Perbandingan Performa Model Klasifikasi Random Forest dan XGBoost untuk Deteksi Dini Diabetes Berdasarkan Teknik Imputasi

Cindy Viona Dzul Qurnayn[1], Ricky Eka Putra[2]

[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya
[1]cindy.19039@mhs.unesa.ac.id
[2]rickyeka@unesa.ac.id

***Abstrak*— Missing value pada dataset medis dapat menurunkan kualitas data dan memengaruhi performa model machine learning dalam prediksi penyakit. Pada Pima Indians Diabetes Dataset, beberapa fitur seperti glucose, bloodpressure, skinthickness, insulin, dan BMI memiliki nilai 0 yang dianggap tidak valid secara medis. Penelitian ini bertujuan untuk menganalisis pengaruh missing value terhadap performa model klasifikasi diabetes, membandingkan teknik imputasi, serta menentukan kombinasi model terbaik untuk implementasi sistem prediksi diabetes berbasis web. Penelitian menggunakan model Random Forest, XGBoost, dan Stacking dengan teknik imputasi KNN, MICE, dan Autoencoder (AE). Evaluasi dilakukan menggunakan K-Fold CV dengan k=5, 10, dan 20 serta metrik AUC-ROC, accuracy, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa penanganan missing value berpengaruh signifikan terhadap performa model. Teknik imputasi KNN dan MICE menghasilkan performa yang lebih baik dibandingkan AE. Kombinasi terbaik diperoleh pada model XGB_KNN dengan nilai AUC-ROC sebesar 0.8458, accuracy 0.7772, precision 0.7249, dan F1-score 0.649 pada k=10. Model terbaik kemudian berhasil diimplementasikan ke dalam sistem prediksi diabetes berbasis web menggunakan Flask.**

***Kata Kunci*— Diabetes melitus, Missing value, Imputasi data, K-Fold Cross Validation, Machine learning.**

I. PENDAHULUAN

Diabetes melitus adalah kelompok penyakit metabolik yang ditandai dengan kondisi kadar gula (glukosa) dalam darah terlalu tinggi atau melebihi batas normal akibat gangguan pada sekresi insulin, kinerja insulin, atau keduanya secara bersamaan [1]. Kondisi ini tidak hanya berdampak pada kualitas hidup penderita secara langsung, tetapi juga memicu rangkaian komplikasi serius yang meliputi penyakit kardiovaskular dan gagal ginjal. Menurut laporan terbaru International Diabetes Federation [2] jumlah penderita diabetes di seluruh dunia telah melampaui angka 537 juta jiwa pada tahun 2022, dan proyeksi menunjukkan angka tersebut berpotensi mencapai 783 juta jiwa pada tahun 2045 apabila tidak ada intervensi yang memadai.

Di Indonesia, situasi ini tidak kalah mengkhawatirkan. Data dari Kementrian Kesehatan RI mengungkapkan bahwa Indonesia menempati posisi kelima dalam daftar negara dengan jumlah penderita diabetes terbanyak di dunia, dengan estimasi sekitar 19,5 juta penderita. Ironisnya, sebagian besar penderita baru menyadari kondisinya setelah memasuki stadium lanjut, ketika komplikasi telah terjadi dan biaya pengobatan menjadi jauh lebih tinggi [3]. Kesenjangan antara angka kejadian dan keterlambatan diagnosis ini menjadikan deteksi dini sebagai kebutuhan yang sangat mendesak.

Perkembangan teknologi kecerdasan buatan, khususnya dalam machine learning, membuka peluang besar untuk membantu proses skrining dan deteksi diabetes secara otomatis dan efisien. Berbagai algoritma klasifikasi telah diujicobakan untuk keperluan ini, dan dua di antaranya yang paling banyak mendapat perhatian adalah Random Forest (RF) dan XGBoost. Referensi [4] yang pertama kali menunjukkan bahwa RF bekerja berdasarkan prinsip ensemble learning dengan membangun sejumlah besar pohon keputusan secara acak dan menggabungkan hasilnya untuk mendapatkan prediksi yang lebih stabil dan robust. Sementara itu, referensi [5] menunjukkan XGBoost dikembangkan menggunakan pendekatan gradient boosting yang dioptimalkan, menjadikannya salah satu algoritma dengan performa terdepan dalam berbagai kompetisi data science.

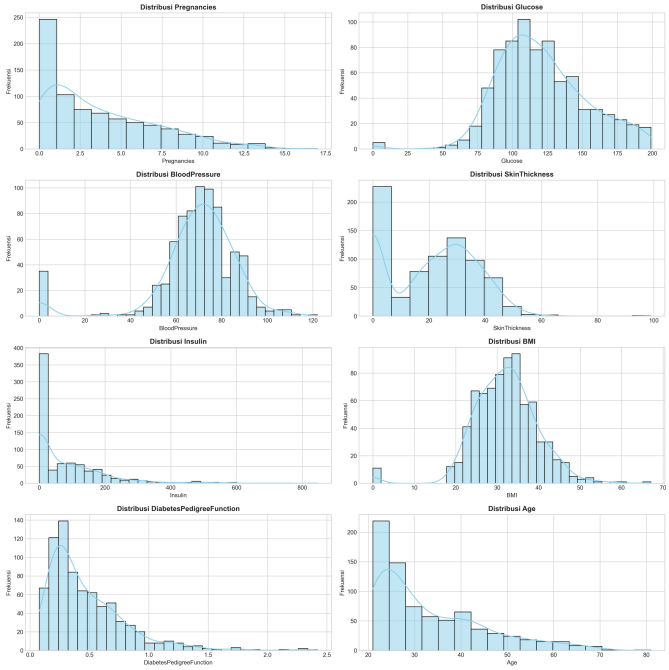
Namun, salah satu tantangan dalam penerapan model machine learning pada data medis adalah masalah missing value. Dataset klinis di dunia nyata hampir selalu mengandung ketidaklengkapan data akibat berbagai faktor, salah satunya keterbatasan alat diagnosis dalam pemeriksaan tertentu. Pima Indians Diabetes Dataset (PIDD), yang menjadi objek penelitian ini, merupakan contoh nyata dari permasalahan tersebut. Dataset ini memiliki nilai nol (0) pada beberapa fitur klinis seperti kadar glukosa, tekanan darah, ketebalan lipatan kulit, kadar insulin, dan BMI yang secara biologis tidak mungkin terjadi pada manusia hidup, sehingga harus diperlakukan sebagai missing value yang memerlukan penanganan khusus yaitu imputasi.

Sehingga fokus penelitian ini adalah pada pengembangan sistem deteksi dini diabetes menggunakan algoritma machine learning seperti RF dan XGBoost dengan perbandingan performa berdasarkan berbagai metode handling missing data. Selain penggunaan metode imputasi, integrasi preprocessing dengan ensemble stacking dan optimasi hyperparameter memiliki kemungkinan deteksi dini menjadi lebih robust terhadap data yang imbalanced seperti data diabetes. Sehingga dibutuhkan model machine learning yang andal untuk menangani data yang tidak lengkap di dunia nyata. Diharapkan penelitian ini dapat memberi wawasan pertimbangan dalam pemilihan model untuk diaplikasikan pada sistem deteksi dini diabetes di dunia nyata.

II. METODE PENELITIAN

A. Dataset dan Pra-pemrosesan Awal

Dataset yang digunakan dalam penelitian ini adalah Pima Indians Diabetes Dataset (PIDD), yang tersedia secara publik melalui UCI Machine Learning Repository. Dataset ini memuat 768 rekam data dengan 9 atribut: 8 fitur prediktor dan 1 label kelas biner (Outcome). Distribusi kelas menunjukkan rasio distribusi target sebesar 65,1%:34,9%, yaitu 500 data dengan outcome 0 atau tidak diabetes dan 268 data dengan outcome 1 atau diabetes.



Gbr. 1 Distribusi fitur PIDD

Pada tahap pra-pemrosesan, langkah pertama yang dilakukan adalah identifikasi nilai nol tidak valid, yaitu seperti yang dipresentasikan pada Gbr. 1, terdapat 5 variabel yang memiliki missing value dalam PIDD yaitu, 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', dan 'BMI'. Pada Tabel I berikut menunjukkan ketidaklengkapan data dimana dataset ini memiliki nilai 0 (nol) pada beberapa atribut yang secara medis tidak mungkin bernilai nol, sehingga diperlakukan sebagai missing value dalam dataset. Variabel Insulin dan SkinThickness memiliki proporsi missing value tertinggi masing-masing sebesar 48,7% dan 29,56%.

TABEL I
REPRESENTASI MISSING VALUE DALAM PIDD

No	Variabel	Jumlah nilai 0	Persentase missing	Keterangan
1.	Glucose	5	0,65%	Glukosa darah tidak mungkin bernilai 0
2.	BloodPressure	35	4,56%	Tekanan darah tidak mungkin bernilai 0
3.	SkinThickness	227	29,56%	Ketebalan lipatan kulit tidak mungkin bernilai 0
4.	Insulin	374	48,70%	Kadar insulin tidak mungkin bernilai 0
5.	BMI	11	1,43%	BMI tidak mungkin bernilai 0

Bergantung pada skenario yang dijalankan, nilai 0 pada kelima fitur tersebut dikonversi menjadi NaN sebelum dilakukan imputasi atau penghapusan. Fitur Pregnancies tidak dimasukkan dalam daftar ini karena nilai 0 secara valid berarti pasien belum pernah hamil.

B. Desain Eksperimen

Keseluruhan eksperimen dalam penelitian ini dirancang dalam empat skenario, berikut adalah desain ekspe

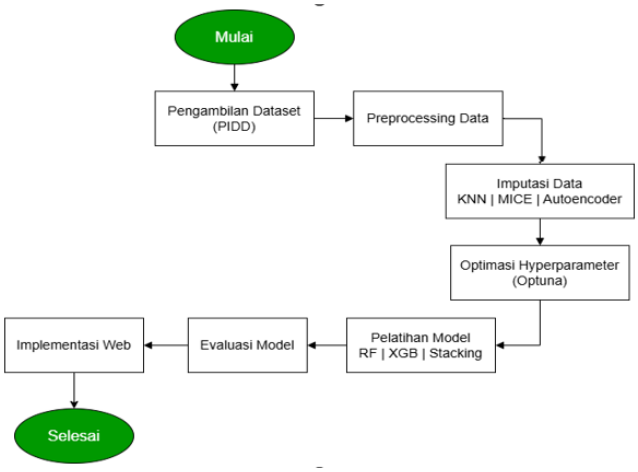
1) Skenario 1 Baseline

Pada skenario pertama, model dilatih langsung menggunakan dataset asli tanpa penanganan apapun pada nilai 0. Tujuannya adalah untuk memperoleh baseline performa yang dapat dijadikan acuan komparatif bagi skenario berikutnya. Evaluasi dilakukan menggunakan K-Fold Cross Validation dengan k=5, 10, dan 20.

2) Skenario 2 Remove Missing Value

Nilai 0 pada `invalid_features` terlebih dahulu dikonversi menjadi NaN, kemudian semua baris yang mengandung minimal satu NaN dihapus menggunakan fungsi `dropna()`. Proses ini mereduksi dataset menjadi sekitar 392 data yang valid. Model kemudian dilatih dan dievaluasi dengan K-Fold k=5, 10, dan 20. Skenario ini merepresentasikan pendekatan penghapusan data yang seringkali digunakan dalam praktik, namun memiliki risiko kehilangan informasi yang signifikan.

3) Skenario 3 Imputasi



Gbr. 2 Diagram alur penelitian

Skenario ketiga merupakan inti dari penelitian ini, seperti pada Gbr. 2 yaitu alur penelitian bahwa setelah konversi nilai 0 menjadi NaN, selanjutnya dataset diimputasi dengan tiga teknik imputasi (KNN, MICE, AE) sehingga menghasilkan 3 dataset. Sebelum dilakukan

pelatihan model, Optuna menjalankan 30 trial untuk menemukan kombinasi hyperparameter terbaik pada kombinasi model dan imputasi yang akan diuji. Selanjutnya split data train dan data test dengan K-Fold k=5, 10, dan 20 yang kemudian model dilatih dengan hyperparameter optimal dan dievaluasi pada fold sesuai nilai k. Penggunaan Optuna dalam penelitian ini berdasarkan keunggulan Optuna dalam efisiensinya yang mampu menemukan hyperprameter optimal dalam jumlah trial yang jauh lebih sedikit dibandingkan GridSearch. Referensi [8] mendemonstrasikan bahwa pendekatan TPE dalam Optuna secara konsistem menghasilkan performa yang lebih baik dibandingkan metode optimasi konvensional pada berbagai benchmark. Keseluruhan proses ini menghasilkan 9 kombinasi (3 model × 3 imputasi) yang masing-masing dievaluasi pada 3 nilai k, menghasilkan total 27 skenario evaluasi. Tiga model dan imputer terbaik disimpan menggunakan `joblib` untuk digunakan pada skenario selanjutnya.

4) Skenario 4

Tiga kombinasi dengan nilai AUC-ROC tertinggi dari Skenario 3 dipilih dan dimuat kembali. Evaluasi pada dataset dengan nilai missing value diulang menggunakan nilai k yang sesuai dengan k pada saat model tersebut mencapai performa terbaik. Skenario ini berfungsi sebagai konfirmasi final bahwa performa terbaik yang diraih pada Skenario 3 dapat direproduksi secara konsisten.

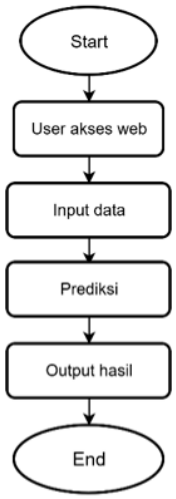
C. *Optimasi Hyperparameter dengan Optuna*

Konfigurasi pencarian hyperparameter yang didefinisikan untuk setiap model adalah sebagai berikut:

- RF: ` n_estimators`, `max_depth`
- XGBoost: ` n_estimators`, `max_depth`, `learning_rate`
- Stacking:`rf_ n_estimators`, `xgb_ n_estimators`

Metrik yang digunakan sebagai fungsi objektif Optuna adalah AUC-ROC rata-rata dari validasi silang internal. Setiap trial menggunakan stratified split untuk mempertahankan proporsi kelas yang seimbang.

D. *Implementasi Aplikasi Web*



Gbr. 3 Flowchart aplikasi web deteksi diabetes

Model terbaik yang diidentifikasi dari skenario 4 diimplementasikan dalam sebuah aplikasi web menggunakan framework Flask, berikut adalah alur aplikasi web. Pada Gbr. 3 adalah alur flowchart aplikasi web yang memungkinkan pengguna memasukkan delapan parameter klinis dan mendapatkan hasil prediksi beserta skor probabilitas secara real-time. Antarmuka pengguna dibangun menggunakan Bootstrap untuk memastikan tampilan yang responsif. Model dan imputer tersimpan dalam format `.joblib` dan dimuat saat aplikasi diinisialisasi.

III. HASIL DAN PEMBAHASAN

A. *Hasil Evaluasi Skenario 1 Baseline*

Berikut Tabel II adalah hasil evaluasi performa model pada skenario baseline dengan StratifiedK-Fold CV k=5, 10, dan 20.

TABEL II
HASIL EVALUASI SKENARIO 1 BASELINE

K	Model	AUC	Acc	Prec	Rec	F1
5	RF	0.8250	**0.7643**	0.6936	0.5945	0.636
	XGB	0.8006	0.7539	0.6518	**0.6408**	**0.643**
	Stacking	0.8248	0.7603	0.6947	0.5756	0.624
10	RF	0.8286	0.7577	0.6728	0.5878	0.623
	XGB	0.7901	0.7408	0.6437	0.6038	0.619
	Stacking	**0.8255**	0.7603	0.6893	0.5725	0.621
20	RF	0.8242	0.7629	**0.6984**	0.5741	0.618
	XGB	0.8034	0.7292	0.6245	0.5870	0.597
	Stacking	0.8229	0.7564	0.6981	0.5526	0.605

RF secara konsisten mendominasi nilai AUC-ROC pada skenario baseline, dengan nilai tertinggi mencapai 0,8286 pada k=10. Ini mengindikasikan bahwa sifat ensemble dari RF yang membangun banyak pohon independen secara paralel memberikan kestabilan yang lebih baik saat berhadapan dengan data yang mengandung noise, termasuk nilai 0 yang tidak valid. XGB pada k=10 justru mencatatkan nilai AUC-ROC terendah (0,7901) di antara semua model dan nilai k. Hal ini menunjukkan bahwa mekanisme gradient boosting pada XGBoost lebih sensitif terhadap anomali dalam data pelatihan,

sehingga keberadaan nilai 0 yang tidak valid mempengaruhi arah gradien secara negatif.

Sementara itu, stacking menunjukkan performa yang berfluktuasi di antara kedua model dasarnya. Meskipun secara teori stacking seharusnya mampu mengkombinasikan keunggulan RF dan XGBoost, tanpa penanganan missing value dan tanpa optimasi hyperparameter, meta learner mungkin tidak mendapatkan sinyal yang cukup informatif dari prediksi base learner yang juga terkontaminasi oleh missing value. Selanjutnya nilai Recall yang secara umum lebih rendah dibandingkan Precision menunjukkan bahwa semua model pada baseline cenderung berhati-hati dalam mengklasifikasikan kasus positif (diabetes), sehingga lebih sering melewatkan kasus positif yang sebenarnya (false negative tinggi).

B. *Hasil Evaluasi Skenario 2 Remove Missing Value*

Berikut Tabel III adalah hasil evaluasi performa model pada skenario remove yang menghapus seluruh baris data missing dan dievaluasi dengan Stratified K-Fold CV k=5, 10, dan 20.

TABEL III
HASIL EVALUASI SKENARIO 2 REMOVE

K	Model	AUC	Acc	Prec	Rec	F1
5	RF	0.8378	0.7653	0.6679	0.5870	0.622
	XGB	0.8151	0.7321	0.5955	0.5967	0.587
	Stacking	0.8371	0.7526	0.6424	0.5789	0.603
10	RF	0.8379	0.7728	**0.6784**	0.6061	0.635
	XGB	0.8335	**0.7782**	0.6705	**0.6564**	**0.654**
	Stacking	0.8429	0.7756	0.6930	0.6047	0.638
20	RF	**0.8511**	0.7736	0.6528	0.6103	0.616
	XGB	0.8248	0.7503	0.6103	0.5995	0.592
	Stacking	0.8459	0.7736	0.6782	0.6116	0.623

Perbandingan antara skenario 1 dan 2 mengungkapkan temuan yang secara teoritis sesuai. Penghapusan baris dengan missing value secara umum meningkatkan nilai AUC-ROC pada sebagian besar konfigurasi model dan nilai k. Nilai AUC-ROC tertinggi pada skenario 2 diraih oleh RF pada k=20 (0,8511), menunjukkan peningkatan yang berarti dibandingkan baseline RF (0,8286 pada k=10). Namun, peningkatan ini harus diinterpretasikan dengan hati-hati. Proses penghapusan baris mereduksi dataset dari 768 menjadi 392 baris data dimana pengurangannya mencapai 49% yang dapat meningkatkan variance estimasi, sehingga hasil evaluasi menjadi kurang stabil dan berpotensi overestimating kemampuan generalisasi model. Meski begitu, pada skenario ini XGB menunjukkan perbaikan yang lebih dramatis dibandingkan skenario 1, terutama pada k=10 (AUC-ROC meningkat dari 0,7901 ke 0,8335). Ini semakin mengonfirmasi bahwa XGB memang lebih sensitif terhadap kualitas data, dan data yang lebih bersih memberinya keuntungan yang lebih besar dibandingkan RF.

C. *Hasil Evaluasi Skenario 3 Imputasi*

Skenario ini merupakan bagian paling substansial dari penelitian. Sebelum melakukan pelatihan model, dataset terlrbih dahulu diimputasi dengan teknik imputasi berikut:

1) Imputasi KNN-imputer: menggunakan parameter n_neighbors=5.

2) Imputasi MICE: menggunakan class `IterativeImputer` *dengan* `max_iter=10`.

3) Imputasi AE: Pada proses imputasi AE dilakukan pendefinisian autoencoder dengan pytorch.

TABEL IV
HASIL IMPUTASI

Set	Preg	Glu	BP	ST	Ins	BMI	DPF
Asli	10	115	NaN	NaN	NaN	35.3	0.134
KNN	10	115	77.6	34.4	132.6	35.3	0.134
MICE	10	115	72.95	31.59	136.27	35.3	0.134
AE	10	115	72	30.67	151.3	35.3	0.134
Asli	7	105	NaN	NaN	NaN	NaN	0.305
KNN	7	105	66	26.4	150.8	31.8	0.305
MICE	7	105	69.25	27.6	110.67	31.92	0.305
AE	7	105	71.91	29.3	143.57	33.41	0.305

Pada Tabel IV terlihat bahwa data ke-8 dan data ke-50, jumlah missing value lebih banyak karena beberapa atribut penting seperti tekanan darah, ketebalan kulit, insulin, dan BMI mengalami missing value secara bersamaan. Pada kondisi seperti ini, perbedaan antar metode imputasi menjadi lebih terlihat. Metode KNN menghasilkan nilai yang cenderung mendekati rata-rata lokal berdasarkan tetangga terdekat, misalnya pada data ke-8 menghasilkan Blood Pressure sebesar 77.6 dan insulin sebesar 132.6. Metode MICE menghasilkan nilai yang lebih bervariasi seperti Blood Pressure sebesar 72.95 dan insulin sebesar 136.27 karena metode ini memanfaatkan hubungan regresi antarfitur. Sementara itu, Autoencoder menghasilkan nilai yang relatif lebih tinggi pada beberapa atribut, seperti insulin sebesar 151.3 pada data ke-8 dan BMI sebesar 33.41 pada data ke-50. Hal ini menunjukkan bahwa AE mencoba merekonstruksi pola data secara menyeluruh melalui representasi laten pada neural network.

Sembilan kombinasi model dan imputasi yang dievaluasi pada tiga nilai k, dengan setiap kombinasi telah melalui optimasi hyperparameter menggunakan Optuna. Untuk setiap studi fungsi optimize melakukan n_trials=30 dan n_jobs=1 untuk melakukan sebanyak 30 percobaan (fungsi berjalan paralel). Setelah optimasi selesai, fungsi mengembalikan study.best_params (parameter terbaik yang ditemukan). Kemudian optimasi dijalankan pada setiap kombinasi model dan imputed_datasets sebagai data imputasi dan menyimpan parameter terbaik dalam dictionary best_params_all. Berikut adalah hasil konfigurasi hyperparameter Optuna.

TABEL V
HASIL OPTIMASI HYPERPARAMETER OPTUNA

Imp	RF		Stacking	
	n_estim	max_depth	rf_n_estim	xgb_n_estim
KNN	55	5	117	111

MICE	145	3	61	54
AE	120	7	107	111
Imp	XGB			
	n_estimators	max_depth	learning_rate	
KNN	75	3	0.03340345547890528	
MICE	127	3	0.04576447083777803	
AE	95	5	0.06540251149590108	

Dari Tabel V dapat dilihat bahwa RF dengan imputasi MICE membutuhkan jumlah pohon yang lebih banyak dengan kedalaman yang lebih dangkal, mengindikasikan bahwa data hasil imputasi MICE memerlukan diversitas model yang lebih besar namun dengan kompleksitas individual yang terbatas. Sebaliknya, RF dengan AE menggunakan kedalaman yang lebih dalam dengan jumlah pohon moderat, kemungkinan karena representasi laten yang dihasilkan AE lebih informatif sehingga setiap pohon dapat menangkap pola yang lebih dalam. Di lain sisi, XGB dengan imputasi KNN menghasilkan learning rate yang paling rendah dengan jumlah estimator yang sedikit, menunjukkan bahwa model belajar secara lambat namun stabil. Sebaliknya, XGB dengan AE menggunakan learning rate tertinggi, yang konsisten dengan temuan bahwa representasi laten AE cenderung memiliki struktur yang lebih halus dan teratur, sehingga model dapat belajar dengan langkah yang lebih besar tanpa risiko overfitting yang berlebihan.

Setelah pelatihan model dengan parameter optimal, model dievaluasi dengan K-Fold CV. Berikut adalah hasil evaluasi skenario 3.

1) Hasil evaluasi model pada k=5

Berikut Tabel VI adalah hasil evaluasi performa model pada skenario imputasi dengan nilai k=5.

TABEL VI
HASIL EVALUASI SKENARIO 3 K=5

Kombinasi	AUC	Acc	Prec	Rec	F1
RF_KNN	0.8401	0.7707	0.7060	0.5969	0.644
XGB_KNN	0.8377	**0.7759**	0.7060	0.6193	0.658
Stack_KNN	**0.8403**	0.7720	0.7085	0.5969	0.646
RF_MICE	0.8383	0.7551	**0.7094**	0.5187	0.595
XGB_MICE	0.8343	0.7733	0.6898	**0.6416**	**0.664**
Stack_MICE	0.8381	0.7629	0.6955	0.5783	0.629
RF_AE	0.8363	0.7524	0.6654	0.5781	0.617
XGB_AE	0.8196	0.7296	0.6609	0.6042	0.631
Stack_AE	0.8327	0.7459	0.6946	0.5595	0.619

Secara keseluruhan, hasil evaluasi dengan k=5 pada Tabel VI menunjukkan bahwa metode imputasi KNN dan MICE mampu meningkatkan performa model klasifikasi dibandingkan skenario baseline. Imputasi KNN menunjukkan performa yang paling konsisten pada hampir seluruh model dengan nilai metrik yang relatif seimbang. Sementara itu, imputasi MICE menunjukkan kemampuan yang sangat baik dalam meningkatkan recall, khususnya pada kombinasi XGB_MICE. Di sisi lain, imputasi Autoencoder masih menghasilkan performa yang lebih rendah dibanding dua metode lainnya. Berdasarkan seluruh kombinasi model, XGB_KNN dan XGB_MICE menjadi kombinasi yang paling baik karena mampu menghasilkan keseimbangan performa yang optimal antara kemampuan diskriminatif model, sensitivitas deteksi kasus diabetes, serta kestabilan klasifikasi.

2) Hasil evaluasi model pada k=10

Berikut Tabel VII adalah hasil evaluasi performa model pada skenario imputasi dengan nilai k=10.

TABEL VII
HASIL EVALUASI SKENARIO 3 K=10

Kombinasi	AUC	Acc	Prec	Rec	F1
RF_KNN	0.8396	0.7655	0.7048	0.5970	0.641
XGB_KNN	**0.8458**	**0.7772**	**0.7249**	0.5972	0.649
Stack_KNN	0.8451	0.7721	0.7097	0.6047	0.648
RF_MICE	0.8435	0.7603	0.7221	0.5301	0.605
XGB_MICE	0.8400	0.7759	0.6985	**0.6379**	**0.662**
Stack_MICE	0.8437	0.7668	0.7055	0.5897	0.638
RF_AE	0.8343	0.7655	0.7034	0.5898	0.636
XGB_AE	0.8263	0.7538	0.6566	0.6115	0.631
Stack_AE	0.8345	0.7564	0.6901	0.5596	0.614

Secara keseluruhan, hasil evaluasi dengan k=10 pada Tabel VI menunjukkan bahwa metode imputasi KNN dan MICE mampu meningkatkan performa model secara lebih optimal dibandingkan imputasi Autoencoder. Imputasi KNN menghasilkan performa yang paling konsisten pada hampir seluruh metrik evaluasi, khususnya pada kombinasi XGB_KNN yang memperoleh nilai AUC-ROC, accuracy, dan precision tertinggi. Sementara itu, imputasi MICE menunjukkan kemampuan yang sangat baik dalam meningkatkan recall dan F1-score, terutama pada kombinasi XGB_MICE. Hal ini menunjukkan bahwa metode MICE mampu mempertahankan sensitivitas model dalam mendeteksi kasus diabetes. Di sisi lain, imputasi Autoencoder masih menunjukkan performa yang relatif lebih rendah sehingga belum menjadi metode imputasi yang optimal pada penelitian ini.

3) Hasil evaluasi model pada k=20

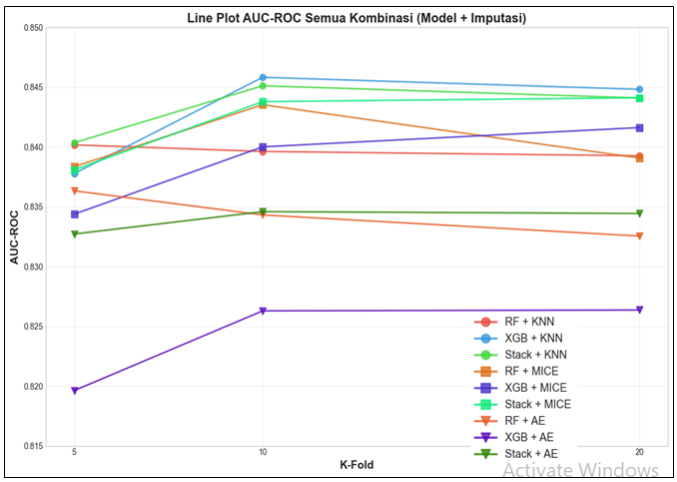
Berikut Tabel VIII adalah hasil evaluasi performa model pada skenario imputasi dengan nilai k=20.

TABEL VIII
HASIL EVALUASI SKENARIO 3 K=20

Kombinasi	AUC	Acc	Prec	Rec	F1
RF_KNN	0.8392	0.7576	0.6823	0.5826	0.617
XGB_KNN	**0.8448**	0.7705	0.7001	0.6038	0.641
Stack_KNN	0.8441	0.7680	0.6945	0.5931	0.633
RF_MICE	0.8391	0.7577	**0.7096**	0.5228	0.592
XGB_MICE	0.8416	**0.7731**	0.6924	0.6341	**0.656**
Stack_MICE	0.8441	0.7719	**0.7096**	0.5893	0.638
RF_AE	0.8325	0.7486	0.6645	0.5557	0.599
XGB_AE	0.8263	0.7692	0.6791	**0.6406**	0.653
Stack_AE	0.8344	0.7628	0.6877	0.5701	0.617

Secara keseluruhan, hasil evaluasi dengan k=20 pada Tabel VIII menunjukkan bahwa imputasi KNN dan MICE tetap menjadi metode imputasi yang paling efektif dalam meningkatkan performa model klasifikasi diabetes. Imputasi KNN menunjukkan performa yang paling stabil

pada hampir seluruh kombinasi model, khususnya pada kombinasi XGB_KNN yang berhasil mempertahankan nilai AUC-ROC, accuracy, dan precision yang tinggi. Sementara itu, imputasi MICE menunjukkan kemampuan yang sangat baik dalam meningkatkan recall dan F1-score, terutama pada kombinasi XGB_MICE.



Gbr. 4 Lineplot AUC-ROC skenario 3 terhadap K-Fold

Pada Gbr. 4 menunjukkan perbandingan antara k=5, k=10, dan k=20 memberikan wawasan tentang trade-off antara bias dan variance dalam estimasi performa. Secara umum, k=10 menghasilkan estimasi yang paling stabil dan sering kali menunjukkan nilai AUC-ROC tertinggi di antara ketiga opsi. Ini sejalan dengan panduan umum dalam literatur machine learning yang merekomendasikan k=10 sebagai keseimbangan optimal antara bias estimasi yang cenderung tinggi pada k kecil dan variance yang cenderung tinggi pada k besar [7].

Secara keseluruhan, KNN menghasilkan nilai AUC-ROC rata-rata yang tertinggi di antara ketiga teknik imputasi, khususnya ketika dikombinasikan dengan model XGB dan Stacking. Temuan ini sejalan dengan referensi [6] yang menunjukkan keunggulan KNN dalam mempertahankan struktur lokal data. Sedangkan dari perspektif model, XGBt secara umum mendominasi pada metrik AUC-ROC ketika dikombinasikan dengan imputasi KNN dan MICE. Ini bertolak belakang dengan performa XGB pada skenario 1, yang justru paling rendah. Temuan ini menunjukkan bahwa sensitivitas XGBoost terhadap kualitas data bersifat dua arah, yaitu kualitas data yang buruk sangat menghambat XGBoost, namun kualitas data yang baik memberinya keunggulan yang signifikan. Karakteristik ini konsisten dengan sifat gradient boosting yang sensitif terhadap noise karena membangun model secara sekuensial berdasarkan error [5].

D. *Analisis Perbandingan Hasil Teknik Imputasi*

Secara keseluruhan, KNN menghasilkan nilai AUC-ROC rata-rata tertinggi di antara ketiga teknik imputasi, khususnya ketika dikombinasikan dengan model XGBoost dan Stacking. Referensi [6] menunjukkan temuan ini sejalan bahwa keunggulan KNN dalam mempertahankan struktur lokal data. MICE menunjukkan performa yang kompetitif, terutama dalam nilai F1-Score yaitu XGB_MICE mencatatkan F1 terbaik (0.664 pada k=5 dan 10). Ini mengindikasikan bahwa MICE mengidentifikasi kasus positif secara lebih seimbang. Namun, pada metrik AUC-ROC masih sedikit di bawah KNN. Autoencoder secara konsisten menunjukkan performa yang lebih rndah dibandingkan dua teknik imputasi lainnya pada hampir semua konfigurasi. Hal ini mungkin disebabkan oleh beberapa faktor, yaitu PIDD memiliki dimensi (8 fitur) yang relatif rendah, sehingga keunggulan deep learning dalam menangkap representasi nonlinear kompleks tidak maksimal dan jumlah data pelatihan (768 sampel) relatif kecil untuk melatih neural network secara optimal [9].

E. *Analisis Perbandingan Hasil Model Klasifikasi*

Secara umum, XGBoost mendominasi pada metrik AUC-ROC ketika dikombinaskan dengan teknik imputasi KNN dan MICE. Hal ini bertolak belakang dengan performa XGBoost pada skenario baseline yang justru paling rendah. Temuan ini mengilustrasikan secara jelas bahwa XGBoost memiliki sensitivitas yang tinggi terhadap kualitas data dimana jika kualitas data buruk maka akan menghambat XGBoost dan kualitas data yang baik memberikan keunggulan yang signifikan. Random Forest menunjukkan konsistensi yang lebih besar di seluruh skenario dan teknik imputasi. Standar deviasi AUC-ROC RF lebih kecil di berbagai konfigurasi dibandingkan XGBoost dan Stacking yang merupakan bukti nyata dari kelebihan inheren dalam robustness. Namun, RF jarang mencapai performa tertinggi walaupun dengan teknik imputasi terbaik sekalipun. Stacking tidak selalu mengungguli kedua base learnernya. Pada sebagian besar konfigurasi, performanya berada di antara RF dan XGBoost. Hal ini bisa terjadi ketika base learner sudah cukup kuat dan berkorelasi tinggi satu sama lain dan meta learner kesulitan mengekstraksi nilai tambah yang signifikan yaitu Logistic Regression sebagai meta learner dapat dinilai terlalu sederhana untuk menangkap pola interaksi kompleks antara prediksi RF dan XGBoost.

F. *Hasil Evaluasi Skenario 4*

Berikut Tabel IX adalah hasil evaluasi performa kombinasi trained_model terbaik pada skenario 4 mengonfirmasi sepenuhnya bahwa performa yang dicapai pada skenario 3 dapat direproduksi secara konsisten. Tidak ada penurunan performa yang signifikan antara skenario 3 dan skenario 4, yang mengindikasikan bahwa pipeline penyimpanan dan pemuatan model berjalan dengan benar dan tidak terdapat data leakage dalam proses pelatihan. Kombinasi XGB + KNN pada k=10 ditetapkan sebagai model terbaik untuk diimplementasikan dalam aplikasi web, dengan pertimbangan nilai AUC-ROC tertinggi (0,8458), presisi yang sangat baik (0,7249), dan F1-score yang kompetitif (0,649).

TABEL IX
HASIL EVALUASI SKENARIO 4

K	Kombinasi	AUC	Acc	Prec	Rec	F1
10	XGB_KNN	0.8458	0.7772	0.7249	0.597	0.649

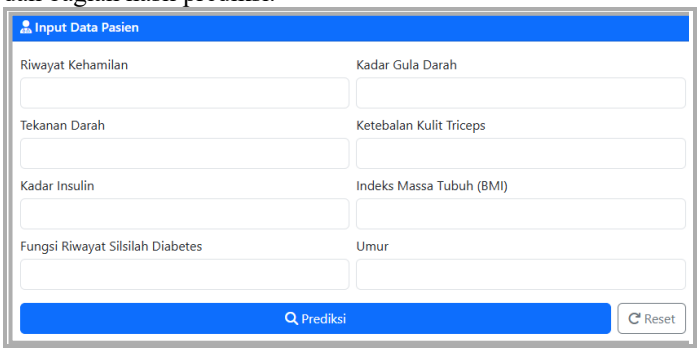
10	Stack_KNN	0.8451	0.7721	0.7097	0.605	0.648
20	XGB_KNN	0.8448	0.7705	0.7001	0.604	0.641

G. *Implementasi Aplikasi Web*

Kombinasi model terbaik (XGB + KNN) diintegrasikan ke dalam aplikasi web berbasis Flask yang dirancang untuk kemudahan penggunaan. Pada penelitian ini, aplikasi web diakses melalui server lokal http://localhost:5000 kemudian akan menampilkan tampilan web seperti pada gambar berikut.

Gbr. 5 Tampilan aplikasi web deteksi dini diabetes

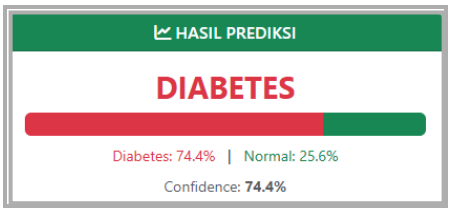
Pada Gbr. 5 adalah tampilan aplikasi web untuk deteksi dini diabetes yang berisi informasi tentang pilihan model yang dapat digunakan dalam aplikasi web, form input data pasien, dan bagian hasil prediksi.



Gbr. 6 Form input data pasien

Seperti terlihat pada Gbr. 6 aplikasi menerima input delapan parameter klinis melalui formulir HTML yang dilengkapi validasi sisi klien dan server. Validasi mencakup pengecekan bahwa nilai untuk `invalid_features` (Glucose, BloodPressure, SkinThickness, Insulin, BMI) tidak boleh bernilai 0, serta tipe data integer untuk Pregnancies, Glucose, BloodPressure, Insulin, dan Age. Setelah input tervalidasi, data pengguna diproses melalui pipeline yang sama dengan yang digunakan saat pelatihan:

- Nilai input yang kosong atau tidak valid diidentifikasi.
- Model XGBoost menghasilkan prediksi kelas biner dan skor probabilitas.
- Hasil ditampilkan kepada pengguna dalam format yang mudah dipahami, yaitu "Diabetes" atau "No Diabetes" beserta persentase probabilitas seperti pada Gbr. 7.



Gbr. 7 Hasil prediksi diabetes (DIABETES)

Implementasi aplikasi web ini mendemonstrasikan kemampuan model klasifikasi diabetes yang telah dikembangkan dalam penelitian. Aplikasi ini menyediakan antarmuka yang user-friendly untuk melakukan prediksi. Meskipun tidak untuk penggunaan klinis real-time, aplikasi ini memiliki nilai edukatif sebagai media pembelajaran machine learning dan implementasi deployment model.

IV. KESIMPULAN

Penelitian ini telah berhasil menjawab seluruh tujuan yang ditetapkan di awal melalui serangkaian eksperimen yang sistematis dan terukur. Beberapa kesimpulan utama dapat ditarik dari keseluruhan proses penelitian:

1. Keberadaan missing value yang tidak ditangani terbukti menurunkan performa model secara signifikan, terutama untuk XGBoost yang lebih sensitif terhadap kualitas data. Pengabaian missing value pada skenario 1 menghasilkan AUC-ROC terendah secara keseluruhan untuk XGBoost (0,7901 pada k=10), jauh di bawah potensi yang dicapainya setelah data dibersihkan.
2. KNN terbukti sebagai teknik imputasi paling efektif untuk PIDD di antara ketiga metode yang diujikan, menghasilkan nilai AUC-ROC tertinggi pada hampir semua konfigurasi model. Keunggulan ini kemungkinan besar dipengaruhi oleh karakteristik PIDD yang memiliki korelasi struktural kuat antar-fitur klinis, yang dimanfaatkan secara optimal oleh pendekatan berbasis tetangga terdekat.
3. MICE menunjukkan keunggulan spesifik pada metrik F1-Score, mengindikasikan kemampuannya yang lebih baik dalam menyeimbangkan presisi dan recall. Ini menjadikan MICE pilihan yang layak dipertimbangkan ketika tujuan utama adalah meminimalkan false negative dalam konteks skrining klinis.
4. Autoencoder Imputation, meskipun secara teoritis paling canggih, tidak berhasil mengungguli metode konvensional pada dataset berdimensi rendah seperti PIDD. Potensinya mungkin lebih terlihat pada dataset yang lebih besar dan berdimensi tinggi.
5. Kombinasi XGBoost dengan KNN pada k=10 menghasilkan performa terbaik secara keseluruhan dengan AUC-ROC 0,8458, akurasi 0,7772, presisi 0,7249, dan F1-score 0,649. Ini adalah model yang direkomendasikan untuk aplikasi deteksi dini diabetes berbasis PIDD.
6. Optimasi hyperparameter menggunakan Optuna memberikan kontribusi yang terukur terhadap

peningkatan performa. Perbandingan antara Skenario 1 (tanpa tuning) dan Skenario 3 (dengan tuning Optuna) menunjukkan peningkatan AUC-ROC rata-rata sekitar 0,01–0,02 poin untuk setiap model.

7. Implementasi model terbaik dalam aplikasi web Flask berhasil mengintegrasikan seluruh pipeline—dari validasi input, imputasi, hingga prediksi—ke dalam antarmuka yang mudah digunakan, menjembatani penelitian akademis dengan aplikasi praktis.

V. SARAN

Berdasarkan temuan dan keterbatasan yang diidentifikasi, beberapa arah penelitian yang dapat dikembangkan di masa mendatang meliputi:

1. Penelitian selanjutnya disarankan untuk menggunakan eksperimen ini pada dataset diabetes dari populasi yang berbeda atau dari rumah sakit lokal Indonesia untuk mengevaluasi kemampuan generalisasi model.
2. Penelitian ini hanya membandingkan tiga teknik imputasi, untuk pengembangan selanjutnya dapat mengeksplorasi arsitektur Autoencoder yang lebih canggih seperti Variational Autoencoder (VAE) atau GAIN (Generative Adversarial Imputation Network) yang secara teoritis lebih mampu menangani distribusi data medis yang kompleks.
3. Pengembangan aplikasi web ditambahkan fitur logging prediksi, dashboard statistik, dan integrasi dengan sistem rekam medis rumah sakit untuk penggunaan klinis, serta dapat dikembangkan menjadi aplikasi berbasis mobile (Android atau iOS).

VI. UCAPAN TERIMA KASIH

Puji syukur kehadirat Allah SWT atas segala rahmat dan hidayah-Nya, sehingga proses penelitian ini dapat terselesaikan dengan baik dan tepat waktu, menghasilkan penyusunan artikel jurnal ini. Keberhasilan penyelesaian artikel ini merupakan buah dari perjuangan mengatasi berbagai kendala penelitian, yang seluruhnya dapat dituntaskan berkat pertolongan-Nya.

Penghargaan dan terima kasih yang setinggi-tingginya disampaikan kepada berbagai pihak yang telah memberikan kontribusi dan dukungan berharga, baik secara moril maupun akademis. Rasa terima kasih yang utama ditujukan kepada dosen pembimbing, atas arahan, masukan konstruktif, dan bimbingan yang konsisten selama seluruh tahapan penelitian hingga penulisan artikel ini.Selain itu, penulis menyampaikan terima kasih yang tak terhingga kepada keluarga dan rekan-rekan yang senantiasa menjadi sumbermotivasi dan semangat dalam menyelesaikan studi dan karya tulis ini.

Semoga seluruh kebaikan, dukungan, dan ilmu yang telah diberikan mendapatkan balasan yang terbaik dari Allah SWT. Penulis menyadari bahwa artikel ini masih memiliki kekurangan, oleh karena itu, saran dan kritik yang membangun akan diterima dengan lapang dada demi perbaikan di masa mendatang. Artikel ini diharapkan dapat memberikan kontribusi yang bermanfaat dalam pengembangan ilmu di bidang Sistem Informasi, khususnya pada manajemen persediaan.

REFERENSI

[1] (2023) World Health Organization, ”Diabetes,”. [Online], https://www.who.int/newa-room/fact-sheets/detail/diabetes, tanggal akses: 28 Mei 2026.

[2] (2023) International Diabetes Federation, ”IDF Diabetes Atlas (11th ed.),”. [Online], https://www.diabetesatlas.org/, tanggal akses: 28 Mei 2026.

[3] (2023) Kementerian Kesehatan Republik Indonesia, ”Profil kesehatan Indonesia 2022,” [Online], https://www.kemkes.go.id/, tanggal akses: 28 Mei 2026.

[4] L. Breiman, ”Random Forests,” *Machine Learning,* vol. 45, hal. 5-32, 2001.

[5] T. Chen, C. Guestrin, ”XGBoost: A Scalable Tree Boosting System,” *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Fransisco, Association for Computing Machinery, 2016, hal. 785-794.

[6] M. Kokla, J. Virtanen, M. Kolehmainen, J. Paananen och K. Hanhineva, ”Random forest-based imputation outperforms other methods for imputing LC-MS metabolomics data: A comparative study,” *BMC Bioinformatics,* vol. 20, hal. 492, 2019.

[7] R. Kohavi, ”A study of cross-validation and bootstrap for accuracy estimation and model selection,” *Proc. 14th International Joint Conference on Artificial Intelligence - Volume 2*, Montreal, Morgan Kaufmann Publishers Inc., 1995, hal. 1137-1145.

[8] T. Akiba, S. Sano, T. Yanase, T. Ohta, dan M. Koyama, “Optuna: A Next-Generation Hyperparameter Optimization Framework,” *Proc. The 25th ACM SIGKDD International COneference on Knowledge Discovery and Data Mining, 2019, hal. 2623-2631.*

[9] L. Gondara, K. Wang, “MIDA: Multiple Imputation Using Denoising Autoncoder,” *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining,* hal. 260-272.