Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Comparison of Data Normalization Techniques on KNN Classification Performance on Diabetes Dataset

Yohanes Dimas Pratama [1]*, Abu Salam [2]**
* Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
111202113254@mhs.dinus.ac.id [1], abu.salam@dsn.dinus.ac.id [2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

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

ABSTRACT

This study analyzes the comparison of various data normalization techniques on the performance of the K-Nearest Neighbors (KNN) model in diabetes classification on the Pima Indians Diabetes Database dataset. The three normalization techniques evaluated are Min-Max Scaling, Z-Score Scaling, and Decimal Scaling. Data that has gone through the preprocessing and feature selection stages is then applied to the KNN model to predict the likelihood of someone having diabetes. Evaluation is done using several performance metrics, such as accuracy, precision, recall, F1-Score, specificity, and ROC AUC. The results show that Min-Max Scaling provides significant improvement on all metrics, with the highest accuracy recorded at 0.8117 and ROC AUC reaching 0.8050. Z-Score Scaling also showed good results, but not as good as Min-Max Scaling statistically. Meanwhile, Decimal Scaling showed lower performance compared to the other two methods. Overall, Min-Max Scaling proved to be the most effective normalization method to improve the performance of KNN model in diabetes classification.

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][4]. Diabetes dapat menyebabkan berbagai komplikasi kesehatan serius, seperti kerusakan pada jantung, ginjal, mata, dan sistem saraf [4][5]. Oleh karena itu, deteksi dini dan prediksi risiko diabetes sangat penting untuk mencegah terjadinya komplikasi lebih lanjut. Dalam hal ini, teknologi informasi dan pembelajaran mesin dapat memainkan peran penting dalam mempermudah dan mempercepat diagnosis serta prediksi 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 [6]. Meskipun KNN sederhana dan mudah diimplementasikan, namun ada salah satu tantangan utama dalam menggunakan algoritma ini adalah sensitifitasnya terhadap data yang belum dinormalisasi. KNN mengandalkan perhitungan jarak, seperti Euclidean, untuk mengukur kedekatan antar data [7]. Jika data memiliki skala atau satuan yang berbeda-beda, hal ini dapat menyebabkan ketidakseimbangan dalam perhitungan jarak, yang pada gilirannya dapat menurunkan kinerja model [8]. 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 oleh perbedaan skala [9]. Terdapat tiga teknik normalisasi yang relevan dengan perhitungan Euclidean distance, yaitu Min-Max Scaling, Z-Score, dan Decimal Scaling [10].

Beberapa penelitian sebelumnya telah mengkaji perbandingan teknik normalisasi terhadap kinerja berbagai algoritma klasifikasi, yang memiliki dampak signifikan pada akurasi prediksi dalam berbagai dataset. Sebagai contoh,

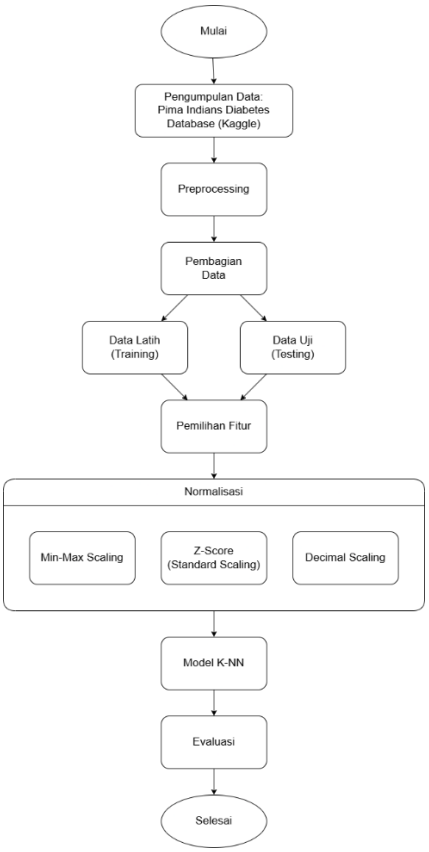
e-ISSN: 2548-6861

penelitian Muasir Pagan et al. [11] mengkaji perbandingan teknik normalisasi terhadap kinerja algoritma K-Nearest Neighbor (K-NN) dengan menggunakan sepuluh dataset. Mereka mengevaluasi tiga teknik skala data (min-max normalization, Z-score, dan decimal scaling). Hasilnya menunjukkan bahwa Z-score dan decimal scaling memberikan kinerja yang lebih baik dibandingkan min-max normalization, dengan Z-score secara konsisten menghasilkan akurasi, presisi, recall, dan F1-score yang lebih tinggi di sebagian besar dataset. Temuan ini menyoroti pentingnya pemilihan teknik normalisasi yang sesuai berdasarkan karakteristik dataset. Selanjutnya, penelitian Alshdaifat et al. [12] juga mengevaluasi dampak teknik normalisasi (Min-Max, Z-Score, dan Decimal Scaling) terhadap kinerja algoritma klasifikasi seperti SVM dan ANN, dengan temuan yang juga relevan untuk K-Nearest Neighbor (KNN). Hasil penelitian pada 18 dataset menunjukkan bahwa Z-Score Normalization sering kali memberikan hasil terbaik karena kemampuannya menangani outlier, sementara Decimal Scaling dianggap kurang efektif dalam meningkatkan akurasi model secara keseluruhan. Selain itu, penelitian Saichon Sinsomboonthong [13] membandingkan kinerja delapan teknik normalisasi dalam klasifikasi menggunakan ANN pada enam dataset. Hasil penelitian menunjukkan bahwa min-max normalization umumnya memberikan akurasi tertinggi dan MSE terendah. Namun, pada beberapa dataset seperti White Wine Quality dan Pima Indians Diabetes, Adjusted-2 min-max normalization memberikan hasil yang lebih baik. Teknik normalisasi lain seperti Statistical Column dan Decimal Scaling juga menunjukkan hasil kompetitif pada dataset tertentu, namun tidak mengungguli min-max normalization.

Penelitian ini bertujuan untuk menganalisis perbandingan berbagai teknik normalisasi terhadap kinerja model KNN dalam klasifikasi diabetes pada dataset Pima Indians Diabetes Database. Teknik normalisasi diterapkan sebagai bagian dari preprocessing data untuk mengevaluasi dampaknya terhadap kinerja model [14]. Selain itu, pemilihan fitur juga dilakukan untuk memastikan hanya fitur relevan yang digunakan, guna mengurangi overfitting dan meningkatkan interpretabilitas [15]. 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, pemeriksaan nilai hilang (missing values), pemeriksaan duplikasi, pemeriksaan skala data numerik, dan pemeriksaan nilai unik pada data kategorikal. Dataset kemudian dibagi menjadi dua bagian, yaitu data latih (training) dan data uji (testing). Selanjutnya, dilakukan pemilihan fitur (feature selection) untuk menentukan atribut yang paling relevan terhadap model. Setelah itu, dilakukan normalisasi data menggunakan tiga teknik, yaitu Min-Max Scaling, Z-Score Scaling (Standard Scaling), dan Decimal Scaling. Setelah normalisasi, dilakukan pelatihan model K-Nearest Neighbor (KNN) dan evaluasi kinerja model. Terakhir, dilakukan analisis terhadap hasil evaluasi model untuk setiap metode normalisasi guna mengetahui perbandingan masing-masing metode terhadap kinerja model. 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 Database yang diambil dari Kaggle. Dataset ini berasal dari National Institute of Diabetes and Digestive and Kidney Diseases dengan tujuan untuk memprediksi secara diagnostik apakah seorang pasien mengidap diabetes atau tidak berdasarkan berbagai fitur pengukur medis [16]. Dataset ini mencakup 768 baris data, di mana setiap baris mewakili satu pasien. Setiap baris terdiri dari 8 fitur independen yang digunakan untuk memprediksi kemungkinan diabetes serta satu fitur target yang menunjukkan hasil diagnosis diabetes. Fitur tersebut meliputi jumlah kehamilan (Pregnancies), konsentrasi glukosa plasma (Glucose), tekanan darah diastolik (BloodPressure), ketebalan lipatan kulit (SkinThickness), kadar insulin (Insulin), indeks

massa tubuh (BMI), fungsi keturunan diabetes (DiabetesPedigreeFunction), usia (Age), serta fitur target yang menunjukkan hasil diagnosis diabetes (Outcome) [17]. Semua pasien dalam dataset ini adalah perempuan berusia minimal 21 tahun dengan latar belakang keturunan Pima Indian [18]. Visualisasi dataset digambarkan pada Tabel 1.

TABEL 1
FITUR PIMA INDIANS DIABETES DATASET

Fitur	Deskripsi
Pregnancies	Jumlah kehamilan yang pernah dialami oleh pasien.
Glucose	Konsentrasi glukosa plasma saat berpuasa.
BloodPressure	Tekanan darah diastolik (tekanan darah saat relaksasi jantung).
SkinThickness	Ketebalan lipatan kulit yang diukur di lengan atas.
Insulin	Kadar insulin yang terdapat dalam darah pasien.
BMI	Indeks massa tubuh, yang mencerminkan berat badan pasien relatif terhadap tinggi badan.
Diabetes Pedigree Function	Fungsi keturunan diabetes, yang menggambarkan riwayat keluarga 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 [19]. 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 [20]. 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 [21]. 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 prediksi [22]. 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 [23]. Khusus pada fitur Outcome, distribusi nilai unik dianalisis untuk memastikan keseimbangan kelas data.

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 memprediksi hasil pada data uji yang belum pernah dilihat sebelumnya, sehingga dapat dinilai kemampuan generalisasi model [24].

D. Pemilihan Fitur

Setelah pembagian dataset menjadi data pelatihan dan pengujian, langkah selanjutnya adalah melakukan feature selection menggunakan metode Random Forest. Feature selection adalah proses penting untuk memilih fitur-fitur yang paling relevan dan signifikan dalam model, sehingga dapat meningkatkan akurasi serta efisiensi komputasi [25]. Random Forest, yang merupakan algoritma berbasis pohon keputusan, dapat digunakan untuk menentukan pentingnya setiap fitur dalam memprediksi target fitur [26]. Feature selection 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 dari setiap fitur dengan cara menghitung feature importance berdasarkan seberapa besar kontribusi masing-masing fitur dalam mengurangi ketidakpastian (impurity) dalam pohon keputusan. Fitur yang memiliki nilai importance tinggi dianggap lebih berpengaruh dalam proses prediksi dan akan dipertahankan, sementara fitur dengan importance rendah dapat dipertimbangkan untuk dihapus guna menyederhanakan model dan meningkatkan performa. Deteksi feature importance dilakukan hanya pada data training, agar tidak terjadi kebocoran data (data leakage). Namun, penghapusan fitur yang dianggap tidak relevan harus diterapkan secara konsisten pada kedua dataset, baik training maupun testing, agar struktur data tetap selaras saat proses pelatihan dan evaluasi model dilakukan [27]. Proses ini membantu mengurangi kompleksitas model dan mencegah overfitting, yang pada gilirannya dapat meningkatkan performa model dalam memprediksi data yang belum pernah dilihat sebelumnya.

E. Normalisasi

Setelah tahap pemilihan fitur, langkah berikutnya adalah melakukan normalisasi data. Normalisasi dilakukan untuk

e-ISSN: 2548-6861

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 tiga metode, yaitu Min-Max Scaling, Z-Score Normalization (Standard Scaling), dan Decimal Scaling. Masing-masing metode memiliki karakteristik dan manfaat spesifik dalam mengubah distribusi data agar lebih optimal untuk analisis KNN. Penjelasan lebih lanjut mengenai masing-masing metode akan dijelaskan sebagai berikut:

1. Min-Max Scaling

   Min-Max Scaling diterapkan 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. Dengan demikian, distribusi data tetap terjaga, tetapi dalam skala yang lebih seragam, sehingga model dapat mengolahnya tanpa bias akibat perbedaan skala antar fitur [29]. Rumus Min-Max Scaling adalah sebagai berikut:

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

   Nilai x merupakan data asli sebelum dinormalisasi, sementara $\min(X)$ dan $\max(X)$ masing-masing mewakili nilai terkecil dan terbesar dalam dataset. Setelah dilakukan proses normalisasi, nilai x' menjadi data yang telah disesuaikan dalam rentang [0, 1].

2. Z-Score (Standard Scaling)

   Z-Score atau Standard Scaling digunakan untuk menstandarisasi data dengan mereskalakan 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 yang memiliki skala berbeda [30]. Rumus Z-Score adalah sebagai berikut:

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

   Nilai x merupakan data asli sebelum distandarisasi, sedangkan μ mewakili rata-rata (mean) dari suatu fitur, dan σ adalah standar deviasi yang menggambarkan sebaran data. Setelah proses standardisasi, nilai z diperoleh untuk menunjukkan seberapa jauh x berada dari rata-rata dalam satuan deviasi standar.

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, sehingga skala data tetap proporsional tanpa mengubah distribusi relatif antar nilai [13]. Rumus Decimal Scaling adalah sebagai berikut:

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

   Nilai x merupakan data asli sebelum dinormalisasi, sementara 10^j adalah faktor pembagi yang ditentukan berdasarkan jumlah digit desimal yang diperlukan agar nilai x berada dalam rentang yang lebih kecil. Setelah proses normalisasi menggunakan Decimal Scaling, nilai x^* diperoleh sebagai hasil pembagian x dengan faktor 10^j.

F. Model KNN (K-Nearest Neighbors)

Setelah proses pemilihan fitur, tahap berikutnya adalah melatih model K-Nearest Neighbors (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, untuk menganalisis performa model 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 [6]. Oleh karena itu, pemilihan nilai k menjadi faktor krusial dalam kinerja model. Nilai k yang terlalu kecil dapat menyebabkan model terlalu sensitif terhadap data training (overfitting), sedangkan nilai k yang terlalu besar dapat menyebabkan model menjadi terlalu umum (underfitting) [31][32]. Hasil dari tahap ini digunakan untuk menganalisis pengaruh setiap metode normalisasi terhadap performa algoritma KNN pada berbagai nilai k, serta untuk mengevaluasi metode normalisasi mana yang paling optimal dalam meningkatkan kualitas klasifikasi.

G. Evaluasi

Pada tahap evaluasi, kinerja model K-Nearest Neighbors (KNN) yang telah dilatih akan diuji menggunakan berbagai metrik evaluasi untuk menilai efektivitas model dalam mengidentifikasi pasien dengan diabetes serta menganalisis perbandingan metode normalisasi terhadap performa klasifikasi. Evaluasi dilakukan dengan menggunakan seluruh variasi nilai k, yaitu 1, 3, 5, 7, 9, dan 11, untuk mengamati dampak masing-masing metode normalisasi secara menyeluruh terhadap hasil klasifikasi.

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

ketika model salah memprediksi pasien tidak mengidap diabetes padahal sebenarnya mengidap [33]. Keempat komponen ini menjadi dasar dalam menghitung berbagai metrik evaluasi yang digunakan untuk mengukur kinerja model 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 prediksi yang benar dibandingkan dengan total prediksi. Akurasi memberikan gambaran umum tentang performa model, tetapi kurang berguna pada data yang tidak seimbang. Akurasi dihitung dengan rumus:
   $$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$
2. Precision
   Precision mengukur akurasi prediksi positif yang dilakukan oleh model. Metrik ini digunakan untuk mengetahui seberapa banyak prediksi positif yang akurat atau sesuai dengan kondisi yang sebenarnya. Precision dihitung dengan rumus:
   $$Precision = \frac{TP}{TP + FP}$$
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:
   $$Recall = \frac{TP}{TP + FN}$$
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:
   $$Specificity = \frac{TN}{TN + FP}$$
6. ROC AUC
   ROC-AUC mengukur kemampuan model dalam membedakan antara kelas positif dan negatif. Nilai AUC yang lebih tinggi menunjukkan model yang lebih baik dalam klasifikasi pada berbagai threshold.
7. P-value
   P-Value digunakan untuk menguji signifikansi perbedaan performa model berdasarkan metode normalisasi yang digunakan. Dalam konteks ini, uji statistik dilakukan dengan metode Paired t-test, yaitu uji dua sisi untuk membandingkan hasil metrik dari dua model atau dua perlakuan pada dataset yang sama. Nilai p yang lebih kecil dari 0,05 dianggap menunjukkan perbedaan yang signifikan secara statistik. Dengan demikian, p-value membantu mengonfirmasi apakah perbedaan performa antara dua pendekatan normalisasi dapat dipercaya secara ilmiah atau hanya terjadi karena kebetulan.

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 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 akan dapat melakukan perhitungan jarak yang benar, sehingga dapat mempengaruhi akurasi hasil prediksi.

TABEL 2
TIPE DATA PADA SETIAP 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, dapat dilihat bahwa sebagian besar fitur memiliki tipe data int64, kecuali fitur BMI dan DiabetesPedigreeFunction yang bertipe float64. Meskipun terdapat perbedaan antara int64 dan float64, kedua tipe data tersebut tetap termasuk dalam kategori numerik dan dapat diproses oleh algoritma KNN tanpa memerlukan konversi tipe data tambahan.

Pada tahap selanjutnya dalam preprocessing, dilakukan pemeriksaan terhadap missing values dan data duplikat. Jika ditemukan data yang memiliki missing values, data tersebut akan dihapus untuk memastikan hanya data yang lengkap yang digunakan dalam analisis. Begitu juga, jika ditemukan data duplikat, data tersebut akan dihapus untuk mencegah pengaruhnya terhadap hasil analisis, yang dapat

e-ISSN: 2548-6861

menyebabkan model menjadi terlalu fokus pada pola tertentu atau memberikan 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. Ini menunjukkan bahwa dataset sudah dalam kondisi yang baik, dengan data yang lengkap dan unik. Dengan demikian, tidak perlu ada penghapusan data atau penanganan lebih lanjut terkait missing values atau duplikat.

Langkah selanjutnya dalam preprocessing adalah mengklasifikasikan fitur dalam dataset berdasarkan jenisnya, 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. Hasil klasifikasi fitur berdasarkan jenisnya pada Tabel 4.

TABEL 4
KLASIFIKASI FITUR BERDASARKAN JENISNYA

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 (0 atau 1). Setelah pengklasifikasian ini, langkah selanjutnya adalah melakukan 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. Analisis terhadap fitur numerik dan kategorikal telah dilakukan, seperti yang ditampilkan pada Tabel berikut:

1. Data Numerik
   Distribusi dan rentang nilai fitur numerik diperiksa menggunakan 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 DESKRIPTIF PADA DATA 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 dapat dilihat pada Tabel 5, terlihat bahwa terdapat variasi yang 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. Sementara itu, fitur "Age" memiliki rentang nilai antara 21 hingga 81, dengan rata-rata 33.24 dan deviasi standar 11.76. Selain itu, fitur "Pregnancies" memiliki nilai maksimum 17 dan rata-rata 3.85, sementara fitur "Insulin" menunjukkan deviasi standar yang sangat tinggi, mencapai 115.24, yang menunjukkan adanya variasi yang besar dalam data. Fitur seperti "SkinThickness" memiliki nilai minimum 0, dan rata-rata yang cukup rendah (20.54), yang bisa mempengaruhi model karena perbedaan skala yang sangat besar antar fitur. Fitur-fitur seperti "BMI" juga menunjukkan variabilitas yang cukup besar, dengan rentang nilai antara 0 hingga 67.1, rata-rata 31.99, dan deviasi standar 7.88. Perbedaan skala yang sangat besar antar fitur ini dapat mempengaruhi kinerja model, karena fitur dengan rentang nilai yang lebih besar cenderung mendominasi pembelajaran model. Oleh karena itu, normalisasi

diperlukan untuk menyelaraskan skala dan rentang nilai antar fitur, sehingga setiap fitur dapat berkontribusi secara seimbang dalam model yang dibangun.

2. Data 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 [0, 1].

TABEL 6
NILAI UNIK PADA DATA KATEGORIKAL

Fitur	Nilai Unik
Outcome	0
	1

Fitur Outcome ini merupakan fitur target dalam dataset yang mengindikasikan apakah seorang pasien mengidap diabetes (1) atau tidak mengidap diabetes (0). Karena hanya memiliki dua nilai unik, fitur ini dapat diperlakukan sebagai fitur kategorikal biner. Dengan hanya dua kelas, 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 digunakan dalam tahap pemodelan dan evaluasi selanjutnya.

TABEL 7
PEMBAGIAN DATASET LATIH DAN UJI

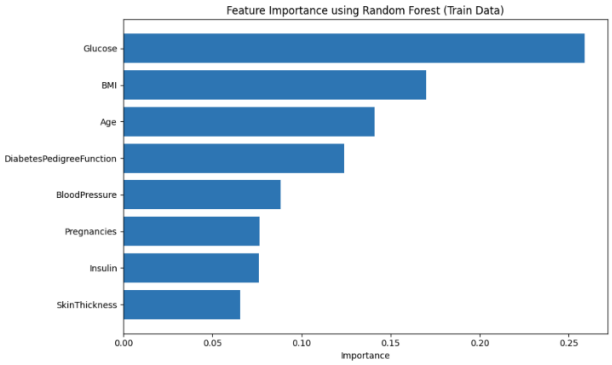
Data	Jumlah Baris	Jumlah Kolom
Data Training	614	9
Data Testing	154	9

C. Pemilihan Fitur

Setelah pembagian dataset menjadi data latih dan data uji, tahap berikutnya adalah pemilihan fitur. Pemilihan fitur bertujuan untuk mengidentifikasi atribut-atribut mana saja dalam dataset yang memiliki kontribusi paling besar dalam memprediksi target yang diinginkan, dalam hal ini adalah kemungkinan seseorang mengidap diabetes. Untuk melakukan pemilihan fitur, digunakan metode Random Forest, yang memungkinkan kita untuk mengukur tingkat pentingnya setiap fitur dalam menentukan hasil prediksi. Proses deteksi pentingnya fitur dilakukan pada fitur numerik, karena fitur kategorikal seperti Outcome sudah merupakan hasil klasifikasi dan tidak relevan untuk diukur pentingnya menggunakan metode ini. Feature importance yang diperoleh dari model Random Forest hanya dihitung berdasarkan data latih, karena model dilatih menggunakan data tersebut. Namun, penghapusan fitur yang dianggap tidak relevan harus diterapkan secara konsisten pada kedua dataset, baik data latih maupun data uji, agar model yang dievaluasi tidak terpengaruh oleh fitur yang tidak memberikan kontribusi signifikan terhadap hasil prediksi.

TABEL 8
FEATURE IMPORTANCE

Fitur	Importance
Pregnancies	0.076551
Glucose	0.258864
BloodPressure	0.088134
SkinThickness	0.065646
Insulin	0.076122
BMI	0.169984
DiabetesPedigreeFunction	0.123768
Age	0.140931



Gambar 2. Feature Importance Plot

Berdasarkan analisis feature importance yang ditunjukkan dalam Gambar 2 dan Tabel 8, beberapa fitur memiliki kontribusi yang lebih besar dibandingkan yang lain. Fitur Glucose, BMI, dan Age memiliki nilai penting yang jauh lebih tinggi dibandingkan fitur lainnya. Sebaliknya, 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 prediksi. Fitur Pregnancies (0.076551) dan BloodPressure (0.088134) juga menunjukkan nilai yang rendah, meskipun sedikit lebih tinggi, namun tidak cukup signifikan untuk dipertahankan dalam model. Fitur SkinThickness, Insulin, Pregnancies, dan BloodPressure diputuskan untuk dihapus karena kontribusinya yang sangat kecil terhadap model. Menghapus fitur-fitur ini akan meningkatkan efisiensi model, mengurangi risiko overfitting, dan memastikan fokus pada fitur yang lebih relevan, 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. Hasil dari feature selection ini dapat dilihat pada Tabel 9.

e-ISSN: 2548-6861

TABEL 9
FITUR YANG TERSISA SETELAH FEATURE SELECTION

Jenis Data	Fitur
Numerik	Glucose
	BMI
	DiabetesPedigreeFunction
	Age
Kategorikal	Outcome

D. Normalisasi

Setelah pembagian data menjadi data training dan testing, data akan dilakukan normalisasi. Data asli dapat dilihat pada Tabel 10 dan 11 yang memberikan gambaran tentang rentang dan variasi nilai pada setiap fitur dalam data training dan data testing sebelum proses normalisasi dilakukan. Dengan melihat data asli ini, dapat lebih jelas dipahami adanya perbedaan skala yang signifikan antar fitur.

TABEL 10
DATA TRAINING 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 11
DATA TESTING 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. Oleh karena itu, 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 training dan testing. Selanjutnya, akan dibahas lebih lanjut mengenai detail metode normalisasi yang diterapkan pada data ini:

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 training, 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 testing, transformasi dilakukan menggunakan nilai minimum dan maksimum yang telah dihitung dari data training tanpa melakukan perhitungan ulang. Hasil dari normalisasi ini dapat dilihat pada Tabel 12 untuk data training dan Tabel 13 untuk data testing.

TABEL 12
DATA TRAINING 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 13
DATA TESTING 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 (Standard Scaling)

Z-Score Scaling mengubah data dengan cara menstandarisasi setiap fitur sehingga memiliki rata-rata 0 dan deviasi standar 1. Proses Z-Score ini menggunakan nilai mean dan standar deviasi yang dihitung dari data training untuk menstandarkan fitur sehingga memiliki distribusi dengan mean nol dan standar deviasi satu. Parameter ini kemudian diterapkan pada data testing tanpa menghitung ulang statistik baru. Dengan cara ini, data pada data testing akan disesuaikan menggunakan parameter yang diperoleh dari data training, memastikan konsistensi dalam distribusi data. Hasil dari normalisasi ini dapat dilihat pada Tabel 14 untuk data training dan Tabel 15 untuk data testing.

TABEL 14
DATA TRAINING SETELAH DILAKUKAN Z-SCORE (STANDARD 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 15
DATA TESTING SETELAH DILAKUKAN Z-SCORE (STANDARD 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 menyesuaikan skala fitur dengan membagi nilai setiap fitur dengan pangkat 10 berdasarkan jumlah digit terbesar dalam dataset. Karena skala ini ditentukan berdasarkan distribusi keseluruhan data, jika hanya dihitung dari data training, distribusi data testing bisa berbeda sehingga skala menjadi tidak konsisten. Pembagian ini dilakukan agar nilai-nilai dalam dataset tidak terlalu besar atau kecil, tetapi tetap mempertahankan proporsi relatif antar data. Teknik ini sederhana, karena hanya melibatkan pembagian dengan angka tetap, dan memastikan distribusi data tetap terjaga dalam rentang lebih kecil.

TABEL 16
DATA TRAINING 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 17
DATA TESTING 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 pemilihan fitur, langkah berikutnya adalah melatih model KNN (K-Nearest Neighbors). Proses pelatihan ini akan dilakukan pada dua versi data: satu yang telah dinormalisasi menggunakan berbagai metode dan satu lagi tanpa normalisasi, guna membandingkan kinerjanya secara menyeluruh. Setiap model KNN akan diuji dengan berbagai nilai k, yaitu 1, 3, 5, 7, 9, dan 11. Dalam KNN, nilai k merujuk pada jumlah tetangga terdekat yang digunakan untuk menentukan kelas data yang sedang diuji. Seluruh nilai k tersebut akan digunakan dalam proses evaluasi untuk menganalisis bagaimana performa model berubah seiring dengan variasi jumlah tetangga, baik pada data yang telah dinormalisasi maupun yang tidak, sehingga dapat menentukan pendekatan terbaik untuk meningkatkan akurasi model.

F. Evaluasi

Setelah model KNN dilatih pada data tanpa normalisasi dan data yang dinormalisasi dengan berbagai metode, tahap selanjutnya adalah mengevaluasi performa masing-masing model. Evaluasi dilakukan menggunakan enam metrik utama, yaitu akurasi, precision, recall, F1-score, specificity, dan ROC AUC. Metrik-metrik ini dipilih karena mampu memberikan gambaran menyeluruh mengenai kekuatan dan kelemahan model, termasuk kemampuannya dalam mengklasifikasikan data positif dan negatif secara tepat. Pengujian dilakukan pada setiap nilai K (1, 3, 5, 7, 9, dan 11) untuk mengamati stabilitas dan sensitivitas model terhadap variasi jumlah tetangga terdekat. Hasil evaluasi performa model KNN berdasarkan kombinasi nilai K dan metode normalisasi dapat dilihat pada Tabel 18.

TABEL 18
HASIL EVALUASI PERFORMA MODEL KNN BERDASARKAN METODE NORMALISASI DAN NILAI K

Metode Normalisasi	Neighbors (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 (Standard 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 penerapan teknik normalisasi, performa model menunjukkan hasil yang kurang optimal jika dibandingkan dengan skenario lainnya. Akurasi tertinggi yang diperoleh hanya sebesar 0.7597 pada nilai K = 11, dan hasil ini tidak menunjukkan tren yang konsisten seiring bertambahnya nilai K. Sebagai contoh, akurasi sempat menurun menjadi 0.7273 pada K = 7, yang merupakan angka yang sama dengan saat K = 1. Hal ini menunjukkan bahwa tanpa penyamaan skala antar fitur, model kesulitan melakukan perhitungan jarak secara akurat, sehingga memengaruhi kemampuan klasifikasi. Precision tertinggi hanya mencapai 0.7390 dan mengalami fluktuasi pada beberapa nilai K, yang mengindikasikan bahwa model belum cukup efisien dalam mengklasifikasikan data ke kelas positif. Recall juga relatif rendah, dengan nilai tertinggi sebesar 0.7444, menunjukkan bahwa sebagian data positif tidak berhasil dikenali oleh model. F1-Score, yang mencerminkan keseimbangan antara precision dan recall, hanya mencapai angka maksimal 0.7414, dengan tren performa yang cenderung stagnan. Specificity mencapai nilai terbaik sebesar 0.7980 pada K = 9 dan 11, namun nilai ini belum menunjukkan peningkatan signifikan dibanding metode lainnya. ROC AUC yang diperoleh berada pada angka maksimal 0.7802, yang menandakan bahwa kemampuan model dalam membedakan antara kelas positif dan negatif masih terbatas. Secara keseluruhan, hasil ini menunjukkan bahwa tanpa proses normalisasi, KNN tidak dapat memproses perhitungan jarak secara seimbang, terutama ketika data memiliki fitur dengan skala yang bervariasi, sehingga berdampak negatif terhadap performa klasifikasi.

Penerapan Min-Max Scaling menghasilkan peningkatan performa yang signifikan pada seluruh metrik evaluasi KNN. Akurasi tertinggi tercapai pada nilai K = 7 dengan nilai sebesar 0.8117, yang merupakan angka tertinggi dibanding semua metode yang diuji. Precision meningkat secara konsisten hingga mencapai 0.7946, menunjukkan kemampuan model dalam mengklasifikasikan data positif secara lebih tepat dengan tingkat kesalahan false positive yang lebih rendah. Recall juga mengalami peningkatan signifikan, dengan nilai tertinggi sebesar 0.7970, yang mencerminkan sensitivitas model terhadap data positif yang lebih baik dibandingkan metode lainnya. F1-Score yang tertinggi mencapai angka 0.7958, yang mengindikasikan keseimbangan performa precision dan recall yang optimal. Specificity pun menunjukkan performa yang sangat baik dengan nilai maksimal sebesar 0.8485, menandakan bahwa model berhasil mengidentifikasi data negatif secara akurat. Nilai ROC AUC yang diperoleh mencapai 0.8050, memperlihatkan kemampuan model yang kuat dalam membedakan antara dua kelas secara keseluruhan. Efektivitas Min-Max Scaling berasal dari kemampuannya dalam menyamakan rentang semua fitur ke dalam skala 0 hingga 1, sehingga setiap fitur memiliki pengaruh yang seimbang dalam perhitungan jarak. Dengan demikian, model menjadi lebih stabil, lebih akurat, dan dapat diandalkan untuk berbagai jenis data multivariat yang memiliki skala berbeda.

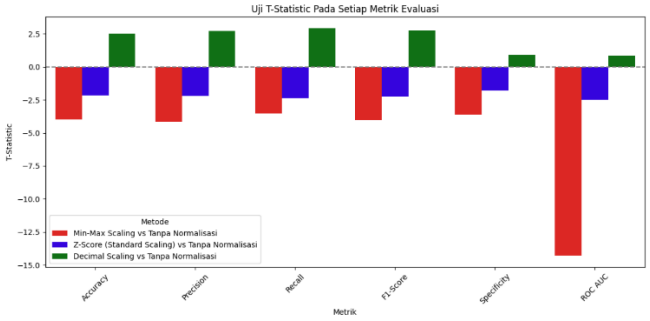
Normalisasi dengan metode Z-Score atau Standard Scaling juga menunjukkan hasil yang kompetitif dalam meningkatkan performa KNN, meskipun dalam beberapa metrik sedikit berada di bawah Min-Max Scaling. Akurasi tertinggi tercatat sebesar 0.7922 pada K = 7, dan menunjukkan pola yang stabil di berbagai nilai K. Precision maksimal berada pada angka 0.7737, dengan tren yang cukup konsisten dan mengindikasikan kemampuan model dalam menekan false positive secara efektif. Recall juga menunjukkan performa yang baik dengan nilai tertinggi sebesar 0.7586, yang menandakan bahwa model cukup sensitif terhadap data positif. F1-Score mencapai nilai maksimal sebesar 0.7554, mencerminkan keseimbangan yang solid antara precision dan recall. Specificity tertinggi mencapai 0.8384, yang menunjukkan kemampuan model dalam mengenali data negatif secara akurat dan konsisten. Hal yang menonjol dari Z-Score adalah nilai ROC AUC yang tertinggi dibanding metode lainnya, yaitu sebesar 0.8129. Nilai ini mengindikasikan bahwa Z-Score sangat efektif dalam meningkatkan kemampuan model dalam membedakan dua kelas secara global. Z-Score bekerja dengan mengurangi setiap nilai fitur dengan rata-rata dan membaginya dengan standar deviasi, sehingga fitur memiliki distribusi standar yang setara. Pendekatan ini tidak hanya menyetarakan skala, tetapi juga mempertahankan struktur distribusi data,

e-ISSN: 2548-6861

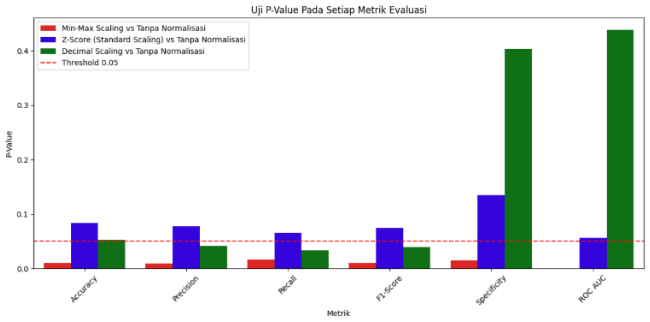
menjadikan Z-Score sangat sesuai untuk data dengan distribusi normal.

Normalisasi menggunakan Decimal Scaling menghasilkan performa yang paling rendah dibandingkan dengan metode normalisasi lainnya maupun kondisi tanpa normalisasi. Akurasi tertinggi yang diperoleh hanya sebesar 0.7403 pada nilai K = 9, dengan tren yang tidak stabil pada nilai K lainnya. Precision maksimum juga hanya mencapai 0.7172, yang menunjukkan bahwa model memiliki keterbatasan dalam mengidentifikasi data positif secara akurat. Hal ini juga tercermin pada nilai recall tertinggi yang sama-sama berada pada angka 0.7172, menandakan bahwa sensitivitas model terhadap data positif belum optimal. F1-Score yang tertinggi tidak menunjukkan perbedaan signifikan karena nilainya tetap sebesar 0.7172, mengindikasikan bahwa keseimbangan antara precision dan recall tidak berhasil dicapai secara efektif. Specificity tertinggi berada pada angka 0.7980, setara dengan hasil terbaik pada kondisi tanpa normalisasi, sehingga tidak menunjukkan adanya peningkatan performa. Nilai ROC AUC tertinggi yang dicapai adalah sebesar 0.7874, masih lebih rendah dibandingkan metode Z-Score maupun Min-Max Scaling. Decimal Scaling bekerja dengan menggeser titik desimal berdasarkan nilai maksimum fitur, tanpa mempertimbangkan distribusi atau penyebaran nilai fitur tersebut. Akibatnya, skala antar fitur tetap tidak seragam dan perhitungan jarak dalam KNN menjadi bias. Oleh karena itu, Decimal Scaling kurang direkomendasikan untuk digunakan pada model klasifikasi berbasis jarak seperti KNN, khususnya saat menangani data multivariat dengan rentang nilai yang beragam.

Uji statistik dilakukan untuk mengetahui apakah perbedaan performa model K-Nearest Neighbors (KNN) yang disebabkan oleh penerapan metode normalisasi bersifat signifikan secara statistik. Pengujian dilakukan menggunakan metode paired t-test dua sisi, dengan batas signifikansi sebesar 0,05. Uji ini membandingkan performa setiap metode normalisasi, yaitu Min-Max Scaling, Z-Score atau Standard Scaling, dan Decimal Scaling, dengan kondisi Tanpa Normalisasi pada enam metrik evaluasi utama yang meliputi Accuracy, Precision, Recall, F1-Score, Specificity, dan ROC AUC. Hasil uji statistik secara lengkap dapat dilihat pada Gambar 3 dan Gambar 4, yang menyajikan nilai t-statistic dan p-value untuk masing-masing perbandingan.



Gambar 3. Hasil Uji T-Statistic Pada Setiap Metrik Evaluasi



Gambar 4. Hasil Uji P-Value Pada Setiap Metrik Evaluasi

Pada metrik Accuracy, hasil uji menunjukkan bahwa Min-Max Scaling memberikan peningkatan performa yang signifikan secara statistik jika dibandingkan dengan Tanpa Normalisasi. Hal ini ditunjukkan oleh nilai p sebesar 0,01035 yang berada di bawah ambang 0,05. Dengan demikian, peningkatan akurasi yang diperoleh dari penerapan Min-Max Scaling dapat dianggap bukan sekadar kebetulan, melainkan benar-benar signifikan secara matematis. Sebaliknya, Z-Score menghasilkan nilai p sebesar 0,08363 dan Decimal Scaling sebesar 0,05273. Kedua nilai ini berada di atas batas signifikansi, sehingga perbedaan performa yang ditimbulkan oleh Z-Score dan Decimal Scaling belum dapat dikatakan signifikan secara statistik pada metrik ini.

Pada metrik Precision, hasil yang konsisten kembali ditemukan. Min-Max Scaling menunjukkan perbedaan yang signifikan dengan nilai p sebesar 0,00909. Artinya, model dengan Min-Max Scaling secara nyata lebih presisi dalam mengklasifikasikan kelas positif dibandingkan tanpa normalisasi. Z-Score kembali tidak menunjukkan signifikansi karena nilai p-nya sebesar 0,07737 masih berada di atas ambang yang ditentukan. Menariknya, Decimal Scaling justru menunjukkan perbedaan yang signifikan pada metrik Precision dengan nilai p sebesar 0,04112. Hal ini mengindikasikan bahwa meskipun performa Decimal Scaling secara rata-rata tidak lebih tinggi, dampaknya terhadap precision secara statistik terdeteksi sebagai berbeda dari kondisi Tanpa Normalisasi.

Hasil pada metrik Recall menunjukkan bahwa Min-Max Scaling tetap mempertahankan konsistensinya sebagai metode yang signifikan, dengan nilai p sebesar 0,01648. Artinya, model dengan Min-Max Scaling mampu meningkatkan sensitivitas terhadap kelas positif secara signifikan. Z-Score kembali menunjukkan nilai p sebesar 0,06497 yang masih belum mencapai kriteria signifikansi. Di sisi lain, Decimal Scaling memperlihatkan hasil yang signifikan dengan nilai p sebesar 0,03284, menunjukkan bahwa metode ini berpengaruh dalam meningkatkan recall meskipun peningkatan absolutnya tetap lebih rendah dibandingkan dengan Min-Max Scaling.

Evaluasi pada metrik F1-Score yang merupakan keseimbangan antara precision dan recall, memperlihatkan pola yang serupa. Min-Max Scaling kembali menunjukkan

hasil yang signifikan dengan nilai p sebesar 0,01011. Decimal Scaling juga menunjukkan perbedaan yang signifikan dengan nilai p sebesar 0,03892. Sementara itu, Z-Score tidak menunjukkan perbedaan yang signifikan karena nilai p sebesar 0,07413 masih berada di atas batas ambang yang ditentukan. Hal ini memperkuat temuan sebelumnya bahwa Decimal Scaling memberikan perubahan statistik pada performa model, meskipun bukan peningkatan yang substansial dalam hal akurasi umum.

Pada metrik Specificity yang mengukur kemampuan model dalam mengklasifikasikan kelas negatif secara benar, Min-Max Scaling kembali menunjukkan hasil yang signifikan dengan nilai p sebesar 0,01545. Hal ini menunjukkan bahwa penerapan Min-Max Scaling mampu menekan false positive secara nyata. Sebaliknya, Z-Score dan Decimal Scaling menunjukkan nilai p masing-masing sebesar 0,13454 dan 0,40318. Keduanya tidak menunjukkan perbedaan signifikan secara statistik pada metrik ini, sehingga tidak dapat disimpulkan bahwa kedua metode tersebut meningkatkan kemampuan klasifikasi terhadap kelas negatif.

Terakhir, pada metrik ROC AUC yang mengukur kemampuan model dalam membedakan kelas positif dan negatif secara keseluruhan, Min-Max Scaling menunjukkan hasil yang sangat signifikan dengan nilai p sebesar 0,00003. Nilai ini menunjukkan kekuatan bukti yang sangat kuat bahwa Min-Max Scaling secara nyata meningkatkan performa klasifikasi global. Z-Score mendekati signifikansi dengan nilai p sebesar 0,05603, tetapi tetap berada sedikit di atas ambang batas. Decimal Scaling menunjukkan nilai p yang jauh lebih tinggi, yaitu sebesar 0,43839, yang menunjukkan bahwa metode ini tidak memberikan dampak signifikan dalam meningkatkan kapasitas diskriminatif model.

Berdasarkan seluruh hasil pengujian statistik, dapat disimpulkan bahwa Min-Max Scaling merupakan metode normalisasi yang paling konsisten dan unggul secara statistik. Metode ini menghasilkan perbedaan signifikan pada semua metrik evaluasi, yang mengindikasikan peningkatan performa model KNN yang stabil dan dapat diandalkan. Meskipun Z-Score secara rata-rata menunjukkan performa yang kompetitif, tidak satu pun metrik yang mencapai signifikansi statistik, yang kemungkinan disebabkan oleh variabilitas hasil atau ukuran sampel. Adapun Decimal Scaling memang menunjukkan signifikansi pada beberapa metrik seperti Precision, Recall, dan F1-Score, namun performa absolutnya masih tertinggal dan peningkatannya tidak merata. Oleh karena itu, Min-Max Scaling dapat direkomendasikan sebagai metode normalisasi terbaik dalam pengembangan model KNN, terutama ketika tujuannya adalah memperoleh peningkatan performa yang valid secara statistik dan konsisten di berbagai metrik evaluasi.

IV. KESIMPULAN

Berdasarkan hasil yang telah diperoleh, 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. Metode ini terbukti memberikan dampak positif terhadap semua metrik evaluasi yang digunakan, termasuk akurasi, precision, recall, F1-score, specificity, dan ROC AUC. 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. Di sisi lain, Z-Score Scaling juga menunjukkan hasil yang cukup baik, dengan peningkatan yang signifikan pada ROC AUC. Namun, peningkatan pada metrik lainnya tidak cukup kuat untuk dianggap signifikan secara statistik. Hal ini menunjukkan bahwa meskipun Z-Score dapat menormalkan distribusi data, dampaknya tidak selalu konsisten pada semua metrik evaluasi. Sementara itu, Decimal Scaling memberikan hasil yang paling rendah jika dibandingkan dengan kedua metode lainnya. Meskipun pada beberapa metrik, seperti precision dan recall, terdapat peningkatan yang signifikan, namun performa keseluruhan tetap lebih rendah dibandingkan dengan Min-Max Scaling dan Z-Score. Selain itu, fluktuasi pada nilai K juga menunjukkan ketidakstabilan dalam model yang menggunakan Decimal Scaling.

Berdasarkan temuan tersebut, dapat disimpulkan bahwa Min-Max Scaling adalah metode normalisasi yang paling efektif untuk model KNN, karena secara konsisten memberikan hasil yang lebih baik dan terbukti signifikan secara statistik. Oleh karena itu, untuk meningkatkan performa model KNN dalam klasifikasi data multivariat, terutama dalam kasus ini untuk prediksi diabetes, Min-Max Scaling dapat direkomendasikan sebagai metode normalisasi yang paling tepat.

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. A. Antar *et al.*, "Diabetes mellitus: Classification, mediators, and complications; A gate to identify potential targets for the development of new effective treatments," Dec. 01, 2023, *Elsevier Masson s.r.l.* doi: 10.1016/j.biopha.2023.115734.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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/aait.05.2022.5.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] M. N. Maskuri, K. Sukerti, and R. M. Herdian Bhakti, "Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Memprediksi Penyakit Stroke Stroke Desease Predict Using KNN Algorithm," *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 4, no. 1, May 2022.

[21] 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.

[22] O. Alotaibi, E. Pardede, and S. Tomy, "Cleaning Big Data Streams: A Systematic Literature Review," Aug. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/technologies11040101.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[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.