BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Teori dasar memberikan pemahaman tentang konsep-konsep mendasar dan hubungan antar variabel, serta membantu menjelaskan fenomena tertentu. Pemahaman ini memungkinkan aplikasi yang lebih efektif dan analisis yang lebih mendalam.

2.1.1. Artificial Intelligence

Kecerdasan buatan (*Artificial Intelligence*) adalah bidang ilmu komputer yang berfokus pada pembuatan sistem yang dapat melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia (Oktavianus et al., 2023). Teknologi ini dapat mengambil keputusan dengan menganalisis dan memanfaatkan data yang ada di dalam sistem. Proses yang terjadi dalam kecerdasan buatan meliputi pembelajaran (*learning*), penalaran (*reasoning*), dan perbaikan diri (*self-correction*). Proses ini serupa dengan cara manusia menganalisis informasi sebelum membuat keputusan (Sobron & Lubis, 2021).

Sebagai cabang ilmu pengetahuan, kecerdasan buatan berfokus pada penggunaan mesin untuk menyelesaikan masalah kompleks dengan cara yang menyerupai kerja manusia. Dalam ranah ilmu komputer, AI berperan dalam pengembangan mesin (komputer) yang bisa melakukan tugas-tugas yang biasanya dilakukan oleh manusia, dan dalam beberapa kasus, bahkan bisa melampaui kemampuan manusia itu sendiri (Ramadhan et al., 2020).

Artificial Intelligence memiliki kemampuan unggul dalam menganalisis dan memproses data dalam skala besar secara efisien. Dalam konteks prediksi, AI memanfaatkan algoritma *machine learning* dan *deep learning* yang mampu mengenali pola, membuat generalisasi dari data pelatihan, serta memberikan hasil prediksi yang akurat. Beberapa kelebihan utama AI dalam melakukan prediksi data antara lain:

1. Mengolah dan menganalisis data yang besar dan kompleks

   AI dapat menganalisis data yang besar dan kompleks, meliputi fitur kategorikal dan numerik secara bersamaan. Dalam studi prediksi harga mobil bekas berdasarkan tahun perakitan, AI mampu menganalisis faktor-faktor seperti kondisi kendaraan, tren pasar, dan data historis secara bersamaan. Model seperti *Random Forest* dan *Deep Learning* mampu mengenali pola *non-linier* serta hubungan antar variabel yang sulit dideteksi oleh metode konvensional. AI juga dapat belajar secara otomatis dari data *historis* dan beradaptasi terhadap perubahan kondisi sehingga menghasilkan prediksi yang lebih akurat dan konsisten (Wilianto et al., 2024).

2. Analisis cepat dan akurat pada data medis

   Dalam bidang medis, AI berperan penting dalam prediksi penyakit jantung dengan menganalisis data pasien yang kompleks seperti tekanan darah, kadar kolesterol, detak jantung, dan riwayat medis. Dengan menggunakan algoritma seperti *Neural Network*, SVM, dan KNN, AI mampu mengenali pola-pola tersembunyi dari data historis yang sering kali sulit diidentifikasi secara manual. Hasil analisis ini dapat mendukung proses deteksi dini risiko penyakit

dan membantu tenaga medis dalam pengambilan keputusan klinis secara cepat, tepat, dan berbasis data (Arfian et al., 2025).

3. Pengolahan pada data tidak pasti dan berubah-ubah

   AI juga efektif dalam mengolah data yang bersifat dinamis dan tidak pasti. Salah satu penerapannya adalah penggunaan metode *Fuzzy Time Series* dalam memprediksi jumlah pengunjung Semarang Zoo. Metode seperti *Fuzzy Time Series* efektif dalam mengenali pola dari data yang tidak konsisten atau bersifat fluktuatif. Hal ini membuat AI sangat berguna dalam situasi yang membutuhkan adaptasi terhadap data yang berubah, seperti dalam perencanaan operasional atau prediksi jumlah pengunjung (Marzuqi et al., 2022).

2.1.2. Pembelajaran Mesin (*Machine Learning*)

Machine learning (ML) adalah teknologi pembelajaran mesin yang sangat bermanfaat dalam mempermudah pekerjaan dan menyelesaikan berbagai masalah. *Machine learning* bermula saat manusia berpikir tentang bagaimana komputer dapat mengingat apa yang baru saja dilakukan atau belajar dari pengalaman. Hal ini terbukti pada tahun 1952 ketika Arthur Samuel membuat program *game chekers* pada komputer IBM yang dapat mempelajari gerakan untuk memenangkan permainan *chekers* dan menyimpan gerakan tersebut dalam memorinya. *Machine learning* dirancang untuk membantu manusia menyelesaikan masalah dan membuatnya mudah digunakan. Ini juga membuatnya tidak perlu diunduh lagi (Telaumbanua et al., 2020).

Machine learning ialah studi tentang algoritma untuk mengajarkan sistem untuk melakukan tugas tertentu secara otomatis, seperti yang biasa dilakukan manusia. Dalam hal ini, istilah "belajar" mengacu pada kemampuan sistem untuk melakukan aktivitas yang telah dipelajari sebelumnya atau menemukan pola-pola yang sudah diamati untuk memahami informasi baru. *Machine Learning* memengaruhi banyak disiplin ilmu lainnya, seperti matematika, statistika, dan teori komputer, sebagai salah satu cabang dari AI. Menciptakan algoritma yang mampu menghasilkan sistem pembelajaran mandiri (*autonomous learning system*) dengan campur tangan manusia yang minimal adalah tujuan utama pembelajaran mesin (Fathurohman, 2021).

Berikut beberapa jenis utama dalam *Machine Learning* yang paling umum digunakan, beserta algoritma yang biasanya diterapkan pada masing-masing jenis:

1. *Supervised Learning* (Pembelajaran Terbimbing)

   Menggunakan data berlabel untuk mempelajari hubungan antara *input* dan *output*. Digunakan untuk tugas klasifikasi dan regresi. Contoh algoritmanya adalah *Linear Regression*, *Support Vector Machine* (SVM), *Random Forest*, dan algoritma klasifikasi biner (*Binary Classification*) seperti *Logistic Regression* dan *Naive Bayes*.
2. *Unsupervised Learning* (Pembelajaran Tak Terbimbing)

   Mengolah data tanpa label untuk menemukan pola atau struktur tersembunyi. Cocok untuk *clustering* dan reduksi dimensi. Contoh algoritmanya yaitu *K-Means*, *Hierarchical Clustering*, dan *Principal Component Analysis* (PCA).

3. *Reinforcement Learning* (Pembelajaran Penguatan)

   Melibatkan agen yang belajar dari interaksi dengan lingkungan melalui *reward* dan *penalty*. Digunakan untuk pengambilan keputusan berkelanjutan. Contoh algoritmanya seperti *Q-Learning* dan *Deep Q-Network* (DQN).

Penerapan ML terdiri atas beberapa tahapan penting yang harus dilakukan secara sistematis agar model yang dihasilkan memiliki akurasi dan performa yang optimal (Maulani et al., 2025). Berikut langkah-langkah yang diuraikan:

1. Pengumpulan data: mengambil data dari berbagai sumber (internal/eksternal) secara terstruktur untuk mendukung pelatihan model.
2. Praproses data: melibatkan pembersihan data, transformasi fitur, dan normalisasi agar data siap diproses secara optimal.
3. Pemisahan data dan pembuatan dataset: data dibagi menjadi *training* dan *test set* untuk menghindari *overfitting* dan mengevaluasi model secara objektif.
4. Pemilihan algoritma: menyesuaikan algoritma dengan jenis masalah, seperti regresi untuk nilai kontinu dan klasifikasi untuk label diskrit.
5. Pelatihan dan evaluasi model: model dilatih dengan data dan dievaluasi menggunakan metrik seperti akurasi, *precision*, *recall*, atau MSE.
6. Pemantauan dan pemeliharaan model: model dipantau dan diperbarui secara berkala agar tetap relevan terhadap perubahan pola data.

2.1.3. Binary Classification

Binary Classification ialah proses mengklasifikasikan sesuatu ke dalam salah satu dari dua kategori yang telah ditetapkan (Sebastian, 2019). Dalam pendekatan ini, setiap data yang dianalisis hanya dapat dikategorikan ke dalam

salah satu dari dua kelompok yang saling eksklusif. Metode ini sering digunakan dalam berbagai bidang, seperti deteksi spam, diagnosis medis, dan prediksi status kesehatan, di mana hasil akhirnya hanya terdiri dari dua kemungkinan, misalnya "positif" atau "negatif", "ya" atau "tidak", "sehat" atau "sakit".

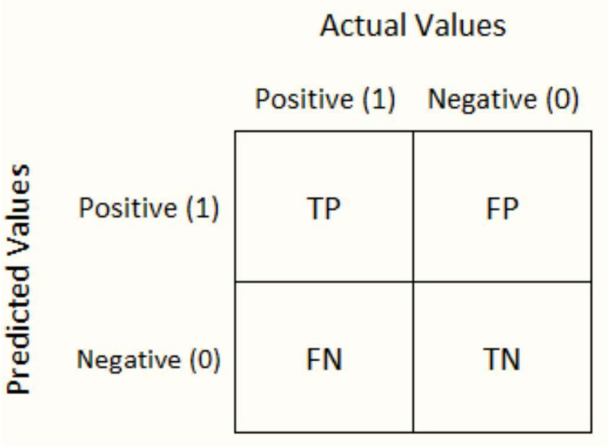
Mengelompokkan setiap *instance* (titik data) ke dalam salah satu dari dua kategori yang telah ditentukan sebelumnya adalah tujuan utama dari *binary classification* (Fitriani, 2024). Untuk mencapai tujuan ini, algoritma dilatih menggunakan kumpulan data yang diberi label. Ini memungkinkan model untuk mempelajari pola atau karakteristik dari setiap kelas. Setelah proses pelatihan selesai, diharapkan model dapat menggunakan pengetahuan yang diperoleh dari pelatihan ini untuk mengklasifikasikan data baru yang belum pernah dilihat sebelumnya.

Agar model *binary classification* dapat bekerja dengan baik, proses sistematis yang dimulai dengan pengumpulan data dan diakhiri dengan evaluasi kinerja. Proses kerja dari *binary classification* terdiri dari beberapa tahapan sebagai berikut:

1. Pengumpulan Data: Menghimpun data yang memuat fitur (*input*) dan label (*output*). Contoh: pada sistem deteksi spam, fitur berupa teks *email* dan labelnya adalah "spam" atau "bukan spam".
2. Pra-pemrosesan Data: Membersihkan data dari nilai kosong atau duplikat, serta mengonversi data kategorikal ke bentuk numerik agar dapat diproses oleh algoritma.

3. Seleksi fitur: menentukan fitur paling relevan melalui normalisasi, penskalaan, atau eliminasi variabel yang kurang penting untuk meningkatkan akurasi model.
4. Pemilihan model: menentukan algoritma yang sesuai, seperti *Gradient Boosting Classifier*, *Logistic Regression*, *Decision Tree*, *Random Forest*, *Support Vector Machine* (SVM), dan *Neural Network*.
5. Pelatihan model: melatih model menggunakan data berlabel untuk mengenali pola klasifikasi antar dua kelas.
6. Model evaluasi: menilai performa model dengan metrik seperti akurasi, presisi, *recall*, dan *F1-score*.
7. Prediksi data baru: model yang telah terlatih digunakan untuk memprediksi label pada data yang belum dikenal.

Untuk mengevaluasi model klasifikasi, digunakan beberapa metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Metrik-metrik ini dihitung berdasarkan hasil klasifikasi yang disusun dalam *confusion matrix*.



Gambar 2. 1 *Confusion Matrix*
Sumber: (Akrom, 2024)

Keterangan:

1. *True Positive* (TP): Kasus positif yang diprediksi positif oleh model.
2. *True Negative* (TN): Kasus negatif yang diprediksi negatif oleh model.
3. *False Positive* (FP): Kasus negatif yang salah diprediksi sebagai positif.
4. *False Negative* (FN): Kasus positif yang salah diprediksi sebagai negatif.

Penjelasan dan Rumus Metrik Evaluasi:

a. *Accuracy* mengukur rasio prediksi yang benar terhadap jumlah keseluruhan prediksi.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100$$ **Rumus 2. 1** *Accuracy*

b. *Precision* menunjukkan seberapa besar proporsi prediksi positif yang benar dari seluruh prediksi positif yang dihasilkan.

$$Precision = \frac{TP}{(TP + FP)}$$ **Rumus 2. 2** *Precision*

c. *Recall* mengukur sejauh mana model mampu mengidentifikasi seluruh kasus positif yang sebenarnya.

$$Recall = \frac{TP}{(TP + FN)}$$ **Rumus 2. 3** *Recall*

d. *F1-score* adalah rata-rata harmonis antara *precision* dan *recall*, yang digunakan saat diperlukan keseimbangan antara keduanya, terutama ketika data tidak seimbang

$$F1 - score = 2 \times \frac{Precision.Recall}{Precision + Recall}$$ **Rumus 2. 4** *F1-score*

Selain metrik-metrik tersebut, digunakan juga kurva ROC (*Receiver Operating Characteristic*) untuk menganalisis kinerja model pada berbagai nilai

ambang klasifikasi. Seberapa baik model dapat membedakan dua kelas ditunjukkan oleh nilai Area Under the Curve (AUC) dari kurva ROC. Semakin mendekati 1, maka semakin baik performa klasifikasi model (Akrom, 2024).

2.1.4. Gradient Boosting Classifier

Salah satu algoritma yang umum digunakan dalam *binary classification* adalah *Gradient Boosting Classifier*. Algoritma ini menggunakan pendekatan *ensemble learning* (pembelajaran kelompok), yang berarti bahwa beberapa model yang lemah (*weak learners*) secara bertahap digabungkan untuk membentuk model yang kuat (*strong learner*).

Proses kerja *Gradient Boosting* dilakukan secara berurutan, dengan dilakukan melalui beberapa tahapan sebagai berikut:

1. Inisialisasi Prediksi Awal

   Tahap pertama adalah menentukan prediksi awal untuk setiap data. Dalam kasus klasifikasi, nilai awal ini diperoleh dari log-odds dari variabel target, yang kemudian diubah menjadi probabilitas menggunakan fungsi logistik:

   $$P(y = 1) = \frac{e^{\log(odds)}}{1+e^{\log(odds)}}$$ **Rumus 2. 5** Fungsi Logistik

2. Menghitung Nilai Residual

   Setelah prediksi awal ditentukan, langkah selanjutnya adalah menghitung residual untuk masing-masing data. Residual merupakan selisih antara nilai aktual dan nilai prediksi:

   $$Residual_i = y_i - \hat{y}_i$$ **Rumus 2. 6** Residual

Di mana y_i adalah nilai aktual (0 untuk *No*, 1 untuk *Yes*) dan $\hat{y}_i$ adalah nilai prediksi sebelumnya.

3. Memprediksi Nilai Residual

   Pada tahap ini, algoritma membentuk sebuah decision tree berdasarkan nilai residual. Model ini berfungsi untuk mempelajari pola dari residual. Selanjutnya, nilai prediksi residual akan ditransformasi menggunakan perhitungan koefisien bobot:

   $$\gamma = \frac{\sum Residual_i}{\sum[Prev\ Probability_i \times (1 - Prev\ Probability_i)]}$$ **Rumus 2. 7** Koefisien Koreksi

   Nilai γ digunakan sebagai skala koreksi terhadap prediksi model sebelumnya.

4. Pembaruan Prediksi dengan Probabilitas Baru

   Hasil prediksi dari decision tree yang berisi residual digunakan untuk memperbarui prediksi sebelumnya. Proses ini menggunakan *learning rate* untuk mengontrol besarnya pembaruan:

   $$\hat{y}_{baru} = \hat{y}_{lama} + v \times Predicted\ Residual$$ **Rumus 2. 8** Pembaruan Prediksi

   Dimana:

   a. $\hat{y}_{baru}$: prediksi setelah diperbarui

   b. $\hat{y}_{lama}$: prediksi sebelumnya

   c. v: *learning rate*, biasanya antara 0.01–0.1

   d. Predicted Residual: hasil prediksi dari model lemah terhadap residual

5. Iterasi dan Pembaruan Residual

   Setelah prediksi baru dihitung, residual kembali dihitung dengan mengurangkan nilai aktual dengan prediksi baru. Langkah ke-3 hingga ke-5

akan terus diulang hingga nilai residual mendekati nol, atau iterasi mencapai jumlah maksimum sesuai pengaturan *hyperparameter*.

6. Komputasi Akhir

   Model akhir akan diperoleh setelah seluruh iterasi selesai, yang merupakan akumulasi dari seluruh koreksi residual terhadap prediksi awal:

   $F(x) = Initial\ Prediction + v \cdot$ **Rumus 2. 9** Prediksi Akhir

   $Predicted\ Residual\ 1\ + v \cdot$

   $Predicted\ Residual\ 2\ + \cdots$

   Dimana:

   a. F(x) = prediksi akhir dari model
   b. ν = learning rate
   c. Predicted Residual = output dari model yang mempelajari residual

Gradient Boosting memiliki beberapa keunggulan, seperti dapat digunakan untuk berbagai jenis data, efektif menangani data dengan outlier, dan memberikan hasil prediksi yang akurat pada data dengan pola yang kompleks. Selain itu, algoritma ini fleksibel karena dapat disesuaikan dengan berbagai fungsi kerugian sesuai dengan kebutuhan kasus klasifikasi (Sri Diantika et al., 2023).

2.1.5. Diabetes

Diabetes melitus adalah penyakit kronis yang bersifat jangka panjang dan dapat dialami oleh penderita sepanjang hidupnya (Sihotang, 2017). Diabetes merupakan gangguan metabolik yang terjadi akibat meningkatnya kadar glokosa dalam tubuh. Karena berfungsi sebagai sumber energi utama bagi tubuh, glukosa darah sangat penting untuk menjaga kesehatan tubuh. Penyakit ini dapat

menyebabkan berbagai komplikasi serius, seperti gangguan jantung, *stroke*, obesitas, serta kerusakan pada mata, ginjal, dan sistem saraf jika tidak ditangani dengan baik (Argina, 2020).

Berdasarkan klasifikasinya, diabetes melitus terbagi menjadi beberapa tipe, yaitu:

a. Diabetes melitus tipe 1 merupakan penyakit autoimun yang ditandai dengan ketidakmampuan tubuh memproduksi insulin akibat kerusakan sel-sel di pankreas. Penyakit ini paling sering terjadi pada remaja, terutama saat pubertas, meskipun dapat menyerang siapa saja di berbagai usia. Jumlah kasus DM tipe 1 sebanding antara laki-laki dan perempuan pada masa kanak-kanak, tetapi laki-laki cenderung lebih sering mengalaminya pada awal masa dewasa. Awalnya, penyakit ini paling banyak ditemukan di wilayah Eropa, namun kini kasusnya juga semakin sering dijumpai pada kelompok etnis lainnya di berbagai belahan dunia (Fauziani et al., 2024).

b. Diabetes melitus tipe 2 merupakan gangguan metabolik yang terjadi akibat adanya resistensi terhadap insulin serta gangguan fungsi sel beta pada pankreas yang berperan dalam produksi insulin. Gaya hidup menjadi faktor utama pemicu munculnya, terutama kebiasaan makan yang tidak sehat dan kurangnya aktivitas fisik. Peningkatan jumlah penderita diabetes tipe 2 secara signifikan juga dipengaruhi oleh berbagai perubahan dalam pola hidup masyarakat modern, termasuk rendahnya kesadaran untuk melakukan deteksi dini terhadap penyakit ini, minimnya aktivitas fisik harian, pola makan yang

tidak teratur dan tidak seimbang (Murtiningsih et al., 2021). Kombinasi faktor-faktor tersebut mempercepat munculnya dan penyebaran diabetes tipe 2 di tengah masyarakat.

c. Diabetes melitus tipe lain beberapa jenis diabetes disebabkan oleh satu kelainan genetik yang mempengaruhi bagaimana sel beta pankreas bekerja. Diabetes ini biasanya muncul di usia muda, yaitu sebelum berusia 25 tahun, dan disebut sebagai *maturity-onset diabetes of the young* atau MODY. MODY menunjukkan bahwa tubuh tidak dapat memproduksi insulin lagi, meskipun sensitivitas terhadap insulin tetap normal atau hanya terpengaruh secara kecil-kecilan (American Diabetes Association, 2014).
d. Diabetes melitus gestasional, merupakan kondisi gangguan toleransi glukosa yang terdeteksi pertama kali saat seorang wanita sedang hamil, di mana sebelumnya ia belum pernah didiagnosis menderita diabetes. Kondisi ini ditandai dengan meningkatnya kadar glukosa darah selama masa kehamilan (Wahyuni et al., 2021).

Untuk membantu proses identifikasi jenis diabetes berdasarkan karakteristik data pasien, penelitian ini menggunakan dataset dari Kaggle yang berisi 8 parameter yaitu *Pregnancies, Glucose, BloodPressure, SkinThickness,* Insulin*,* BMI*, DiabetesPedigreeFunction*, dan *Age*. Nilai-nilai untuk masing-masing jenis diabetes berbeda. Oleh karena itu, sangat penting untuk memahami berbagai nilai dari setiap parameter yang dapat menunjukkan apakah seseorang termasuk dalam kondisi normal, diabetes tipe 1, tipe 2, MODY, maupun diabetes gestasional.

Tabel 2. 1 Identifikasi Tipe Diabetes Berdasarkan Nilai Parameter

Parameter	Normal	Diabetes Tipe 1	Diabetes Tipe 2	MODY	Gestasional
Pregnancies (Jumlah kehamilan)	0 – 2	Variatif, sering rendah	0 – 5	0-1	≥ 1 (wanita hamil)
Glucose (mg/dL)	< 140	≥ 126 (puasa) / ≥ 200 (2 jam)	≥ 140 (puasa) / ≥ 200 (2 jam)	≥ 126 – 200	≥ 140 (2 jam)
Blood Pressure (mm Hg)	60 – 80 (normal)	Normal atau sedikit rendah	≥ 80	Normal	Normal atau sedikit naik
Skin Thickness (mm)	10 – 20	< 10	≥ 20	Normal	Normal
Insulin (mu U/ml)	15 – 100	Sangat rendah (< 10)	Normal sampai tinggi (> 100)	Normal atau rendah	Normal
BMI (kg/m^2)	18.5 – 24.9	Biasanya < 25	≥ 25 (*overweight*/obesitas)	18.5 – 24.9	Variatif
Diabetes Pedigree Function	< 0.3	Variatif, bisa rendah	≥ 0.3	≥ 0.3	Variatif
Age (tahun)	20 – 40	< 25	> 40	< 25	20 – 40 (wanita hamil)

Sumber: (ElSayed et al., 2024), (American Diabetes Association, 2020)

2.1.6. AutoAI

AutoAI adalah fitur canggih yang disediakan oleh IBM dalam *platform* Watson Studio yaitu IBM Cloud Pak for Data untuk membangun dan mengimplementasikan model *machine learning* secara otomatis tanpa memerlukan kemampuan pemrograman. Alat ini dirancang untuk menyederhanakan proses pengembangan model dengan mengotomatiskan seluruh tahapan penting dalam siklus *machine learning*, sehingga pengguna dapat fokus

pada interpretasi hasil tanpa harus memahami algoritma secara teknis mendalam (Ibm, 2025).

Gambar 2. 2 IBM Cloud Pak For Data
Sumber: Data Penelitian, 2025

Proses *AutoAI* dimulai dengan menerima data dari *file* terstruktur, seperti CSV, lalu melanjutkan ke tahap persiapan data, pemilihan jenis model, hingga membangun dan memberi peringkat *pipeline* model yang dihasilkan. *AutoAI* akan menampilkan *pipeline* terbaik berdasarkan kriteria evaluasi tertentu (misalnya akurasi), dan pengguna dapat menyimpan *pipeline* tersebut sebagai model siap pakai.

Secara umum, *AutoAI* secara otomatis menjalankan beberapa tugas utama berikut:

1. Data *Pre-processing*: membersihkan, mengubah, dan mempersiapkan data agar sesuai untuk proses pelatihan model, termasuk penanganan nilai kosong dan normalisasi.

2. Pemilihan Model Otomatis (*Automated Model Selection*): memilih jenis algoritma yang paling sesuai dari berbagai kandidat model klasifikasi atau regresi berdasarkan karakteristik data.
3. Rekayasa Fitur Otomatis (*Automated Feature Engineering*): menghasilkan fitur baru dari fitur yang ada untuk meningkatkan performa model, menggunakan teknik transformasi statistik dan pemilihan fitur terbaik.
4. Optimasi *Hyperparameter* (*Hyperparameter Optimization*): menyesuaikan parameter model secara otomatis untuk mendapatkan kinerja prediktif yang maksimal dengan cara iteratif dan berbasis evaluasi metrik.

2.2. Penelitian Terdahulu

Berikut beberapa penelitian sebelumnya yang menjadi acuan dalam penelitian ini pada uraian berikut:

1. Penelitian "Prediksi Penyakit Diabetes Menggunakan Algoritma *Support Vector Machine* (SVM)" mengangkat permasalahan rendahnya pemanfaatan algoritma SVM *Radial Basis Function* (RBF) dalam prediksi penyakit diabetes. Dengan menggunakan dataset Pima Indian Diabetes dan metode *Forward Selection* untuk pemilihan fitur, algoritma SVM RBF berhasil mencapai akurasi sebesar 91,2%. Fokus utama terletak pada proses *preprocessing* data, terutama pada atribut glukosa dan insulin yang mengandung banyak nilai kosong (Hovi et al., 2022).
2. Penelitian "Implementasi *Data Mining* dalam Melakukan Prediksi Penyakit Diabetes Menggunakan Metode *Random Forest* dan *XGBoost*" membandingkan dua algoritma klasifikasi pada dataset dari Kaggle. Penelitian

ini menerapkan teknik *preprocessing* serta evaluasi dengan *5-fold cross-validation*. Hasil menunjukkan bahwa *XGBoost* memberikan akurasi lebih tinggi (76%) dibandingkan *Random Forest* (74%). Aspek penting dalam penelitian ini ialah pemilihan parameter model secara optimal (Salsabil et al., 2024).

3. Penelitian "Prediksi Penyakit Diabetes untuk Pencegahan Dini dengan Metode *Regresi Linear*" menggunakan algoritma *Regresi Linear* pada dataset dengan dua parameter utama, yaitu glukosa dan insulin. Evaluasi dilakukan dengan metrik RMSE dan menghasilkan nilai 0.000, menunjukkan tingkat presisi tinggi. Namun, terbatasnya jumlah fitur sehingga model kurang dapat digeneralisasi secara luas (Niko et al., 2023).

4. Penelitian "Prediksi Penyakit Diabetes dengan Metode *K-Nearest Neighbor* (KNN) dan Seleksi Fitur *Information Gain*" membandingkan performa algoritma KNN sebelum dan sesudah dilakukan seleksi fitur. Dataset dari Kaggle yang digunakan memiliki 70.693 data dan 18 atribut. Setelah seleksi fitur, akurasi meningkat dari 69,11% menjadi 72,93%. Kendala penelitian ini adalah pemilihan nilai K yang tepat serta proses praproses yang cukup rumit (Devian et al., 2024).

5. Penelitian "Klasifikasi Penyakit Diabetes Mellitus Berdasarkan Faktor-Faktor Penyebab Diabetes Menggunakan Algoritma C4.5" menerapkan algoritma C4.5 pada dataset Pima Indians Diabetes dan menggunakan teknik *heatmap* untuk seleksi fitur. Hasil menunjukkan akurasi sebesar 76%, lebih tinggi dibandingkan SVM dalam studi sebelumnya. Pemilihan atribut yang tepat

menjadi aspek krusial agar hasil klasifikasi lebih optimal (Fadhillah et al., 2022).

6. Penelitian *"Diabetes Prediction Using Machine Learning and Explainable AI Techniques"* mengombinasikan dataset Pima Indian dan data RTML lokal dari Bangladesh. Penelitian ini menggunakan berbagai algoritma seperti XGBoost, SVM, *Random Forest*, dan KNN, serta pendekatan *explainable* AI menggunakan LIME dan SHAP. Model *XGBoost* + ADASYN menunjukkan performa terbaik dengan akurasi 81% dan AUC 0.84. Penelitian ini juga menghasilkan aplikasi web dan Android untuk prediksi *real-time* (Tasin et al., 2023).

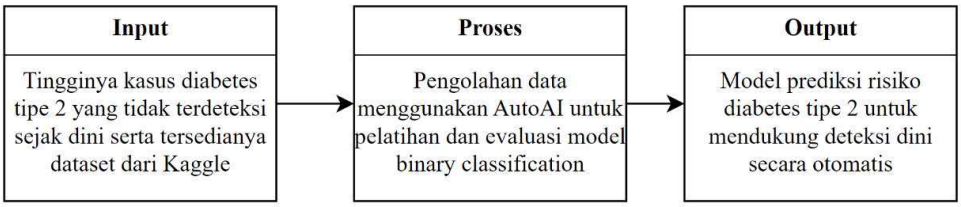
7. Penelitian "Prediksi Status Pengiriman Barang Menggunakan Metode *Machine Learning*" memanfaatkan *Random Forest*, ANN, dan *Logistic Regression* untuk memprediksi status keterlambatan pengiriman barang. Data yang digunakan berasal dari proses logistik IATA. Hasil evaluasi menunjukkan bahwa *Random Forest* memiliki akurasi tertinggi sebesar 76,6%. Penelitian ini menunjukkan penerapan *machine learning* tidak terbatas pada bidang medis, namun juga dapat digunakan dalam efisiensi operasional logistik (Pambudi et al., 2020).

8. Penelitian "Prediksi Tingkat Obesitas Menggunakan *Neural Network*: Pendekatan Klasifikasi Biner" menerapkan metode *Neural Network* untuk memprediksi tingkat obesitas dari data kesehatan penduduk Meksiko, Peru, dan Kolombia. Dataset terdiri dari 2.111 data dan diklasifikasi ulang menjadi

dua kelas. Hasil evaluasi menunjukkan akurasi tinggi yaitu 96,84% dan *F1-score* 97,91%. Proses konversi data kategorik dan seleksi fitur relevan menjadi langkah penting dalam keberhasilan pemodelan ini (Pambudi et al., 2020).

2.3. Kerangka Pemikiran

Kerangka pemikiran disebut gambaran sistematis dan logis yang menjelaskan cara peneliti berpikir untuk menjawab masalah dan mencapai tujuan penelitian.



Gambar 2. 3 Kerangka Pemikiran
Sumber: Data Penelitian, 2025

Berikut penjelasan dari kerangka pemikiran dalam gambar 2.2 yang terdiri dari *input*, proses, dan *output* dalam penelitian ini:

1. *Input*

   Data yang digunakan dalam penelitian ini berasal dari *platform* Kaggle, yaitu dataset diabetes yang memuat 8 parameter. Dataset ini menjadi dasar untuk membangun sistem prediksi risiko diabetes tipe 2.

2. Proses

   Data yang telah dikumpulkan diproses menggunakan teknologi *AutoAI* pada IBM Cloud Pak for Data. Proses ini mencakup tahapan *preprocessing* data, pembagian data untuk pelatihan, pemilihan fitur, penerapan algoritma *binary classification*, hingga evaluasi performa model secara otomatis.

3. *Output*

   Hasil dari proses ini adalah sebuah model prediksi risiko diabetes tipe 2 yang akurat dan efisien. Dapat dijadikan sebagai dasar rekomendasi dalam sistem deteksi dini penyakit diabetes.