0,03892. Sebaliknya, Z-Score Scaling tetap tidak signifikan karena P-Value sebesar 0,07413 masih di atas ambang 0,05. Temuan ini memperkuat bahwa Decimal Scaling memang memberikan perubahan statistik pada performa model, meskipun tidak menghasilkan peningkatan substansial dalam akurasi secara keseluruhan.

Pada metrik Specificity, yang mengukur kemampuan model dalam mengklasifikasikan kelas negatif dengan benar, Min-Max Scaling kembali menunjukkan hasil signifikan dengan P-Value sebesar 0,01545. Ini menandakan bahwa metode ini efektif dalam menekan false positive secara nyata. Sementara itu, Z-Score Scaling dan Decimal Scaling mencatat P-Value sebesar 0,13454 dan 0,40318, yang tidak signifikan secara statistik. Dengan demikian, dapat disimpulkan bahwa kedua metode tersebut tidak memberikan peningkatan nyata dalam mengklasifikasikan kelas negatif.

Pada metrik ROC AUC, yang mengukur kemampuan model dalam membedakan kelas positif dan negatif secara keseluruhan, Min-Max Scaling menunjukkan hasil sangat signifikan dengan P-Value sebesar 0,00003. Nilai ini mengindikasikan bukti yang sangat kuat bahwa metode ini secara nyata meningkatkan performa klasifikasi global. Z-Score Scaling mendekati signifikansi dengan P-Value sebesar 0,05603, namun tetap sedikit di atas ambang batas. Sementara itu, Decimal Scaling mencatat P-Value jauh lebih tinggi, yaitu 0,43839, menunjukkan bahwa metode ini tidak berdampak signifikan terhadap kemampuan diskriminatif model.



Gambar 3. Visualisasi P-Value pada Setiap Metrik Evaluasi

Berdasarkan hasil pengujian statistik dan visualisasi P-Value pada Gambar 3, Min-Max Scaling terbukti sebagai teknik normalisasi yang paling konsisten dan unggul secara statistik. Metode ini menunjukkan perbedaan signifikan pada setiap metrik evaluasi dengan P-Value di bawah 0,05, yang menandakan peningkatan performa model KNN secara stabil. Sementara itu, Z-Score Scaling meskipun cukup kompetitif secara rata-rata, tidak menghasilkan signifikansi statistik pada metrik apa pun, kemungkinan akibat variabilitas hasil atau ukuran sampel yang terbatas. Di sisi lain, Decimal Scaling memang signifikan pada beberapa metrik seperti Precision, Recall, dan F1-Score, namun performa absolutnya masih rendah dan tidak merata. Dengan demikian, Min-Max Scaling dapat direkomendasikan sebagai teknik normalisasi terbaik

untuk model KNN pada dataset Pima Indians Diabetes, terutama jika tujuannya adalah meningkatkan performa secara valid dan konsisten di berbagai metrik evaluasi.

#### IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa normalisasi data memainkan peran penting dalam meningkatkan performa model K-Nearest Neighbors (KNN). Penerapan teknik normalisasi pada data latih dan uji menunjukkan adanya peningkatan yang signifikan, terutama ketika menggunakan Min-Max Scaling.

Min-Max Scaling terbukti memberikan dampak positif terhadap semua metrik evaluasi, dengan P-Value yang konsisten di bawah ambang 0,05. Akurasi tertinggi yang tercatat adalah 0.8117 pada nilai K = 7, dan ROC AUC yang mencapai 0.8050 menunjukkan kemampuan model yang lebih baik dalam membedakan dua kelas secara keseluruhan. Sebaliknya, meskipun Z-Score Scaling menunjukkan hasil yang cukup baik pada beberapa metrik evaluasi, teknik ini tidak memberikan peningkatan yang signifikan secara statistik pada semua metrik evaluasi. Dengan P-Value yang lebih tinggi dari 0,05 di semua metrik, Z-Score Scaling menunjukkan bahwa meskipun dapat menormalkan distribusi data, dampaknya tidak konsisten di seluruh metrik evaluasi. Sementara itu, Decimal Scaling memberikan hasil yang paling rendah jika dibandingkan dengan kedua metode lainnya. Meskipun pada beberapa metrik evaluasi, seperti Precision, Recall dan F1-Score, terdapat peningkatan yang signifikan dengan P-Value di bawah 0,05, performa keseluruhan tetap lebih rendah dibandingkan dengan Min-Max Scaling dan Z-Score. Selain itu, fluktuasi pada nilai K juga menunjukkan ketidakstabilan dalam model.

Secara keseluruhan, dapat disimpulkan bahwa Min-Max Scaling merupakan teknik normalisasi yang paling efektif untuk model K-Nearest Neighbors (KNN) pada dataset Pima Indians Diabetes. Hal ini dikarenakan metode tersebut secara konsisten menghasilkan performa terbaik dalam setiap metrik evaluasi yang dilakukan, serta terbukti memiliki signifikansi statistik yang kuat. Oleh karena itu, untuk meningkatkan kinerja model KNN dalam klasifikasi data multivariat, khususnya dalam konteks prediksi diabetes, Min-Max Scaling dapat direkomendasikan sebagai metode normalisasi yang paling optimal.

#### DAFTAR PUSTAKA

- [1] I. W. Suryasa, M. Rodríguez-Gámez, and T. Koldoris, "Health and Treatment of Diabetes Mellitus," *Int J Health Sci (Qassim)*, vol. 5, no. 1, pp. I–V, 2021, doi: 10.53730/IJHS.V5N1.2864.
- [2] L. Ryden, G. Ferrannini, and E. Standl, "Risk prediction in patients with diabetes: is SCORE 2D the perfect solution?," Jul. 21, 2023, *Oxford University Press*. doi: 10.1093/eurheartj/ehad263.
- [3] S. Alam, M. K. Hasan, S. Neaz, N. Hussain, M. F. Hossain, and T. Rahman, "Diabetes Mellitus: Insights from Epidemiology, Biochemistry, Risk Factors, Diagnosis, Complications and Comprehensive Management," Jun. 01, 2021, *MDPI*. doi: 10.3390/diabetology2020004.
- [4] S. Templer, S. Abdo, and T. Wong, "Preventing diabetes complications," *Intern Med J*, vol. 54, no. 8, pp. 1264–1274, Aug. 2024, doi: 10.1111/imj.16455.
- [5] S. Zhang and J. Li, "KNN Classification With One-Step Computation," *IEEE Trans Knowl Data Eng*, vol. 35, no. 3, pp. 2711–2723, Mar. 2023, doi: 10.1109/TKDE.2021.3119140.
- [6] N. Ukey, Z. Yang, B. Li, G. Zhang, Y. Hu, and W. Zhang, "Survey on Exact kNN Queries over High-Dimensional Data Space," Jan. 01, 2023, *MDPI*. doi: 10.3390/s23020629.
- [7] S. Zhang, "Challenges in KNN Classification," *IEEE Trans Knowl Data Eng*, vol. 34, no. 10, pp. 4663–4675, Oct. 2022, doi: 10.1109/TKDE.2021.3049250.
- [8] M. V. Polyakova and V. N. Krylov, "Data normalization methods to improve the quality of classification in the breast cancer diagnostic system," *Applied Aspects of Information Technology*, vol. 5, no. 1, pp. 55–63, Apr. 2022, doi: 10.15276/aaит.05.2022.5.
- [9] M. Zulkifilu and A. Yasir, "About Some Data Precaution Techniques For K-Means Clustering Algorithm," *UMYU Scientifica*, vol. 1, no. 1, pp. 12–19, 2022, doi: 10.47430/usci.1122.003.
- [10] M. Pagan, M. Zarlis, and A. Candra, "Investigating the impact of data scaling on the k-nearest neighbor algorithm," *Computer Science and Information Technologies*, vol. 4, no. 2, pp. 135–142, Jul. 2023, doi: 10.11591/csit.v4i2.pp135-142.
- [11] A. Alsarhan, F. Hussein, S. Moh, and F. S. El-Salhi, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data (Basel)*, vol. 6, no. 2, 2021, doi: 10.3390/data.
- [12] S. Sinsomboonthong, "Performance Comparison of New Adjusted Min-Max with Decimal Scaling and Statistical Column Normalization Methods for Artificial Neural Network Classification," *Int J Math Math Sci*, vol. 2022, 2022, doi: 10.1155/2022/3584406.
- [13] C. C. Olisah, L. Smith, and M. Smith, "Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective," *Comput Methods Programs Biomed*, vol. 220, Jun. 2022, doi: 10.1016/j.cmpb.2022.106773.
- [14] A. M. Vommi and T. K. Battula, "A hybrid filter-wrapper feature selection using Fuzzy KNN based on Bonferroni mean for medical datasets classification: A COVID-19 case study," *Expert Syst Appl*, vol. 218, May 2023, doi: 10.1016/j.eswa.2023.119612.
- [15] Y. Zhao, "Comparative Analysis of Diabetes Prediction Models Using the Pima Indian Diabetes Database," *ITM Web of Conferences*, vol. 70, p. 02021, Jan. 2025, doi: 10.1051/itmconf/20257002021.
- [16] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," *Neural Comput Appl*, vol. 35, no. 22, pp. 16157–16173, Aug. 2023, doi: 10.1007/s00521-022-07049-z.
- [17] V. Patil and D. R. Ingle, "Comparative Analysis of Different ML Classification Algorithms with Diabetes Prediction through Pima Indian Diabetics Dataset," in *2021 International Conference on Intelligent Technologies, CONIT 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021. doi: 10.1109/CONIT51480.2021.9498361.
- [18] H. Karamti *et al.*, "Improving Prediction of Cervical Cancer Using KNN Imputed SMOTE Features and Multi-Model Ensemble Learning Approach," *Cancers (Basel)*, vol. 15, no. 17, Sep. 2023, doi: 10.3390/cancers15174412.
- [19] M. N. Maskuri, K. Sukerti, and R. M. Herdian Bhakti, "Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Memprediksi Penyakit Stroke Stroke Disease Predict Using KNN Algorithm," *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 4, no. 1, May 2022.
- [20] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," Mar. 29, 2021, *Frontiers Media S.A.* doi: 10.3389/fenrg.2021.652801.
- [21] O. Alotaibi, E. Paredete, and S. Tomy, "Cleaning Big Data Streams: A Systematic Literature Review," Aug. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/technologies11040101.

- [22] M. Arif, maruf Setiawan, A. Dwi Hartono, M. Arif Ma, and ruf Setiawan, "Menggunakan Metode Machine Learning Untuk Memprediksi Nilai Mahasiswa Dengan Model Prediksi Multiclass," *Jurnal Informatika: Jurnal pengembangan IT*, vol. 10, no. 1, p. 2025, 2025, doi: 10.30591/jpit.v9ix.xxx.
- [23] L. A. Demidova, "Two-stage hybrid data classifiers based on svm and knn algorithms," *Symmetry (Basel)*, vol. 13, no. 4, Apr. 2021, doi: 10.3390/sym13040615.
- [24] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," Jun. 27, 2022, *Frontiers Media SA*. doi: 10.3389/fbinf.2022.927312.
- [25] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, "Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method," *Symmetry (Basel)*, vol. 14, no. 6, Jun. 2022, doi: 10.3390/sym14061095.
- [26] A. G. Priya Varshini, K. Anitha Kumari, and V. Varadarajan, "Estimating software development efforts using a random forest-based stacked ensemble approach," *Electronics (Switzerland)*, vol. 10, no. 10, May 2021, doi: 10.3390/electronics10101195.
- [27] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, Dec. 2022, doi: 10.1186/s42400-021-00103-8.
- [28] P. J. Muhammad Ali, "Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements," *ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY*, vol. 10, no. 1, pp. 85–91, Jun. 2022, doi: 10.14500/aro.10955.
- [29] Henderi , T. Wahyuningsih , and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *International Journal of Informatics and Information System*, vol. 4, no. 1, Mar. 2021, [Online]. Available: <http://archive.ics.uci.edu/ml>.
- [30] M. R. Firmansyah and Y. P. Astuti, "Stroke Classification Comparison with KNN through Standardization and Normalization Techniques," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, Jan. 2024, doi: 10.26877/asset.v6i1.17685.
- [31] Emad Majeed Hameed and Hardik Joshi, "Improving Diabetes Prediction by Selecting Optimal K and Distance Measures in KNN Classifier," *Journal of Techniques*, vol. 6, no. 3, pp. 19–25, Aug. 2024, doi: 10.51173/jt.v6i3.2587.
- [32] G. Fatima and S. Saeed, "A Novel Weighted Ensemble Method to Overcome the Impact of Under-fitting and Over-fitting on the Classification Accuracy of the Imbalanced Data Sets," *Pakistan Journal of Statistics and Operation Research*, vol. 17, no. 2, pp. 483–496, 2021, doi: 10.18187/pjsor.v17i2.3640.
- [33] S. Gündoğdu, "Efficient prediction of early-stage diabetes using XGBoost classifier with random forest feature selection technique," *Multimed Tools Appl*, vol. 82, no. 22, pp. 34163–34181, Sep. 2023, doi: 10.1007/s11042-023-15165-8.
- [34] A. S. Maklad, M. A. Mahdy, A. Malki, N. Niki, and A. A. Mohamed, "Advancing Early Detection of Colorectal Adenomatous Polyps via Genetic Data Analysis: A Hybrid Machine Learning Approach," *Journal of Computer and Communications*, vol. 12, no. 07, pp. 23–38, 2024, doi: 10.4236/jcc.2024.127003.
- [35] A. Zweifach, "Samples in many cell-based experiments are matched/paired but taking this into account does not always increase power of statistical tests for differences in means," *Mol Biol Cell*, vol. 35, no. 1, p. br1, Jan. 2024, doi: 10.1091/mbc.E23-05-0159.