BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Kajian-kajian sebelumnya di bidang klasifikasi diabetes memperlihatkan kemajuan yang cukup berarti dalam pemanfaatan *machine learning* sebagai pendekatan untuk meningkatkan ketepatan diagnosis dan mendukung pengambilan keputusan klinis berbasis data.

Sejumlah penelitian telah mengkaji penerapan berbagai algoritma *machine learning*, di antaranya *Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbor (KNN)*, serta metode *ensemble learning* seperti *XGBoost*. Algoritma-algoritma tersebut pada umumnya diarahkan untuk menangani masalah klasifikasi biner dalam mendeteksi risiko diabetes berdasarkan data klinis dan indikator kesehatan lainnya. Di samping itu, tidak sedikit penelitian yang turut mengintegrasikan teknik-teknik optimasi seperti penyeimbangan data (*data balancing*) dan metode *ensemble* guna meningkatkan kinerja model pada *dataset* yang memiliki karakteristik kompleks.

Secara keseluruhan, temuan dari berbagai studi tersebut mengindikasikan bahwa *machine learning* memegang peranan yang signifikan dalam mengoptimalkan performa sistem klasifikasi diabetes. Rangkuman dari penelitian-penelitian terdahulu yang relevan dengan penelitian ini disajikan pada Tabel 2.1. Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti & Tahun	Masalah	Metode	Hasil
1	Ohanyan [2]	Penelitian ini membahas hubungan berbagai faktor *urban exposome* terhadap *diabetes melitus tipe 2*, karena faktor lingkungan seperti	Metode yang digunakan adalah *LASSO*, *Random Forest*, dan *Artificial Neural Network* untuk menganalisis 85 faktor paparan lingkungan terhadap *diabetes melitus*	Hasil penelitian menunjukkan bahwa nilai rata-rata rumah yang rendah, proporsi penduduk non-Barat yang lebih

		polusi udara, suhu, kebisingan, dan kondisi sosial lingkungan diduga berpengaruh terhadap risiko diabetes.	*tipe 2*. Data yang digunakan berasal dari 14.829 partisipan pada studi kohort AMIGO. Evaluasi model dilakukan dengan *nested cross-validation* dan metrik *logLoss*.	tinggi, dan suhu permukaan yang lebih tinggi berhubungan dengan peningkatan risiko *diabetes melitus tipe 2*. Dalam perbandingan model, *LASSO* memiliki performa terbaik dibandingkan *Random Forest* dan *Artificial Neural Network* berdasarkan nilai *logLoss*.
2	Maria Ali, [3]	Penelitian ini membahas klasifikasi penyakit diabetes karena model *classifier* tunggal sebelumnya masih memiliki akurasi yang rendah, sehingga dibutuhkan metode yang dapat meningkatkan hasil klasifikasi.	*KNN, Naive Bayes, LDA, Decision Tree, Random Forest.*	Pengujian yang dilakukan menunjukkan bahwa metode *stacking* mampu menghasilkan kinerja paling unggul, dengan nilai akurasi mencapai 97,35%, *recall* 100%, *precision* 92,17%, dan *F-measure* 96,21%.
3	N. Camerlingo, [4]	Penelitian ini membahas penentuan waktu pemberian *insulin correction bolus* pada penderita	*Decision Tree, Logistic Regression, LASSO.*	Hasil penelitian menunjukkan bahwa *Decision Tree* memberikan

		diabetes tipe 1, karena model simulasi sebelumnya belum mampu menggambarkan perilaku pasien secara realistis dalam pengambilan keputusan terapi.		performa lebih baik dibandingkan *Logistic Regression*, dengan nilai *accuracy* 0,790, *sensitivity* 0,501, *specificity* 0,861, *precision* 0,467, dan *AUROC* 0,774. Model ini juga dinilai lebih mudah diinterpretasikan.
4	Victor Chang, [5]	Penelitian ini membahas klasifikasi diabetes melitus menggunakan *dataset* Pima Indians untuk mendukung *e-diagnosis system*, karena deteksi dini diabetes penting dan model *machine learning* di bidang kesehatan perlu lebih mudah dipahami.	*Naive Bayes, Random Forest, J48 Decision Tree.*	Berdasarkan hasil pengujian pada keseluruhan *dataset*, algoritma *Random Forest* menunjukkan kinerja terbaik dengan nilai *akurasi* 79,57%, *precision* 89,40%, *specificity* 75,00%, *F-score* 85,17%, dan *AUC* 86,24%.
5	Roshi Saxena, [6]	Penelitian ini membahas prediksi dan klasifikasi *diabetes melitus* karena diperlukan metode yang lebih akurat untuk deteksi	*Multilayer Perceptron, Decision Tree, KNN, Random Forest.*	Hasil penelitian menunjukkan bahwa *Random Forest* memberikan performa terbaik dengan *accuracy*

		dini diabetes. Penelitian juga membandingkan beberapa metode seleksi fitur dan *classifier* untuk meningkatkan hasil prediksi.		79,83%, lebih tinggi dibanding *KNN* 78,58%, *Multilayer Perceptron* 77,60%, dan *Decision Tree* 76,07%.
6	Zaigham Mushtaq, [7]	Penelitian ini membahas prediksi *diabetes mellitus* karena deteksi dini diabetes penting, sementara dataset yang digunakan bersifat tidak seimbang dan mengandung *outlier*, sehingga dapat memengaruhi hasil klasifikasi.	*Logistic Regression, SVM, KNN, Gradient Boosting, Naive Bayes, Random Forest, Voting Classifier.*	Hasil penelitian menunjukkan bahwa *Random Forest* memperoleh akurasi terbaik di tahap pertama sebesar 80,7% setelah penerapan *SMOTE*. Selanjutnya, *Voting Classifier* memberikan hasil tertinggi secara keseluruhan dengan *accuracy* 81,7% pada *dataset default* dan 81,5% pada *dataset* seimbang.
7	Koushik Chandra Howlader, [8]	Penelitian ini membahas klasifikasi dan identifikasi atribut penting untuk mendeteksi diabetes tipe 2, karena	*GAMBoost, Logistic Regression, Naive Bayes, GBM.*	Hasil penelitian menunjukkan bahwa *Generalized Additive Model using LOESS*

		diperlukan model yang lebih akurat untuk membedakan pasien diabetes dan non-diabetes serta mengetahui faktor yang paling berpengaruh.		*(GAMLOESS)* menjadi model terbaik, sedangkan atribut yang paling sering muncul sebagai prediktor penting adalah *glucose, BMI, diabetes pedigree function,* dan *age.*
8	Chun-Yang Chou, [9]	Penelitian ini membahas prediksi diabetes karena diabetes merupakan penyakit kronis yang perlu dideteksi lebih dini agar komplikasi dapat dicegah dan kualitas hidup pasien dapat ditingkatkan.	*Logistic Regression, Neural Network, Decision Jungle, Boosted Decision Tree*	Hasil penelitian menunjukkan bahwa *Two-Class Boosted Decision Tree* menjadi model terbaik dengan *accuracy* 0,953, *precision* 0,927, *recall* 0,931, *F1-score* 0,929, dan *AUC* 0,991.
9	Gangani Dharmarathne, [10]	Penelitian ini membahas diagnosis diabetes karena model *machine learning* sebelumnya dinilai belum cukup transparan, sehingga diperlukan pendekatan yang tidak hanya akurat tetapi juga dapat menjelaskan alasan	*Decision Tree, KNN, SVC, XGBoost.*	Hasil penelitian menunjukkan bahwa *XGBoost* menjadi model terbaik dengan *accuracy testing* 77% dan *AUC testing* 0,82. Penelitian ini juga menghasilkan

		prediksi kepada pengguna.		antarmuka *self-explainable* berbasis *SHAP* untuk membantu menjelaskan prediksi diabetes.
10	Rakibul Islam, [11]	Penelitian ini membahas prediksi diabetes tipe 2 untuk mendukung *clinical decision support system (CDSS)*, karena deteksi dini diabetes penting untuk membantu dokter dan pasien dalam pengambilan keputusan diagnosis.	*KNN, Decision Tree, Random Forest, Naive Bayes, SVM, Histogram-based Gradient Boosting.*	Berdasarkan hasil evaluasi yang diperoleh, algoritma *Histogram-based Gradient Boosting (HGBT)* mengungguli metode lainnya dengan capaian *accuracy* 92,21%, *precision* 92,33%, *recall* 92,21%, dan *F1-score* 92,25%.

Analisis Penelitian Terdahulu

Berdasarkan Tabel 2.1 Penelitian Terdahulu, dapat diidentifikasi bahwa penelitian terkait klasifikasi diabetes menggunakan *machine learning* menunjukkan pola utama yang saling berkaitan.

1. Dominasi Penggunaan Algoritma *Ensemble* dan *Machine Learning* Klasik

   Algoritma yang paling banyak digunakan dalam penelitian-penelitian sebelumnya meliputi *Random Forest, Decision Tree, Logistic Regression,* dan *KNN*. Di antara keempatnya, *Random Forest* secara konsisten menunjukkan performa yang tinggi, didukung oleh kemampuannya dalam memproses data non-linear sekaligus menekan risiko *overfitting*. Sementara

itu, penerapan metode *ensemble* seperti *XGBoost* dan *voting classifier* terbukti mampu menghasilkan kinerja yang lebih baik dibandingkan penggunaan model tunggal.

2. Fokus pada Peningkatan Akurasi Model Klasifikasi

   Sebagian besar penelitian terdahulu berfokus pada peningkatan performa model dari sisi *accuracy, precision, recall,* dan *F1-score*. Beberapa penelitian juga menggunakan teknik *balancing data* seperti *SMOTE* untuk mengatasi *data imbalance* yang umum terjadi pada dataset kesehatan. Hal ini menunjukkan bahwa aspek evaluasi model menjadi faktor utama dalam penelitian klasifikasi diabetes.

3. Kurangnya Konsistensi dalam Perbandingan *Logistic Regression* dan *Random Forest*

   Meskipun *Logistic Regression* dan *Random Forest* sama-sama sering digunakan, hanya sedikit penelitian yang secara khusus membandingkan kedua algoritma ini dalam satu *framework* yang sama, terutama pada *dataset* yang menggabungkan indikator klinis dan gaya hidup secara bersamaan. Hal ini menunjukkan adanya ruang penelitian yang masih terbuka untuk analisis komparatif yang lebih terstruktur.

   Selain sering digunakan pada penelitian klasifikasi diabetes, Logistic Regression dan Random Forest dipilih karena memiliki karakteristik yang berbeda. Logistic Regression mewakili pendekatan klasifikasi linear yang sederhana dan mudah diinterpretasikan, sedangkan Random Forest merupakan algoritma ensemble yang mampu menangani hubungan non-linear antar variabel serta memberikan informasi mengenai tingkat kepentingan fitur (*feature importance*). Perbedaan karakteristik tersebut menjadikan kedua algoritma menarik untuk dibandingkan pada kasus klasifikasi diabetes tipe 2.

4. Variasi fitur dan pendekatan dataset juga menjadi aspek penting dalam penelitian klasifikasi diabetes.

   Namun, masih terdapat keterbatasan pada sebagian penelitian terdahulu yang hanya berfokus pada data klinis tanpa mempertimbangkan kombinasi dengan faktor gaya hidup, sehingga ruang eksplorasi model masih terbuka untuk pendekatan yang lebih komprehensif.

Perbedaan Penelitian dengan Penelitian Terdahulu

Berdasarkan analisis terhadap penelitian terdahulu, terdapat beberapa persamaan pendekatan yang umum digunakan, yaitu pemanfaatan algoritma *machine learning* untuk klasifikasi diabetes, penggunaan metrik evaluasi seperti *accuracy, precision, recall*, dan *F1-score*, serta fokus pada peningkatan performa model melalui berbagai teknik optimasi. Namun demikian, penelitian-penelitian tersebut masih memiliki beberapa keterbatasan yang menjadi dasar perbedaan dengan penelitian ini.

Letak kebaruan penelitian ini dibandingkan studi-studi sebelumnya terletak pada pendekatan komparasi yang lebih terfokus. Penelitian terdahulu pada umumnya hanya mengoptimalkan satu algoritma tertentu, atau membandingkan banyak algoritma sekaligus tanpa kajian mendalam terhadap dua model yang spesifik. Berbeda dengan hal tersebut, penelitian ini memusatkan perhatian pada perbandingan antara *Logistic Regression* dan *Random Forest*, yang masing-masing merepresentasikan pendekatan model linear sederhana dan model ensemble berbasis non-linear.

Selain itu, penelitian ini menggunakan dataset yang menggabungkan indikator klinis dan gaya hidup, sehingga memberikan cakupan variabel yang lebih luas dibandingkan beberapa penelitian sebelumnya yang hanya berfokus pada data klinis saja. Hal ini memungkinkan model untuk menangkap pola risiko diabetes yang lebih komprehensif dari berbagai aspek faktor kesehatan.

Perbedaan lainnya terletak pada tujuan analisis, di mana penelitian ini tidak hanya berfokus pada performa klasifikasi, tetapi juga bertujuan untuk

mengidentifikasi algoritma yang paling optimal dalam konteks dataset yang digunakan. Melalui pendekatan tersebut, penelitian ini diharapkan mampu menghasilkan rekomendasi yang lebih terarah dalam menentukan algoritma yang paling efektif untuk penanganan kasus klasifikasi diabetes.

Selain melakukan perbandingan performa algoritma, penelitian ini juga menerapkan proses *hyperparameter tuning* menggunakan *GridSearchCV* untuk memperoleh konfigurasi parameter terbaik pada masing-masing model. Selanjutnya, model terbaik yang diperoleh diimplementasikan ke dalam aplikasi web berbasis *Streamlit* sebagai bentuk *deployment* sehingga hasil penelitian tidak hanya berhenti pada tahap evaluasi model, tetapi juga dapat digunakan untuk melakukan prediksi risiko diabetes secara interaktif.

2.2 Teori yang berkaitan

2.2.1 Diabetes Mellitus

Diabetes mellitus merupakan penyakit metabolik kronis yang ditandai dengan peningkatan kadar *glukosa* dalam darah akibat gangguan pada sekresi *insulin*, kerja *insulin*, atau keduanya [12]. Kondisi ini terjadi ketika tubuh tidak mampu memproduksi *insulin* secara cukup atau tidak dapat menggunakannya secara efektif sehingga kadar gula darah berada di atas batas normal.

Diabetes mellitus merupakan salah satu penyakit tidak menular yang memiliki prevalensi tinggi di masyarakat dan dapat menimbulkan berbagai komplikasi serius apabila tidak ditangani dengan baik. Komplikasi tersebut meliputi penyakit kardiovaskular, kerusakan ginjal, gangguan penglihatan, hingga kerusakan saraf.

Secara umum, *diabetes mellitus* diklasifikasikan menjadi diabetes tipe 1, *diabetes tipe 2*, dan diabetes gestasional [13]. Diabetes tipe 1 terjadi akibat kegagalan pankreas dalam memproduksi *insulin*, sedangkan *diabetes tipe 2* disebabkan oleh resistensi *insulin* atau penurunan sensitivitas tubuh terhadap insulin. Diabetes gestasional terjadi pada masa kehamilan. Dalam penelitian ini, jenis diabetes yang dikaji secara spesifik adalah *Diabetes Mellitus Tipe 2*, yaitu

jenis diabetes yang paling umum terjadi di masyarakat dan erat kaitannya dengan faktor gaya hidup serta kondisi demografis individu.

2.2.2 Machine Learning

Machine learning merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan sistem komputer untuk mempelajari pola dari data dan melakukan prediksi atau pengambilan keputusan tanpa diprogram secara eksplisit [14].

Penerapan *machine learning* telah merambah berbagai sektor, mulai dari kesehatan, pendidikan, ekonomi, hingga transportasi. Khususnya di bidang kesehatan, teknologi ini dimanfaatkan untuk mendukung proses diagnosis penyakit, memperkirakan risiko kesehatan, serta mengolah data medis dengan tingkat efisiensi dan akurasi yang lebih tinggi.

Ditinjau dari pendekatannya, *machine learning* secara umum terbagi menjadi tiga kategori, yaitu *supervised learning, unsupervised learning*, dan *reinforcement learning*. Penelitian ini mengadopsi pendekatan *supervised learning*, mengingat *dataset* yang digunakan telah dilengkapi dengan label target berupa status diabetes, yakni diabetes dan non-diabetes.

2.2.3 Logistic Regression

Equation 1 Logistic Regression

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta n^x n)}}$$

Logistic Regression adalah salah satu algoritma dalam *supervised learning* yang dirancang untuk menangani permasalahan klasifikasi, khususnya klasifikasi biner [15]. Algoritma ini bekerja dengan cara mengestimasi probabilitas suatu data untuk masuk ke dalam kelas tertentu berdasarkan variabel-variabel *input* yang diberikan.

Berbeda dengan regresi linear yang menghasilkan nilai kontinu, *Logistic Regression* menghasilkan nilai probabilitas dengan rentang 0 hingga 1 menggunakan fungsi *sigmoid*. Probabilitas tersebut kemudian dibandingkan dengan nilai *threshold* tertentu untuk menentukan hasil klasifikasi akhir.

Pada persamaan di atas, $P(y = 1 \mid x)$menyatakan probabilitas data termasuk ke dalam kelas positif, sedangkan $\beta_0, \beta_1, \dots, \beta_n$merupakan parameter model yang dipelajari selama proses pelatihan. Variabel $x_1, x_2, \dots, x_n$merepresentasikan fitur atau atribut yang digunakan dalam proses klasifikasi.

Logistic Regression memiliki beberapa kelebihan, seperti proses implementasi yang sederhana, kebutuhan komputasi yang relatif rendah, serta hasil model yang mudah diinterpretasikan. Oleh karena itu, algoritma ini banyak digunakan dalam berbagai bidang, termasuk bidang kesehatan untuk prediksi penyakit. Namun, *Logistic Regression* memiliki keterbatasan dalam menangani hubungan data yang kompleks dan bersifat non-linear.

2.2.4 Random Forest

Equation 2 Random Forest

$$\hat{y} = mode\{T_1(x), T_2(x), \dots, T_n(x)\}$$

Random Forest adalah algoritma berbasis *ensemble learning* yang dirancang untuk meningkatkan kualitas klasifikasi dengan cara menggabungkan sejumlah *decision tree* secara bersamaan [16]. Cara kerja algoritma ini didasarkan pada pembangunan banyak pohon keputusan yang masing-masing dilatih menggunakan subset data dan fitur yang dipilih secara acak.

Setiap pohon keputusan dikonstruksi melalui teknik *bootstrap sampling*, yakni proses pengambilan sampel secara acak dari dataset pelatihan. Pada setiap pembentukan node, *Random Forest* turut menyeleksi subset fitur secara acak, sehingga setiap pohon yang terbentuk memiliki karakteristik dan pola yang saling berbeda.

Prediksi akhir dari model ditentukan melalui mekanisme *majority voting* atas seluruh pohon keputusan yang ada. Kelas yang paling banyak dipilih oleh pohon-pohon tersebut akan ditetapkan sebagai hasil klasifikasi akhir.

Dari sisi keunggulan, *Random Forest* mampu menangani dataset berukuran besar, menekan risiko *overfitting*, dan menghasilkan kinerja yang relatif stabil pada berbagai jenis data. Algoritma ini juga efektif dalam menangkap hubungan antar data yang kompleks dan non-linear. Di sisi lain, interpretabilitas model *Random Forest* terbilang lebih rendah dibandingkan algoritma yang lebih sederhana seperti *Logistic Regression*.

2.2.5 Hyperparameter Tuning

Hyperparameter tuning merupakan proses optimasi parameter model yang dilakukan sebelum proses pelatihan untuk memperoleh performa prediksi yang lebih baik. Berbeda dengan parameter model yang dipelajari secara otomatis selama proses training, hyperparameter ditentukan oleh pengguna dan memengaruhi cara model dibangun. Pada penelitian ini, proses *hyperparameter tuning* dilakukan menggunakan metode *GridSearchCV* yang mengombinasikan berbagai nilai *parameter* dan mengevaluasinya menggunakan *cross validation*. Pada algoritma Logistic Regression, parameter yang dioptimasi meliputi nilai C dan solver, sedangkan pada Random Forest parameter yang dioptimasi meliputi n_estimators, max_depth, min_samples_split, dan min_samples_leaf. Model dengan hasil evaluasi terbaik dipilih sebagai model final yang digunakan dalam penelitian.

Equation 3 Hyperparameter Tuning with Cross Validation

$$CV_{score} = \frac{1}{k}\sum_{i=1}^{k} Score_i$$

Keterangan:

- CV_{score}= rata-rata skor validasi silang

- k= jumlah fold
- $Score_i$= skor evaluasi pada fold ke-i

GridSearchCV mengevaluasi seluruh kombinasi hyperparameter yang ditentukan dan menghitung rata-rata performa model menggunakan metode *k-fold cross validation*. Kombinasi parameter yang menghasilkan nilai rata-rata evaluasi tertinggi dipilih sebagai konfigurasi terbaik model.

Equation 4 Grid Search Optimization

$$\theta * = arg\theta \in \Theta max Score(\theta)$$

Keterangan:

- Θ= himpunan seluruh kombinasi hyperparameter
- θ= satu kombinasi hyperparameter
- $Score(\theta)$= nilai evaluasi model
- θ^*= kombinasi hyperparameter terbaik

Grid Search bertujuan mencari kombinasi hyperparameter terbaik (θ^*)yang menghasilkan nilai evaluasi tertinggi dari seluruh kombinasi parameter yang diuji.

2.2.6 Pemilihan Algoritma dalam Penelitian Ini

Perkembangan *machine learning* dalam beberapa tahun terakhir menghadirkan berbagai algoritma dengan kemampuan prediksi yang semakin tinggi. Sebelum menentukan algoritma yang digunakan dalam penelitian ini, dilakukan kajian terhadap beberapa pendekatan yang umum digunakan untuk klasifikasi data klinis, meliputi *gradient boosting methods, support vector machine*, hingga pendekatan *deep learning.*

XGBoost, LightGBM, dan *CatBoost* merupakan kelompok algoritma berbasis *gradient boosting* yang secara konsisten menunjukkan performa tinggi dalam kompetisi *machine learning*, termasuk pada dataset tabular terstruktur. *XGBoost (Extreme Gradient Boosting)* bekerja dengan membangun pohon secara sekuensial di mana setiap pohon baru memperbaiki residual pohon

sebelumnya [17]. *LightGBM* menggunakan pendekatan *histogram-based splitting* yang lebih efisien secara komputasi, sementara *CatBoost* dirancang khusus untuk menangani variabel kategorikal tanpa memerlukan encoding manual [18]. Ketiga metode ini memiliki kelebihan dalam hal akurasi prediksi, namun memiliki keterbatasan dalam interpretabilitas. Proses tuning *hyperparameter* yang kompleks (*learning rate, max depth, n_estimators, subsample*, dll.) juga meningkatkan risiko *overfitting* apabila *dataset* yang digunakan terbatas ukurannya [19]. Dalam konteks klinis, model yang tidak dapat dijelaskan mekanisme pengambilan keputusannya akan sulit diterima oleh tenaga medis dan melewati standar regulasi kesehatan seperti yang disyaratkan dalam pedoman AI klinis [20].

Support Vector Machine (SVM) merupakan algoritma klasifikasi yang bekerja dengan menemukan *hyperplane* optimal yang memaksimalkan *margin* antara kelas. *SVM* menunjukkan performa baik pada dataset berdimensi tinggi dan relatif robust terhadap overfitting pada ruang fitur yang besar. Namun, *SVM* memiliki kompleksitas komputasi $O(n^2)$ hingga $O(n^3)$ terhadap jumlah sampel, yang menjadikannya kurang efisien untuk *dataset* berukuran besar [21]. Selain itu, *SVM* pada dasarnya adalah model *black box* yang tidak menyediakan probabilitas prediksi secara langsung tanpa *kalibrasi* tambahan, serta tidak mudah diinterpretasikan oleh klinisi dalam lingkungan klinis nyata.

Neural Network dan *Deep Learning* menawarkan kapasitas representasi yang jauh lebih tinggi karena kemampuannya menangkap pola *non-linear* yang kompleks. Pendekatan ini relevan untuk data bervolume besar dan berstruktur kompleks seperti citra medis atau data deret waktu. Namun untuk dataset *tabular* berukuran sedang seperti yang digunakan dalam penelitian ini (100.000 sampel, 15 fitur), *neural network* cenderung tidak memberikan keunggulan signifikan dibandingkan metode yang lebih sederhana [22]. Jumlah parameter yang besar meningkatkan risiko *overfitting*, kebutuhan *komputasi* jauh lebih tinggi, dan yang paling kritis dalam konteks kesehatan adalah sifatnya sebagai

black box yang tidak dapat memberikan penjelasan medis yang dapat dipertanggungjawabkan secara klinis.

Berdasarkan kajian komparatif tersebut, pemilihan *Logistic Regression* dan *Random Forest* dalam penelitian ini didasarkan pada beberapa pertimbangan ilmiah yang saling menguatkan. Pertama, keduanya termasuk dalam kategori *white-box* atau *grey-box* model yang memiliki tingkat *interpretabilitas* tinggi. *Logistic Regression* menghasilkan koefisien yang dapat langsung dikaitkan dengan kontribusi setiap fitur klinis terhadap probabilitas diagnosis, sehingga memudahkan klinisi dalam memahami dasar pengambilan keputusan model [23]. *Random Forest* menyediakan *feature importance score* yang secara eksplisit menunjukkan fitur mana yang paling dominan, yang dalam konteks penelitian ini terbukti konsisten dengan penanda klinis diabetes yang diakui secara medis seperti *HbA1c* dan *glukosa darah*.

Kedua, ukuran *dataset* dalam penelitian ini (100.000 sampel) berada pada rentang dimana kedua algoritma ini diketahui memiliki performa yang baik pada dataset tabular dan dapat dioptimalkan lebih lanjut melalui proses *hyperparameter tuning* untuk memperoleh konfigurasi parameter yang lebih sesuai dengan karakteristik data penelitian. Ketiga, dalam domain kesehatan, keandalan dan konsistensi prediksi lebih diutamakan dibandingkan peningkatan akurasi marginal dari *model* yang lebih kompleks, mengingat konsekuensi klinis dari prediksi yang salah. Keempat, kemudahan *deployment* dan *inferensi real-time* menjadi pertimbangan praktis yang tidak dapat diabaikan, terutama apabila model ini diintegrasikan ke dalam sistem pendukung keputusan klinis (*Clinical Decision Support System / CDSS*) yang memerlukan respons prediksi dalam hitungan milidetik [24].

Meskipun *XGBoost, LightGBM,* dan *CatBoost* sering dilaporkan menghasilkan performa tinggi pada data *tabular*, tujuan utama penelitian ini bukan untuk mencari model dengan akurasi tertinggi, melainkan mengevaluasi keseimbangan antara performa prediktif dan interpretabilitas pada konteks prediksi diabetes. Oleh karena itu, penelitian difokuskan pada *Logistic*

Regression sebagai model linear yang mudah dijelaskan dan *Random Forest* sebagai model *ensemble* yang mampu menangkap hubungan *non-linear* tanpa kehilangan kemampuan interpretasi global melalui *feature importance*.

2.2.7 Python

Python adalah bahasa pemrograman tingkat tinggi yang populer di kalangan praktisi data science, analisis data, dan *machine learning*, terutama karena sintaksnya yang ringkas dan relatif mudah dipelajari.

Ketersediaan berbagai library pendukung menjadi salah satu keunggulan utama Python dalam ekosistem pengolahan data dan *machine learning*. Di antaranya adalah NumPy untuk komputasi numerik, Pandas untuk manipulasi dan pengelolaan data, Matplotlib serta Seaborn untuk keperluan visualisasi, dan Scikit-learn sebagai platform implementasi algoritma *machine learning*.

Dalam konteks penelitian ini, Python dimanfaatkan pada seluruh tahapan utama, mulai dari preprocessing data, pembangunan model klasifikasi, hingga evaluasi performa algoritma yang digunakan.

2.3 Framework Penelitian

2.3.1 CRISP-DM (Cross Industry Process for Data Mining)

Penelitian ini menggunakan *framework CRISP-DM (Cross Industry Standard Process for Data Mining)* sebagai kerangka kerja utama dalam pengolahan data dan pembangunan model *machine learning*. *Framework* ini dipilih karena menawarkan alur kerja yang sistematis dan fleksibel, serta telah terbukti banyak diterapkan baik dalam lingkungan penelitian maupun industri *data mining*.

Di samping *CRISP-DM*, terdapat framework lain yang juga umum digunakan dalam penelitian sejenis, seperti *KDD (Knowledge Discovery in Databases)* dan *SEMMA (Sample, Explore, Modify, Model, Assess)*. Meskipun demikian, *CRISP-DM* dinilai paling sesuai dengan kebutuhan penelitian ini karena kemampuannya dalam mengakomodasi pemahaman konteks permasalahan

sekaligus mendukung proses analisis data secara iteratif, yang menjadi kebutuhan utama dalam klasifikasi *Diabetes Mellitus Tipe 2*.

Berikut disajikan Tabel 2.3 Perbandingan *Framework*, perbandingan beberapa *framework data mining* yang menjadi pertimbangan dalam penelitian ini:

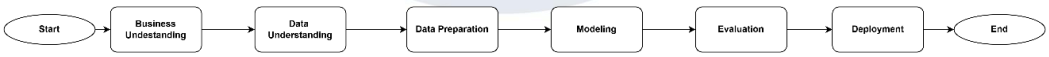
Tabel 2. 2 Perbandingan Framework

Framework	**Karakteristik**	**Kelebihan**	**Kekurangan**	**Alasan Digunakan / Tidak Digunakan**
CRISP-DM	*Framework data mining* berbasis tahapan iteratif	Fleksibel, sistematis, memiliki tahap *Business Understanding*, banyak digunakan dalam industri dan penelitian	Tidak menyediakan implementasi teknis secara spesifik	Digunakan karena sesuai untuk penelitian *machine learning* dan mudah diterapkan secara bertahap
KDD	*Framework* ekstraksi pengetahuan dari data	Fokus pada proses penemuan pola dan *knowledge discovery*	Bersifat lebih teknis dan linier	Tidak dipilih karena kurang fleksibel untuk proses iteratif penelitian
SEMMA	*Framework data mining* yang dikembangkan oleh *SAS*	Fokus pada pemodelan dan analisis data	Lebih bergantung pada tools *SAS* dan kurang menekankan aspek bisnis	Tidak dipilih karena penelitian menggunakan *Python* dan membutuhkan pemahaman konteks bisnis

Dibandingkan framework lain seperti *KDD, CRISP-DM* dipilih karena sifatnya yang lebih fleksibel dan iteratif, memungkinkan peneliti untuk kembali ke tahapan sebelumnya apabila ditemukan kendala dalam proses analisis. Keunggulan lain *CRISP-DM* terletak pada adanya tahapan *Business Understanding*, yang secara eksplisit menempatkan pemahaman konteks permasalahan dari perspektif domain sebagai bagian integral dari alur kerja, tidak sekadar dari sudut pandang teknis. Aspek ini sangat relevan dalam penelitian di bidang kesehatan, di mana pemahaman terhadap konteks klinis menjadi prasyarat agar hasil analisis dapat diimplementasikan secara praktis.

Di sisi lain, *CRISP-DM* juga merupakan *framework* yang paling dominan digunakan dalam praktik *data mining* dan *machine learning* di tingkat global, sehingga mendukung kemudahan dokumentasi, reproduktibilitas penelitian, serta komparasi hasil dengan studi-studi sebelumnya.

Alur tahapan *framework CRISP-DM* yang diterapkan dalam penelitian ini diilustrasikan pada Gambar 2.1 *Flowchart Diagram CRISP-DM*.



Gambar 2. 1 *Flowchart Diagram CRISP-DM* (sumber draw.io) [25]

CRISP-DM terdiri atas enam tahapan utama yang saling berkesinambungan, yaitu *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment*. Alur keseluruhan tahapan tersebut diilustrasikan pada Gambar 2.1 *Flowchart Diagram CRISP-DM*.

1. *Business Understanding* (Pemahaman Bisnis) Tahap ini difokuskan pada perumusan tujuan penelitian, yakni mengklasifikasikan risiko diabetes berdasarkan data kesehatan pasien melalui pendekatan *machine learning*. Pada tahap ini pula ditetapkan tujuan teknis berupa pembangunan model klasifikasi yang mampu menghasilkan tingkat akurasi yang optimal.

2. *Data Understanding* (Pemahaman Data) Tahap ini melibatkan kegiatan pengumpulan dan eksplorasi awal data guna memperoleh pemahaman menyeluruh mengenai struktur *dataset*, distribusi variabel, serta kualitas data secara keseluruhan. Proses ini mencakup identifikasi *missing value*, *outlier*, dan karakteristik tiap variabel yang akan digunakan dalam proses klasifikasi.

3. *Data Preparation* (Persiapan Data) Pada tahap ini dilakukan serangkaian proses persiapan data, meliputi pembersihan data (*data cleaning*), transformasi, dan seleksi fitur yang relevan. Data juga ditransformasikan melalui proses encoding pada variabel kategorikal serta normalisasi menggunakan StandardScaler agar sesuai dengan kebutuhan algoritma *machine learning* yang digunakan.

4. *Modeling* (Pemodelan) Tahap ini mencakup proses pembangunan model klasifikasi menggunakan dua algoritma *machine learning*, yaitu Logistic Regression dan Random Forest. Sebelum dilakukan evaluasi, masing-masing algoritma dioptimalkan menggunakan *hyperparameter tuning* dengan metode *GridSearchCV* untuk memperoleh konfigurasi parameter terbaik. Model yang telah dioptimasi kemudian dilatih untuk mengklasifikasikan status diabetes berdasarkan variabel independen yang tersedia dalam dataset.

5. *Evaluation (Evaluasi)* Performa kedua model diukur menggunakan sejumlah metrik evaluasi, meliputi accuracy, *precision*, *recall*, *F1-score*, dan confusion matrix. Selain itu, dilakukan *cross validation* untuk menguji konsistensi performa model serta analisis *feature importance* pada model terbaik untuk mengidentifikasi faktor-faktor yang paling berpengaruh terhadap prediksi diabetes. Hasil evaluasi dari masing-masing model selanjutnya dibandingkan untuk menentukan algoritma dengan kinerja terbaik.

6. *Deployment* (Penerapan) Tahap akhir ini dilakukan dengan mengimplementasikan model terbaik ke dalam aplikasi web berbasis *Streamlit*. Model yang telah dilatih dan dievaluasi disimpan menggunakan Joblib, kemudian diintegrasikan ke dalam aplikasi sehingga pengguna dapat

melakukan prediksi risiko diabetes secara interaktif berdasarkan data kesehatan yang dimasukkan.

2.4 *Tools/software* yang digunakan

Proses pengolahan data, analisis, dan pembangunan model klasifikasi *Diabetes Mellitus Tipe 2* dalam penelitian ini didukung oleh sejumlah perangkat lunak dan tools, yaitu:

1. *Python*

Dalam penelitian ini, *Python* dipilih sebagai bahasa pemrograman yang digunakan di seluruh tahapan penelitian, mencakup *preprocessing* data hingga evaluasi model *machine learning*. Keputusan ini didasari oleh dukungan ekosistem *Python* yang kaya, khususnya melalui library seperti *Pandas, NumPy, Matplotlib, Seaborn,* dan *Scikit-learn*, yang secara kolektif memfasilitasi analisis data dan pengembangan model secara efektif. Sintaks *Python* yang relatif sederhana turut menjadi pertimbangan, mengingat hal tersebut dapat memperlancar proses pengembangan dan analisis. Kedua algoritma yang diterapkan dalam penelitian ini, yakni *Logistic Regression* dan *Random Forest*, tersedia secara langsung dalam *library Scikit-learn*, sehingga implementasinya dapat dilakukan secara efisien dan konsisten.

Berikut perbandingan *Python* dengan *software* lain yang umum digunakan dalam pengembangan *machine learning* pada Tabel 2.4 Perbandingan *Tools*:

Tabel 2. 3 Perbandingan Tools

Tools/Software	**Kelebihan**	**Kekurangan**	**Alasan Pemilihan**
Python	Memiliki banyak library *machine learning* dan data science, sintaks sederhana, komunitas besar	Membutuhkan pemahaman coding	Dipilih karena fleksibel dan mendukung seluruh proses penelitian *machine learning*
Visual Studio Code	Editor kode ringan, mendukung banyak	Tidak menyediakan library analisis data secara langsung	Digunakan sebagai code editor untuk

	ekstensi dan bahasa pemrograman		menjalankan program Python
Microsoft Excel	Mudah digunakan untuk pengolahan data sederhana	Kurang optimal untuk *machine learning* dan dataset besar	Tidak dipilih sebagai tools utama karena keterbatasan analisis *machine learning*
Tools/Software	**Kelebihan**	**Kekurangan**	**Alasan Pemilihan**
Streamlit	Memudahkan pembuatan aplikasi web berbasis Python	Kustomisasi UI terbatas dibanding framework web penuh	Digunakan untuk *deployment* model prediksi diabetes berbasis web

2. *Jupyter Notebook*

Jupyter Notebook dimanfaatkan sebagai lingkungan pengembangan utama dalam menjalankan kode Python secara interaktif. Platform ini dipilih karena kemampuannya mengintegrasikan penulisan kode, narasi penjelasan, dan visualisasi hasil analisis dalam satu dokumen yang terpadu. Karakteristik tersebut sangat sesuai dengan pendekatan penelitian yang bersifat eksploratif, di mana setiap tahapan analisis dapat didokumentasikan secara langsung beserta keluaran yang dihasilkan. Di samping itu, mekanisme eksekusi berbasis sel (*cell-based execution*) pada *Jupyter Notebook* memberikan kemudahan dalam proses *debugging* serta iterasi analisis data secara bertahap.

3. *Library Python*

Beberapa library Python dimanfaatkan dalam penelitian ini sesuai dengan fungsinya masing-masing. Pandas digunakan untuk pengolahan dan manipulasi data, sedangkan NumPy digunakan untuk komputasi numerik. Implementasi model *machine learning* proses *hyperparameter tuning*, dan evaluasi model dilakukan menggunakan Scikit-learn. Visualisasi data dikerjakan dengan bantuan Matplotlib dan Seaborn. Selain itu, Joblib digunakan untuk menyimpan model hasil pelatihan, sedangkan *Streamlit* dimanfaatkan untuk mengimplementasikan model ke dalam aplikasi web berbasis Python.

.

4. ***Microsoft Excel***

   Microsoft Excel digunakan untuk eksplorasi awal dataset sebelum diproses lebih lanjut menggunakan *Python*.