

Optimalisasi Diagnosis Dini Diabetes Dengan *Machine Learning*: Model Prediktif Berbasis *K-Nearest Neighbor*

Rizki Yusliana Bakti¹, Lukman², Wanda Tyrana Dewi MJ³, Desi Anggreani^{4*}, Muhyiddin A M Hayat⁵

^{1,2,3,4,5}Informatika, Universitas Muhammadiyah Makassar
Jl. Sultan Alauddin No.259, Makassar, 90221, Indonesia

*Corresponding author: desianggreani@unismuh.ac.id

Abstrak

Diabetes merupakan penyakit tidak menular utama yang memberikan beban kesehatan global yang signifikan, dengan prevalensinya terus meningkat. Sebagai respons, penelitian ini mengembangkan aplikasi berbasis web untuk deteksi dini risiko diabetes menggunakan algoritma K-Nearest Neighbor (KNN), yang memanfaatkan Dataset lokal dari RSUD Kolonodale, Indonesia. Aplikasi ini memberikan alat yang efisien dan mudah diakses bagi masyarakat umum dan tenaga medis untuk memprediksi risiko diabetes berdasarkan indikator klinis utama seperti kadar gula darah, tekanan darah, Indeks Massa Tubuh (IMT), kadar insulin, dan riwayat keluarga. Model ini mencapai akurasi 93%, dengan precision 89%, recall 96%, dan F1-score 93%, yang menunjukkan efektivitasnya dalam membedakan antara kasus diabetes dan non-diabetes. Meskipun sistem ini menawarkan antarmuka pengguna yang ramah dan sumber daya edukasi tentang pencegahan diabetes, masih terdapat beberapa area yang perlu perbaikan, terutama dalam hal perluasan Dataset, penanganan kesalahan, dan performa pada beban pengguna yang tinggi. Aplikasi ini juga memiliki potensi untuk diintegrasikan dengan perangkat wearable dan chatbot berbasis AI untuk meningkatkan pemantauan secara real-time dan rekomendasi pencegahan yang dipersonalisasi. Pengembangan di masa depan dapat memperluas aplikasinya di fasilitas kesehatan baik di perkotaan maupun pedesaan.

Kata kunci: Diabetes, *K-Nearest Neighbor*, *Machine Learning*, Aplikasi RasaGula.

Abstract

Diabetes is a leading non-communicable disease that poses a significant global health burden, with its prevalence continuously increasing. In response, this study developed a web-based application for early detection of diabetes risk using the K-Nearest Neighbor (KNN) algorithm, leveraging a local Dataset from RSUD Kolonodale, Indonesia. The application provides an efficient, accessible tool for both the general public and healthcare professionals to predict diabetes risk based on key clinical indicators such as blood glucose levels, blood pressure, body mass index (BMI), insulin levels, and family history. The model achieved an accuracy of 93%, with precision of 89%, recall of 96%, and an F1-score of 93%, demonstrating its effectiveness in distinguishing between diabetic and non-diabetic cases. While the system offers a user-friendly interface and educational resources on diabetes prevention, there are areas for improvement, particularly in Dataset expansion, error handling, and performance under heavy user load. The application also holds potential for integration with wearable devices and AI-driven chatbots to enhance real-time monitoring and personalized prevention recommendations. Future development can further enhance the system's applicability in both urban and rural healthcare settings.

Keywords: Diabetes, *K-Nearest Neighbor*, *Machine Learning*, *RasaGula Application*.

I. PENDAHULUAN

Diabetes merupakan salah satu penyakit tidak menular (PTM) yang paling mematikan dan menjadi beban serius dalam sistem kesehatan global. Berdasarkan laporan *International Diabetes*

Federation (IDF), pada tahun 2021 terdapat sekitar 537 juta orang dewasa (usia 20–79 tahun) yang hidup dengan diabetes, setara dengan 1 dari 10 orang dewasa secara global. Jumlah ini diproyeksikan akan meningkat menjadi 643 juta pada tahun 2030 dan mencapai 783 juta pada tahun 2045 (IDF, 2021).

Peningkatan tercepat terjadi di kawasan Asia Tenggara, dengan proyeksi kenaikan prevalensi sebesar 68% dalam dua dekade mendatang. Indonesia, sebagai negara dengan populasi besar dan urbanisasi pesat, menjadi salah satu penyumbang utama tren ini. Data dari RSUD Kolonodale, Morowali Utara, mencatat 2.550 kasus rawat jalan akibat diabetes pada tahun 2019, turun menjadi 1.232 kasus pada 2020 akibat pandemi, namun melonjak drastis menjadi 3.402 kasus pada 2021 (RSUD Kolonodale, 2021). Hal ini menunjukkan adanya keterbatasan sistem deteksi dini di layanan primer serta perlunya strategi pencegahan yang lebih efektif, cepat, dan terjangkau.

Upaya deteksi dini diabetes saat ini masih bergantung pada metode laboratorium seperti *glucose tolerance test*, *fasting blood sugar*, dan *HbA1c*, yang meskipun akurat, tergolong mahal, memakan waktu, dan sulit dijangkau di daerah terpencil. Banyak individu tidak menyadari risiko yang mereka miliki karena gejala awal diabetes sering kali tidak kentara atau diabaikan (Diallo et al., 2024; Hossain et al., 2024). Selain itu, metode deteksi klinis konvensional belum mampu mengintegrasikan data pasien secara holistik dan prediktif. Dengan semakin meluasnya penggunaan *electronic health records* (EHR), muncul peluang besar untuk mengembangkan sistem deteksi berbasis data yang lebih efisien dan skalabel (Shen et al., 2025). Dalam bidang kesehatan digital (*e-health*), algoritma *machine learning* (ML) seperti *K-Nearest Neighbor* (KNN) telah menunjukkan kinerja kompetitif dalam klasifikasi penyakit. KNN bekerja dengan membandingkan data baru terhadap sejumlah tetangga terdekat, lalu menetapkan kelas berdasarkan mayoritas (Danny et al., 2024). Penelitian oleh Lestari et al. (2021) mencatat akurasi sebesar 96% dalam prediksi diabetes menggunakan *dataset* Pima Indian, sementara Bhatnagar et al. (2022) mencatat akurasi 95% dalam deteksi penyakit ginjal kronis berbasis aplikasi web. Keunggulan KNN terletak pada fleksibilitas, kemudahan implementasi, dan hasil yang mudah diinterpretasi oleh praktisi medis.

Penelitian ini bertujuan untuk mengembangkan aplikasi pendeteksi dini diabetes berbasis web dengan menerapkan algoritma KNN pada data klinis pasien lokal dari RSUD Kolonodale. Sistem ini dirancang menggunakan backend Flask dan frontend Django, agar mudah diakses oleh masyarakat umum tanpa harus mengandalkan fasilitas medis konvensional. Alih-alih menggunakan data terbuka seperti pada studi sebelumnya, sistem ini mengolah data lokal yang mencerminkan kondisi nyata di daerah pelayanan kesehatan. Pemisahan proses antara antarmuka pengguna dan pemodelan prediktif

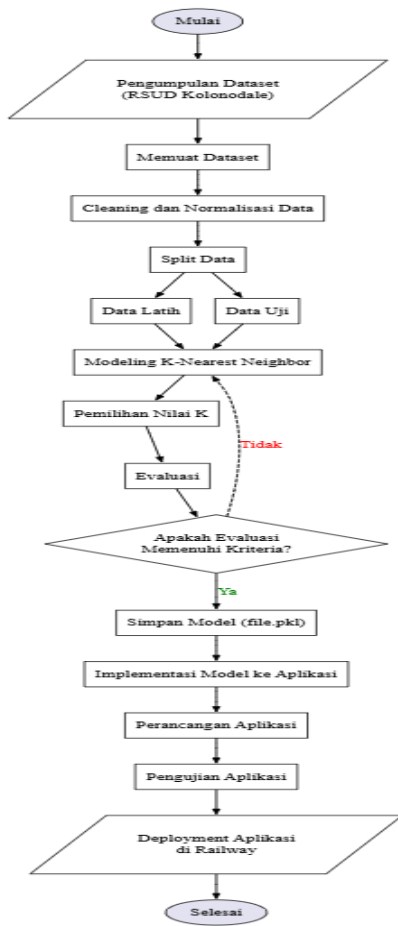
melalui penggunaan framework yang berbeda turut diterapkan untuk meningkatkan efisiensi dan fleksibilitas sistem berbasis web yang dikembangkan. Selain itu, sistem diuji dengan metode *black-box testing* untuk memastikan fungsionalitas dan keandalannya. Gap yang ingin dijawab adalah belum tersedianya sistem deteksi dini diabetes yang cepat, otomatis, murah, serta relevan secara lokal. Dengan mengembangkan sistem prediksi ini, diharapkan dapat mendukung pengambilan keputusan medis dan meningkatkan kesadaran masyarakat akan pentingnya deteksi dini, sejalan dengan tujuan *Sustainable Development Goals* (SDG) poin ke-3, yaitu menjamin kehidupan sehat dan mendukung kesejahteraan bagi semua usia.

II. METODE PENELITIAN

A. Tahapan Penelitian

Penelitian ini mencakup tahapan terintegrasi, dimulai dari pengumpulan data pasien di RSUD Kolonodale, dilanjutkan dengan pra-pemrosesan untuk menjamin kualitas dan konsistensi data. *Dataset* yang telah dibersihkan dan dinormalisasi kemudian dibagi menjadi data latih dan data uji.

Model klasifikasi dibangun menggunakan algoritma *K-Nearest Neighbor* (KNN) dan dievaluasi berdasarkan akurasi serta metrik lainnya. Model terlatih diimplementasikan ke dalam aplikasi web berbasis Django dan Flask. Selanjutnya, sistem diuji menggunakan metode *black-box testing* untuk memastikan fungsionalitas dan responsivitas berjalan optimal. Alur proses penelitian ditampilkan dalam *flowchart* pada Gambar 1.



Gambar 1. Tahapan Penelitian

B. Dataset

Dataset yang digunakan dalam penelitian ini berasal dari RSUD Kolonodale, Morowali Utara, dengan total sebanyak 1.491 data pasien yang telah melalui tahap pembersihan dan normalisasi. Setiap data memiliki delapan atribut fitur medis, yaitu jumlah kehamilan, kadar gula darah, tekanan darah, lingkaran perut, kadar insulin, Indeks Massa Tubuh (IMT), riwayat keluarga dengan diabetes, dan usia. Seluruh atribut tersebut digunakan sebagai masukan dalam proses klasifikasi untuk memprediksi risiko diabetes, sedangkan atribut Outcome digunakan sebagai label keluaran dengan nilai 1 untuk pasien yang terindikasi diabetes dan 0 untuk pasien yang tidak terindikasi.

Setelah diproses, data dibagi menjadi dua bagian, yaitu 1.192 data latih (80%) dan 299 data uji (20%), untuk menjamin objektivitas dalam pelatihan dan pengujian model. Untuk memberikan gambaran struktur data yang digunakan dalam proses pelatihan, berikut ditampilkan cuplikan 10 baris pertama dari data latih yang telah dinormalisasi. Tabel ini memperlihatkan nilai-nilai awal pada setiap atribut yang merepresentasikan data medis pasien.

Tabel 1. Data Latih dari Dataset Penelitian

	Jumlah Kehamilan	Gula Darah	Tekanan Darah	Lingkar Perut	Insulin	IMT	Riwayat Keluarga	Usia	Outcome
0	0	144.00	136.670	89.09	238.81	39.13	1	76	1
1	2	168.83	138.635	71.68	104.13	36.13	0	78	0
2	0	130.80	148.190	86.38	121.35	36.37	1	67	0
3	1	148.94	126.305	90.09	249.56	32.74	0	61	0
4	0	156.56	137.615	100.66	201.93	36.02	1	62	1
5	3	165.55	153.680	82.05	248.62	32.50	0	45	0
6	0	116.01	144.590	68.55	48.53	35.39	0	34	0
7	0	136.84	176.750	110.69	177.87	32.71	0	63	1
8	0	188.08	156.005	104.57	175.80	33.87	1	62	1
9	0	203.86	121.385	88.22	188.65	37.80	0	51	1

C. Pra-Pemrosesan Data

Tahap pra-pemrosesan data merupakan langkah krusial dalam setiap penelitian berbasis machine learning. Dalam penelitian ini, pra-pemrosesan melibatkan dua proses utama, yaitu pembersihan data dan normalisasi.

1. Pembersihan Data: Langkah pertama adalah mengatasi masalah data kosong atau duplikat dalam dataset. Penghilangan atau pengisian nilai kosong dilakukan untuk memastikan model machine learning tidak terpengaruh oleh data yang tidak valid. Data kosong pada atribut tertentu diisi dengan menggunakan rata-rata atau nilai median, tergantung pada jenis data.
2. Normalisasi Min-Max: Selanjutnya, normalisasi dilakukan untuk menyeragamkan skala antar fitur numerik agar tidak ada atribut yang mendominasi proses klasifikasi. Metode yang digunakan adalah Min-Max Scaling, dengan rumus sebagai berikut:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Yang mana X adalah nilai asli, X_{min} dan X_{max} masing-masing adalah nilai minimum dan maksimum dalam fitur tersebut, sedangkan X' adalah nilai hasil normalisasi.

Setelah proses pra-pemrosesan selesai, dataset dibagi menjadi dua bagian: 80% untuk pelatihan model dan 20% untuk pengujian model. Pembagian ini bertujuan untuk memastikan model dapat diuji dengan data yang belum pernah dilihat sebelumnya, sehingga dapat dievaluasi kinerjanya secara objektif.

D. Model Machine Learning

Pada penelitian ini, algoritma yang digunakan untuk memprediksi risiko diabetes adalah K-Nearest Neighbor (KNN), sebuah algoritma yang populer dalam machine learning untuk tugas klasifikasi. KNN bekerja dengan prinsip dasar bahwa data baru akan memiliki label yang sama dengan mayoritas tetangga terdekatnya dalam ruang fitur. Euclidean Distance digunakan sebagai metrik untuk mengukur jarak antara data. Jarak ini dihitung dengan rumus:

$$d = \sqrt{\sum_{i=1}^n (x_{i1} - x_{i2})^2}$$

Yang mana d adalah jarak *Euclidean*, x_1, y_1, \dots, n_1 adalah nilai atribut pada titik data pertama, dan x_2, y_2, \dots, n_2 adalah nilai atribut pada titik data kedua. Setelah menentukan metrik jarak, langkah berikutnya adalah memilih nilai K , yaitu jumlah tetangga yang digunakan untuk klasifikasi. Nilai K yang optimal ditemukan melalui eksperimen, di mana nilai K yang memberikan akurasi terbaik pada *dataset* pengujian dipilih sebagai parameter untuk model.

E. Evaluasi Model

Evaluasi model dilakukan untuk menilai seberapa baik model yang dibangun dalam memprediksi risiko diabetes. Dua metrik utama yang digunakan dalam evaluasi adalah *confusion matrix* dan *classification report*.

1. *Confusion Matrix*: Matriks ini memberikan gambaran visual mengenai jumlah prediksi yang benar dan salah. Matriks ini membedakan antara:
 - a. *True Positive (TP)*: Model memprediksi diabetes dan pasien memang menderita diabetes.
 - b. *True Negative (TN)*: Model memprediksi bukan diabetes dan pasien memang tidak menderita diabetes.
 - c. *False Positive (FP)*: Model memprediksi diabetes, tetapi pasien tidak menderita diabetes.
 - d. *False Negative (FN)*: Model memprediksi bukan diabetes, tetapi pasien sebenarnya menderita diabetes.
2. *Classification Report*: Laporan ini memberikan informasi lebih lanjut mengenai *precision*, *recall*, dan *F1-score*, yang masing-masing mengukur seberapa tepat, sensitif, dan seimbang model dalam klasifikasi positif dan negatif. Metrik ini sangat penting untuk memahami kinerja model secara menyeluruh.

F. Pengembangan Aplikasi Web

Pengembangan aplikasi web dilakukan untuk menyediakan antarmuka yang ramah pengguna bagi sistem deteksi dini diabetes. Meskipun Django merupakan *framework fullstack*, dalam sistem ini Django hanya difungsikan sebagai *frontend* yang bertugas membangun antarmuka web yang dinamis, responsif, dan menerima *input* dari pengguna. Sementara itu, seluruh proses pemodelan dan prediksi dilakukan oleh *backend* berbasis Flask. Flask digunakan untuk mengelola API yang menghubungkan antarmuka pengguna dengan model *machine learning KNN*. Model yang telah dilatih disimpan dalam format file *.pkl* dan diintegrasikan ke dalam *backend* Flask, sehingga ketika pengguna menginput data medis melalui *frontend*, data tersebut dikirimkan ke *backend* untuk diproses dan

hasil prediksi dikembalikan ke pengguna secara otomatis dalam hitungan detik.

G. Pengujian Sistem

Pengujian aplikasi web dilakukan menggunakan metode *black-box testing*. Metode ini bertujuan untuk mengevaluasi fungsi aplikasi berdasarkan *input* dan *output*, tanpa memperhatikan implementasi internalnya. Beberapa jenis pengujian yang dilakukan antara lain:

1. Validasi Input: Pengujian untuk memastikan aplikasi dapat memproses data yang dimasukkan dengan benar, seperti memastikan nilai yang dimasukkan sesuai dengan format yang diharapkan (misalnya, angka untuk usia dan kadar gula darah).
2. Hasil Prediksi: Pengujian dilakukan untuk memastikan bahwa hasil deteksi risiko diabetes sesuai dengan data input dan memberikan rekomendasi yang tepat.
3. Responsivitas Antarmuka: Pengujian dilakukan untuk memastikan bahwa antarmuka aplikasi tetap responsif dan optimal saat diakses melalui berbagai perangkat, seperti *desktop*, *tablet*, dan *smartphone*.

Selain pengujian fungsional, dilakukan pula pengujian performa dasar untuk menilai kecepatan dan kestabilan sistem saat menerima permintaan secara simultan. Pengujian ini menggunakan *tool* Locust, dengan lima pengguna virtual yang mengakses berbagai fitur utama secara paralel. Hasil pengujian menunjukkan bahwa waktu respons rata-rata sistem berada dalam kategori baik, tanpa terjadi *failure*. Hal ini membuktikan bahwa aplikasi mampu beroperasi secara stabil dalam skenario akses bersamaan

Tabel 2. Hasil Pengujian Performa Aplikasi Menggunakan Locust

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	33	0	180	540	820	228.54	117	815	36134	0.9	0
GET	/accounts/login	21	0	210	460	590	246.58	161	591	4545	0.5	0
GET	/dashboard/deteksi/	21	0	230	1000	5000	542.97	152	5041	4545	0.4	0
GET	/dashboard/polemakan/	25	0	210	490	520	248.13	151	516	4545	0.4	0
GET	/dashboard/hasil/	29	0	220	530	640	258.27	143	638	4545	0.3	0
Aggregated		129	0	210	540	1000	283.14	117	5041	12641.26	2.5	0

III. HASIL DAN PEMBAHASAN

A. Deskripsi Aplikasi

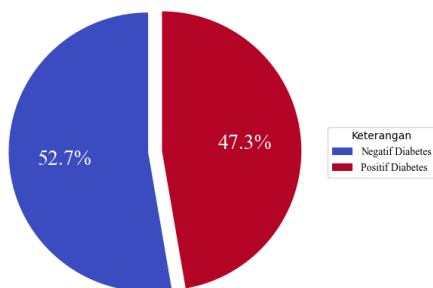
Aplikasi RasaGula merupakan sistem prediksi dini risiko diabetes berbasis web yang dirancang untuk mendekatkan teknologi kepada masyarakat luas. Nama RasaGula sendiri memiliki makna filosofis: kata “Rasa” merepresentasikan kesadaran pengguna terhadap kondisi kesehatannya, sedangkan “Gula” menunjuk pada kadar gula darah sebagai parameter utama penyakit diabetes. Dengan nama ini, aplikasi diharapkan menjadi alat bantu yang sederhana dan informatif dalam memahami risiko kesehatan secara mandiri.

Sistem aplikasi dibangun menggunakan arsitektur dua komponen: *frontend* berbasis Django dan *backend* berbasis Flask. Pengguna hanya perlu mengisi formulir data medis seperti jumlah kehamilan, gula darah, tekanan darah, lingkaran perut, insulin, IMT, riwayat keluarga, dan usia. Data tersebut kemudian dikirim melalui API ke *backend* yang mengolahnya menggunakan algoritma *K-Nearest Neighbor* (KNN) dan langsung menampilkan hasil prediksi dalam hitungan detik.

B. Distribusi dan Korelasi Data

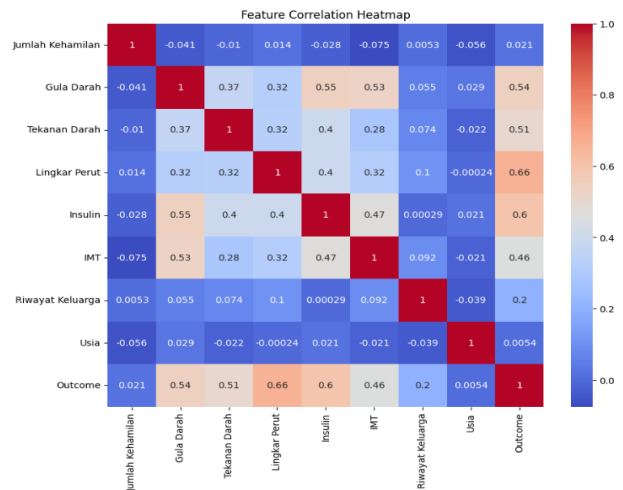
Analisis distribusi dan korelasi merupakan langkah awal penting dalam membangun model prediksi yang akurat. Berdasarkan data pasien dari RSUD Kolonodale sebanyak 1.491 sampel, distribusi kelas antara pasien yang terindikasi diabetes (47,3%) dan tidak terindikasi (52,7%) tergolong seimbang. Keseimbangan ini mendukung proses pelatihan model agar tidak bias terhadap salah satu kelas. Proporsi ini divisualisasikan dalam Gambar 2.

Proporsi Data Positif Diabetes dan Negatif Diabetes



Gambar 2. Proporsi Data Positif dan Negatif

Selanjutnya, hasil analisis korelasi menunjukkan bahwa lingkaran perut memiliki korelasi tertinggi terhadap status diabetes (0,66), diikuti oleh kadar gula darah (0,54) dan tekanan darah (0,51). Fitur-fitur ini kemudian menjadi perhatian utama dalam proses pelatihan model KNN, karena kontribusinya yang lebih signifikan dalam menentukan klasifikasi akhir. Hubungan antar fitur tersebut digambarkan dalam Gambar 3.



Gambar 3. Hasil Korelasi Antar Fitur

C. Proses Implementasi Algoritma KNN

Data sebanyak 1.491 sampel dari RSUD Kolonodale digunakan dan terdiri dari delapan atribut medis yang telah dinormalisasi menggunakan metode *Min-Max Scaling*. Pembagian data dilakukan secara acak dengan rasio 80% sebagai data latih (1.192 data) dan 20% sebagai data uji (299 data), agar model dapat diuji secara objektif.

Selanjutnya, dilakukan eksperimen terhadap berbagai nilai K menggunakan teknik validasi silang sebanyak 5 *fold*. Hasil menunjukkan bahwa nilai K=3 memberikan akurasi tertinggi dan dipilih sebagai parameter terbaik untuk model ini. Visualisasi hasil pengujian nilai K dapat dilihat pada Gambar 4.

- K = 3, Akurasi rata-rata: 92.367%
 - K = 5, Akurasi rata-rata: 91.946%
 - K = 7, Akurasi rata-rata: 90.774%
 - K = 9, Akurasi rata-rata: 90.101%
- Nilai K terbaik berdasarkan validasi silang: 3

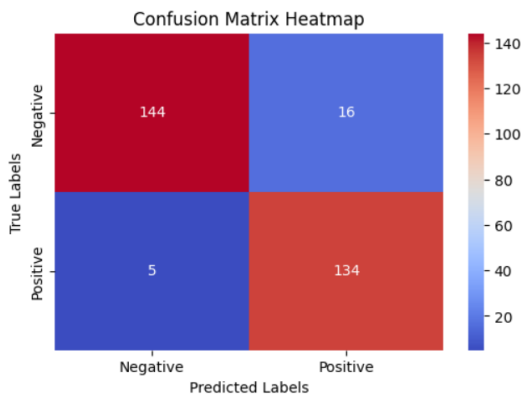
Gambar 4. Hasil Nilai K Terbaik

D. Evaluasi Model

Pada penelitian ini, algoritma K-Nearest Neighbor (KNN) dievaluasi menggunakan beberapa metrik, yakni *confusion matrix*, *classification report*, dan *error rate*.

1. Confusion Matrix

Confusion matrix memperlihatkan bagaimana model mengklasifikasikan data pasien dalam dua kategori, yakni positif diabetes (*Outcome* = 1) dan negatif diabetes (*Outcome* = 0). Hasil *confusion matrix* dapat dilihat dalam bentuk heatmap pada Gambar 5.



Gambar 5. Hasil Confusion Matrix Heatmap

Dari hasil *confusion matrix*, didapatkan data sebagai berikut:

- True Negative (TN)*: 144, yang berarti model berhasil mengklasifikasikan dengan benar 144 pasien yang tidak menderita diabetes.
- True Positive (TP)*: 134, yang berarti model berhasil mengklasifikasikan dengan benar 134 pasien yang terindikasi diabetes.
- False Negative (FN)*: 5, yang berarti model salah mengklasifikasikan 5 pasien yang sebenarnya menderita diabetes sebagai negatif.
- False Positive (FP)*: 16, yang berarti model salah mengklasifikasikan 16 pasien yang sebenarnya tidak menderita diabetes sebagai positif.

Matriks ini menunjukkan bahwa model KNN memiliki performa yang sangat baik dalam membedakan antara pasien yang terindikasi diabetes dan yang tidak. Meski ada beberapa kesalahan klasifikasi, terutama dalam kasus *False Positive* dan *False Negative*, jumlah kesalahan ini sangat kecil, menunjukkan bahwa model mampu memberikan prediksi yang akurat dan andal.

2. Classification Report

Setelah memperoleh *confusion matrix*, langkah berikutnya adalah mengukur performa model dengan menggunakan *classification report*. Metrik yang digunakan dalam *classification report* mencakup *precision*, *recall*, *f1-score*, dan *accuracy*. Berikut adalah hasil *classification report* untuk model KNN:

Tabel 3. Hasil Classification Report

Metrik	Kelas	Kelas	Makro	Rata-rata
	Negatif	Positif	Rata-rata	Berbobot
Precision	97%	89%	93%	93%
Recall	90%	96%	93%	93%
F1-Score	93%	93%	93%	93%
Accuracy	-	-	-	93%

Tabel 3 menunjukkan hasil dari *classification report*, yang menyajikan metrik evaluasi untuk kelas positif dan negatif, serta rata-rata makro dan rata-rata berbobot. Hasilnya menunjukkan:

- Precision* untuk kelas negatif adalah 97%, yang berarti bahwa 97% dari prediksi “negatif diabetes” adalah benar.

- Precision* untuk kelas positif adalah 89%, yang berarti bahwa 89% dari prediksi “positif diabetes” adalah benar.
- Recall* untuk kelas negatif adalah 90%, yang menunjukkan bahwa model dapat menemukan 90% dari kasus negatif diabetes yang ada.
- Recall* untuk kelas positif adalah 96%, yang menunjukkan bahwa model berhasil menemukan sebagian besar kasus positif diabetes.
- F1-score* menunjukkan keseimbangan antara *precision* dan *recall*. Untuk kelas negatif, nilai *F1-score* adalah 93%, dan untuk kelas positif, *F1-score* mencapai 93% juga. Nilai *F1-score* yang tinggi pada kedua kelas ini menunjukkan bahwa model memiliki performa yang baik dalam menemukan keseimbangan antara kesalahan positif dan negatif.
- Akurasi keseluruhan model tercatat sebesar 93%, yang menunjukkan bahwa model KNN memiliki kemampuan yang sangat baik dalam memprediksi risiko diabetes pada *dataset* yang diberikan.

3. Error Rate

Error rate model tercatat sebesar 7% dari total 299 data uji, atau setara dengan 21 kesalahan prediksi yang terdiri atas *False Positive* dan *False Negative*. Meskipun terdapat sejumlah kecil kesalahan klasifikasi, persentase error ini masih tergolong rendah dan menunjukkan bahwa model cukup andal dalam memprediksi risiko diabetes secara dini.

E. Perhitungan Manual dan Perbandingan Akurasi

Untuk memastikan bahwa sistem pendeteksi risiko diabetes yang dibangun telah menerapkan algoritma *K-Nearest Neighbor* (KNN) secara tepat, dilakukan perhitungan manual terhadap data uji yang telah dinormalisasi. Proses ini dibagi menjadi dua tahap: pertama, perhitungan jarak *Euclidean* pada satu data uji sebagai contoh; kedua, perhitungan menyeluruh terhadap seluruh data uji menggunakan Microsoft Excel untuk mengevaluasi akurasi sistem secara manual.

1. Perhitungan Manual pada Satu Data Uji

Satu baris data uji dipilih dan dibandingkan dengan lima data latih. Data yang digunakan belum dinormalisasi dan ditampilkan dalam Tabel 4 dan Tabel 5.

Tabel 4. Data Uji

Jumlah Kehamilan	Gula Darah	Tekanan Darah	Lingkar Perut	Insulin	IMT	Riwayat Keluarga	Usia
0	144.00	136.67	89.09	238.81	39.13	1	76

Tabel 5. Lima Data Latih

ID	Jumlah Kehamilan	Gula Darah	Tekanan Darah	Lingkar Perut	Insulin	IMT	Riwayat Keluarga	Usia	Diabetes (Label)
1	2	168.83	138.64	71.68	104.13	36.13	0	78	Tidak (0)
2	0	130.80	148.19	86.38	121.35	36.37	1	67	Tidak (0)
3	1	148.94	126.30	90.09	249.56	32.74	0	61	Tidak (0)
4	0	156.56	137.61	100.66	201.93	36.02	1	62	Ya (1)
5	3	120.67	140.12	80.32	150.65	29.76	0	70	Tidak (0)

Untuk memperjelas proses, Tabel 6 memperlihatkan hasil perhitungan jarak *Euclidean* antara data uji dan lima data latih:

Tabel 6. Hasil Perhitungan Jarak *Euclidean*

ID	Perhitungan <i>Euclidean</i>	Hasil Akhir (Jarak)	Label Diabetes
1	$\sqrt{((144-168.83)^2 + (136.67-138.64)^2 + (89.09-71.68)^2 + (238.81-104.13)^2 + (39.13-36.13)^2 + (1-0)^2 + (76-78)^2)}$	135.22	Tidak (0)
2	$\sqrt{((144-130.80)^2 + (136.67-148.19)^2 + (89.09-86.38)^2 + (238.81-121.35)^2 + (39.13-36.37)^2 + (1-1)^2 + (76-67)^2)}$	119.15	Tidak (0)
3	$\sqrt{((144-148.94)^2 + (136.67-126.30)^2 + (89.09-90.09)^2 + (238.81-249.56)^2 + (39.13-32.74)^2 + (1-0)^2 + (76-61)^2)}$	22.72	Tidak (0)
4	$\sqrt{((144-156.56)^2 + (136.67-137.61)^2 + (89.09-100.66)^2 + (238.81-201.93)^2 + (39.13-36.02)^2 + (1-1)^2 + (76-62)^2)}$	43.11	Ya (1)
5	$\sqrt{((144-120.67)^2 + (136.67-140.12)^2 + (89.09-80.32)^2 + (238.81-150.65)^2 + (39.13-29.76)^2 + (1-0)^2 + (76-70)^2)}$	92.41	Tidak (0)

Dari hasil perhitungan tersebut, tiga data dengan jarak terkecil adalah ID 3, 4, dan 5. Karena dua dari tiga tetangga memiliki label “Tidak” (0), maka data uji diprediksi sebagai Tidak Diabetes (0).

2. Perhitungan Manual terhadap Seluruh Data Uji

Sebanyak 299 data uji dibandingkan dengan 1.192 data latih yang telah dinormalisasi menggunakan metode Min-Max. Jarak *Euclidean* dihitung di Microsoft Excel menggunakan rumus =SQRT(SUMXMY2(range_data_latih,range_data_uji)). Untuk setiap data uji, dipilih tiga jarak terkecil, lalu kelas ditentukan berdasarkan label terbanyak dari ketiga tetangga. Cuplikan hasil prediksi disajikan pada Tabel 7.

Tabel 7. Hasil Prediksi Manual

ID	Jumlah Kehamilan	Gula Darah	Tekanan Darah	Lingkar Perut	Insulin	IMT	Riwayat Keluarga	Usia	Actual	Predicted	Keterangan
1	0,222222	0,786713	0,470806	0,670272	0,700606	0,797817	1	0,241379	1	1	Benar
2	0,777778	0,505108	0,366129	0,432425	0,336762	0,514553	1	0,551724	1	1	Benar
3	0	0,639136	0,693871	0,675355	0,47486	0,676195	1	0,137931	1	1	Benar
...
299	0,333333	0,600956	0,567097	0,52196	0,48763	0,534563	1	0,224138	1	1	Benar

Akurasi: 92,98%

Dari hasil pengujian, 278 data berhasil diprediksi dengan benar, dan 21 data salah klasifikasi. Akurasi manual sebesar 92,98%, sangat mendekati hasil sistem otomatis 93%, membuktikan bahwa algoritma KNN telah diterapkan dengan tepat dan dapat diverifikasi secara manual.

F. Implementasi Sistem

1. Arsitektur Sistem

Sistem deteksi dini diabetes ini diimplementasikan dalam bentuk aplikasi web yang mengintegrasikan antarmuka pengguna, pengolahan data menggunakan algoritma KNN, serta penyajian hasil prediksi. Arsitektur sistem terdiri atas dua komponen utama:

a. *Frontend*

Frontend aplikasi dibangun menggunakan framework Django, yang bertanggung jawab menampilkan antarmuka web interaktif kepada pengguna. Komponen ini memfasilitasi input data medis melalui formulir yang responsif dan terstruktur. Setelah pengguna mengisi data yang dibutuhkan, seperti usia, IMT, dan kadar gula darah, *frontend* akan mengirimkan data tersebut dalam format JSON melalui metode HTTP POST ke backend untuk diproses lebih lanjut.

Setelah menerima hasil prediksi dari *backend*, *frontend* menampilkan keluaran sistem secara dinamis dalam bentuk informasi status risiko diabetes. Seluruh tampilan dirancang untuk memberikan pengalaman penggunaan yang intuitif dan dapat diakses melalui berbagai perangkat. Visualisasi antarmuka pengguna, termasuk halaman utama, formulir input, dan hasil deteksi, disajikan pada Gambar 6 hingga Gambar 8.



Gambar 6. Halaman Beranda



Gambar 7. Halaman Deteksi

Gambar 8. Halaman Hasil Deteksi Negatif

Tampilan aplikasi ini dirancang dengan mempertimbangkan kenyamanan dan kemudahan penggunaan bagi masyarakat umum dan tenaga medis. Halaman utama aplikasi (Gambar 6) memberikan ajakan yang jelas untuk mendeteksi risiko diabetes dalam beberapa menit. Halaman deteksi (Gambar 7) memungkinkan pengguna untuk mengisi data kesehatan mereka, dan hasil deteksi ditampilkan dalam bentuk *pop-up* yang informatif (Gambar 8).

b. Backend

Backend aplikasi ini dikembangkan menggunakan *framework* Flask, yang berperan sebagai pemroses utama dalam sistem prediksi. Pada saat server dijalankan, Flask secara otomatis memuat model *K-Nearest Neighbor* (KNN) yang telah dilatih sebelumnya dan disimpan dalam file `diabetes-knn.pkl`, beserta objek normalisasi berbasis *Min-Max Scaling*. Ketika pengguna mengisi data melalui antarmuka yang disediakan *frontend* Django, data tersebut dikirim dalam format *JSON* melalui metode *HTTP POST* ke endpoint `/predict` yang disediakan oleh *backend*. Setelah menerima data, sistem melakukan validasi awal untuk memastikan seluruh input sesuai format dan berada dalam rentang nilai yang logis.

Data tervalidasi kemudian dinormalisasi dan diproses oleh model KNN. Algoritma menghitung jarak *Euclidean* antara data input dan data latih untuk menemukan tiga tetangga terdekat ($K=3$), lalu menentukan prediksi berdasarkan mayoritas kelas. Hasil klasifikasi berupa label 0 (tidak terindikasi diabetes) atau 1 (terindikasi diabetes) dikemas dalam format *JSON* dan dikirim ke *frontend* untuk ditampilkan. Selain itu, *backend* menerapkan mekanisme penanganan kesalahan, seperti respons error status 400 jika ditemukan input tidak valid. Dengan desain ini, *backend* menjembatani komunikasi antara antarmuka pengguna dan model prediktif secara cepat, ringan, dan akurat.

G. Analisis Pengujian Sistem

Dari hasil pengujian, sebagian besar fitur utama aplikasi *RasaGula* berfungsi sesuai harapan. Dari 70 skenario pengujian yang dilakukan menggunakan *test data* sistem, sebanyak 62 skenario dinyatakan berhasil. Skenario yang berhasil mencakup pengujian pada halaman utama, login, registrasi, validasi *output*, serta sebagian besar aspek performa dan responsivitas sistem. Navigasi dan elemen interaktif juga bekerja sebagaimana mestinya, mendukung pengalaman pengguna yang optimal. Namun demikian, terdapat 8 skenario yang gagal, khususnya pada validasi *input* angka negatif, penurunan kecepatan saat terjadi akses secara bersamaan, serta elemen tampilan yang bergeser ketika layar diputar.

Selain itu, data uji yang digunakan dalam proses pengujian sistem diambil dari data pasien yang belum pernah dilibatkan dalam pelatihan model, sehingga mampu merepresentasikan kondisi aktual saat sistem digunakan. Berdasarkan hasil analisis, sistem berhasil mengidentifikasi status kesehatan pasien dengan prediksi apakah seseorang *terindikasi diabetes* atau *tidak terindikasi diabetes*. Hasil prediksi yang dihasilkan oleh sistem terbukti sesuai dengan hasil perhitungan manual menggunakan algoritma *K-Nearest Neighbor* (KNN). Pengujian akhir yaitu alat uji beban Locust untuk melihat performa aplikasi web dengan mensimulasikan pengguna (*virtual users*) dan mencatat metrik-metrik penting dari permintaan *HTTP* yang dilakukan. Hasil pengujian terlihat pada Tabel 2. Tidak ada kegagalan semua request berhasil, menandakan aplikasi stabil secara fungsional. Rata-rata respon < 300 ms dimana sebagian besar endpoint, performa berada dalam batas wajar.

IV. KESIMPULAN

Penelitian ini berhasil membangun aplikasi deteksi dini diabetes berbasis web dengan menerapkan algoritma *K-Nearest Neighbor* (KNN) menggunakan data klinis dari RSUD Kolonodale. Model KNN dengan parameter $K=3$ menghasilkan akurasi sebesar 93%, *precision* 89%, *recall* 96%, *F1-score* 93%, serta *error rate* 7%. Hasil ini menunjukkan bahwa KNN dapat diandalkan dalam mengklasifikasikan risiko diabetes berdasarkan atribut medis pasien.

Aplikasi juga telah diuji melalui *black-box testing* dengan 70 skenario, di mana 62 berhasil dan 8 mengalami kegagalan pada validasi input negatif, performa saat akses bersamaan, dan tampilan rotasi layar. Meski demikian, sistem menunjukkan performa stabil, responsif, dan ramah pengguna. Solusi ini dinilai relevan dan terjangkau untuk

mendukung deteksi mandiri risiko diabetes, terutama di daerah dengan akses medis terbatas.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada RSUD Kolonodale, Morowali Utara, atas dukungan dan izin dalam pemberian data pasien yang digunakan dalam penelitian ini. Ucapan terima kasih juga disampaikan kepada Program Studi Informatika, Universitas Muhammadiyah Makassar, atas dukungan dan bimbingan selama proses penelitian. Penelitian ini tidak menerima pendanaan dari skema penelitian manapun dan tidak memiliki nomor kontrak pendanaan.

REFERENSI

- [1] International Diabetes Federation. (2021). *IDF Diabetes Atlas (9th ed.)*. International Diabetes Federation.
- [2] RSUD Kolonodale. (2021). *Laporan Kasus Diabetes di Instalasi Rawat Jalan RSUD Kolonodale Tahun 2019-2021*.
- [3] Diallo, A. I., Dieng, C. M., Tine, J. A. D., Bassoum, O., Diongue, F. B., Ba, M. F., & et al. (2024). Factors associated with diabetes knowledge, attitudes and practices among people aged 18 and over in the commune of Niakhene in Senegal. *PLOS Global Public Health*, 4(3), e0002265. <https://doi.org/10.1371/journal.pgph.0002265>
- [4] Hossain, M. J., Al-Mamun, M., & Islam, M. R. (2024). Diabetes mellitus, the fastest growing global public health concern: Early detection should be focused. *Health Science Reports*, 7(3), e2004. <https://doi.org/10.1002/hsr2.2004>
- [5] Shen, Y., Yu, J., Zhou, J., & Hu, G. (2025). Twenty-Five Years of Evolution and Hurdles in Electronic Health Records and Interoperability in Medical Research: Comprehensive Review. *Journal of Medical Internet Research*, 27, e59024. <https://doi.org/10.2196/59024>
- [6] Danny, M., Muhidin, A., & Jamal, A. (2024). Application of the K-Nearest Neighbor Machine Learning Algorithm to Predict Sales of Best-Selling Products. *Brilliance: Research of Artificial Intelligence*, 4(1), 255–264.
- [7] Bhatnagar, A., Sharma, R., & Verma, P. (2022). K-Nearest Neighbor Algorithm for Early Detection of Chronic Kidney Disease Using Web-Based Application. *International Journal of Computer Applications*, 184(3), 15–22. <https://doi.org/10.5120/ijca2022923456>
- [8] Lestari, U. I., & et al. (2021). Penerapan Metode K-Nearest Neighbor untuk Sistem Pendukung Keputusan Identifikasi Penyakit Diabetes Melitus. *Jurnal Teknologi Informasi Dan Komputer*, 7(2), 45–52.
- [9] Zhang, Y., Li, J., & Wang, X. (2020). Algoritma K-Nearest Neighbor (KNN) dalam Tugas Klasifikasi. *Journal of Machine Learning*, 27(1), 89–102.
- [10] Rahman, M., Ahmed, T., & Chowdhury, F. (2020). Performance analysis of distance metrics in KNN for medical data classification using mobile applications. *Journal of Medical Systems*, 44(2), 56. <https://doi.org/10.1007/s10916-020-1537-5>
- [11] American Diabetes Association. (2021). Standards of medical care in diabetes–2021 abridged for primary care providers. *Clinical Diabetes*, 39(1), 14–43. <https://doi.org/10.2337/cd20-0105>
- [12] Cheung, B. M., & Li, C. (2022). Diabetes and hypertension: Is there a common metabolic pathway? *Current Atherosclerosis Reports*, 14(2), 160–166. <https://doi.org/10.1007/s11883-012-0227>
- [13] Kirkman, M. S., Briscoe, V. J., Clark, N., Florez, H., Haas, L. B., Halter, J. B., ... & Swift, C. S. (2012). Diabetes in older adults: A consensus report. *Journal of the American Geriatrics Society*, 60(12), 2342–2356. <https://doi.org/10.1111/jgs.12035>
- [14] Lyssenko, V., & Laakso, M. (2023). Genetic screening for the risk of type 2 diabetes: Worthless or valuable? *Diabetes Care*, 36(Suppl. 2), S120–S126. <https://doi.org/10.2337/dcS13-2007>
- [15] Pressman, R. S. (2020). Pengujian black box untuk aplikasi web dan pengujian fungsional. *Software Quality Journal*, 26(5), 411–428

