BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Untuk memperkuat landasan teori dan memberikan konteks terhadap penelitian yang dilakukan, bagian ini menyajikan ringkasan beberapa penelitian terdahulu yang relevan. Penelitian-penelitian ini dipilih berdasarkan kesesuaian topik, metode yang digunakan, serta kontribusinya terhadap pengembangan analisis segmentasi pelanggan dan prediksi perilaku konsumen yang disajikan dalam Tabel 2.1.

Tabel 2 1 Tabel Penelitian Terdahulu

Penelitian Terdahulu 1	
Judul	Segmentasi Pelanggan Berdasarkan Perilaku Penggunaan Kartu Kredit Menggunakan Metode *K-Means Clustering* [11]
Penulis	Fatimah Defina Setiti Alhamdani, Ananda Ayu Dianti, Yufis Azhar
Jurnal	JISKA (Jurnal Informatika Sunan Kalijaga) – SINTA 3
Tahun	2021
Metode	*K-Means*, *Agglomerative Clustering*, GMM, DBSCAN
Permasalahan	Permasalahan yang diangkat dalam penelitian terdahulu adalah bagaimana mengelompokkan pelanggan kartu kredit berdasarkan pola perilaku penggunaannya, dengan tujuan utama menyusun strategi pemasaran yang lebih efektif dan tersegmentasi. Fokus penelitian tersebut terletak pada eksplorasi perilaku transaksi dan preferensi pelanggan untuk membentuk klaster yang dapat dijadikan dasar dalam pengambilan keputusan pemasaran yang lebih tepat sasaran.
Hasil Penelitian	Penelitian ini berhasil mengelompokkan pelanggan ke dalam beberapa klaster yang memiliki karakteristik berbeda, berdasarkan pola perilaku penggunaan kartu kredit dari total 9.000 entri data yang terdiri atas 18 fitur karakteristik. Dari empat metode *clustering* yang dibandingkan, algoritma *K-Means* menunjukkan performa terbaik dengan akurasi sebesar 0,207014 dan menghasilkan tiga klaster utama.
Penelitian Terdahulu 2	
Judul	Optimalisasi Strategi Pemasaran dengan Segmentasi Pelanggan Menggunakan Penerapan *K-Means Clustering* pada Transaksi *Online* Retail [12]
Penulis	Eriskiannisa Febrianty Luchia Awalina, Woro Isti Rahayu
Jurnal	JATI (Jurnal Teknologi dan Informasi) - SINTA 4
Tahun	2024
Metode	*K-Means Clustering*
Permasalahan	Perusahaan menghadapi tantangan dalam memahami perilaku pelanggan akibat kompleksitas dan volume data yang besar. Oleh karena itu, diperlukan pendekatan segmentasi berbasis data guna mengelompokkan pelanggan secara lebih terstruktur, sehingga dapat meningkatkan efektivitas strategi pemasaran yang dijalankan.
Hasil Penelitian	Metode *K-Means Clustering* merupakan salah satu pendekatan yang direkomendasikan untuk digunakan dalam menyelesaikan permasalahan segmentasi pelanggan. Dalam penelitian ini, proses segmentasi dilakukan

	terhadap 54.910 entri data pelanggan yang terdiri dari 8 variabel, yang sebelumnya telah melalui tahap *Exploratory Data Analysis* (EDA). Tahap eksplorasi ini bertujuan untuk mengidentifikasi pola tersembunyi, tren penjualan, serta karakteristik pelanggan yang tidak tampak secara eksplisit dalam data mentah. Setelah proses EDA selesai, algoritma K-Means diterapkan untuk mengelompokkan pelanggan ke dalam *cluster* yang memiliki kemiripan perilaku. Penentuan jumlah klaster dilakukan menggunakan *Elbow Method*, yaitu dengan mengevaluasi nilai *Within-Cluster Sum of Squares* (WCSS) untuk berbagai kemungkinan jumlah *cluster*. Berdasarkan hasil analisis, titik siku (*elbow point*) pada grafik WCSS menunjukkan bahwa nilai optimal tercapai ketika k = 4. Oleh karena itu, empat klaster dibentuk berdasarkan tiga fitur utama yang dianggap paling relevan terhadap tujuan segmentasi.
Penelitian Terdahulu 3	
Judul	Segmentasi Pelanggan dengan Algoritma *Clustering* Berdasarkan Atribut *Recency, Frequency* dan *Monetary* (RFM) [13]
Penulis	Muhamad Fikri Fadhillah, Aldo Lovely Arief Suyoso, Ira Puspitasari
Jurnal	MALCOM: *Indonesian Journal of Machine Learning and Computer Science* - SINTA 4
Tahun	2025
Metode	RFM *(Recency, Frequency, Monetary), K-Means, Agglomerative,* DBSCAN
Permasalahan	Perubahan karakteristik konsumen sebagai dampak dari dinamika perdagangan bebas internasional, pandemi COVID-19, serta penurunan signifikan dalam penjualan dan profitabilitas pada sektor *Business-to-Business* (B2B) telah menimbulkan tantangan baru bagi perusahaan. Salah satu tantangan utama adalah kesulitan dalam memahami perilaku pelanggan secara menyeluruh, terutama di tengah volume data transaksi yang sangat besar dan belum terstruktur. Selain itu, belum adanya pemetaan pelanggan yang komprehensif berdasarkan nilai loyalitas maupun potensi *churn* menghambat perusahaan dalam merancang strategi retensi yang efektif dan responsif terhadap dinamika pasar yang semakin kompetitif.
Hasil Penelitian	Dari total 118.314 entri transaksi yang berasal dari 1.570 pelanggan, algoritma *K-Means Clustering* menunjukkan performa segmentasi yang optimal berdasarkan sejumlah matrik evaluasi. Hasil pengukuran menunjukkan nilai *Silhouette Score* sebesar 0,364, *Davies-Bouldin Index* (DBI) sebesar 0,93, serta *Calinski-Harabasz* Index sebesar 1303,6. Berdasarkan evaluasi tersebut, ditetapkan bahwa jumlah *cluster* optimal adalah tiga, yang secara umum merepresentasikan kategori pelanggan berdasarkan tingkat loyalitas dan potensi *churn*. Segmentasi ini memberikan dasar yang kuat untuk pengembangan strategi retensi dan personalisasi layanan yang lebih efektif.
Penelitian Terdahulu 4	
Judul	Analisa Perputaran Pelanggan XYZ *Gym* dengan Pendekatan *Machine Learning* [14]
Penulis	Febriyanti Nainggolan, Indra Kelana Jaya, Yolanda Rumapea
Jurnal	Methotika: Jurnal Ilmiah Teknik Informatika - GARUDA
Tahun	2023
Metode	*Support Vector Machine* (SVM)
Permasalahan	Banyak pelanggan yang berhenti menjadi *member gym* XYZ. Diperlukan analisis *churn* pelanggan untuk memprediksi kemungkinan pelanggan kembali.
Hasil Penelitian	Model SVM memberikan akurasi sebesar 79% dalam memprediksi kemungkinan pelanggan kembali berlangganan.
Penelitian Terdahulu 5	
Judul	*Prediction and Clustering of Bank Customer Churn Based on XGBoost and K-means* [15]
Penulis	Tiansheng Zhang
Jurnal	BCP *Business* & *Management* GEBM - DOAJ
Tahun	2022

Metode	*XGBoost* dan *K-Means*
Permasalahan	Tingginya tingkat *churn* di industri perbankan sebagai dampak dari persaingan yang ketat antar bank, menyebabkan bank perlu menemukan cara untuk memprediksi dan mengelola pelanggan yang berisiko berhenti menggunakan layanan mereka.
Hasil Penelitian	XGBoost menghasilkan akurasi 0.84, *precision* 0.83, *recall* 0.84, dan F1-*score* 0.84 dalam memprediksi *churn* pelanggan bank dari dataset pelanggan bank AS di Kaggle. K-means mengelompokkan pelanggan yang *churn* ke dalam 5 klaster dengan karakteristik yang berbeda, memberikan wawasan bagi bank untuk menentukan strategi pemulihan yang sesuai untuk masing-masing klaster
Penelitian Terdahulu 6	
Judul	*Sales Prediction and Product Recommendation Model Through User Behavior Analytics* [16]
Penulis	Xian Zhao, Pantea Keikhosrokiani
Jurnal	Computers, Materials & Continua (CMC) – Scopus Q2
Tahun	2022
Metode	RFM Analysis, XGBoost, Random Forest, Apriori, K-Fold Cross Validation
Permasalahan	Transformasi bisnis dari B2B ke B2C setelah pandemi dan diperlukan sistem prediksi penjualan dan rekomendasi produk berdasarkan perilaku pelanggan
Hasil Penelitian	Model XGBoost memiliki performa terbaik dengan akurasi 77,82%, F1-score 0,7888 dan AUC 0,8524; Apriori digunakan untuk membangun sistem rekomendasi produk yang efektif
Penelitian Terdahulu 7	
Judul	*Comparative Study of XGBoost, Random Forest, and Logistic Regression Models for Predicting Customer Interest in Vehicle Insurance* [17]
Penulis	Gregorius Airlangga
Jurnal	Sinkron: Jurnal dan Penelitian Teknik Informatika – SINTA 4
Tahun	2024
Metode	XGBoost, Random Forest, Logistic Regression, SMOTE, Cross-Validation
Permasalahan	Kebutuhan untuk memprediksi minat pelanggan terhadap asuransi kendaraan dengan data tidak seimbang (*imbalanced*).
Hasil Penelitian	XGBoost memberikan *recall* tertinggi (95.25%) dan AUC-ROC (0.7854), tetapi *precision* rendah (25.85%). Random Forest lebih seimbang antara *precision* dan *recall*.
Penelitian Terdahulu 8	
Judul	K-Means dan XGBoost untuk Analisis Perilaku Pembayaran Rekening Listrik Pelanggan (Studi Kasus: PLN ULP Panakkukang) [18]
Penulis	Raditya Hari Nugraha, Diana Purwitasari, Agus Budi Raharjo
Jurnal	JUTI: Jurnal Ilmiah Teknologi Informasi - SINTA 2
Tahun	2022
Metode	K-Means *Clustering*, XGBoost, *Hyperparameter Optimization* (*Bayesian*, *Hillclimbing*, *Random Search*), PCA
Permasalahan	Tingginya jumlah pelanggan PLN yang menunggak, serta belum adanya strategi berbasis data untuk memetakan dan memprediksi pelanggan menunggak.
Hasil Penelitian	Model gabungan K-Means + XGBoost dengan optimasi bayesian menghasilkan akurasi 89,27% dan AUC 0,923. Ditemukan bahwa pelanggan kategori subsidi dan yang sering mengalami pemadaman lebih berpotensi menunggak.
Penelitian Terdahulu 9	
Judul	*Bakery Demand Forecasting Using XGBoost and K-Means Clustering* [19]
Penulis	Nathaniel Wikamulia, Maverick Jonathan, Sani Muhamad Isa
Jurnal	ICIC Express Letters, Part B: Applications - Scopus
Tahun	2023
Metode	K-Means *Clustering*, XGBoost

Permasalahan	Permintaan roti sulit diprediksi, banyak produk dikembalikan karena tak laku dijual dan perlu model prediksi yang akurat.
Hasil Penelitian	Pendekatan dengan XGBoost untuk tiap pelanggan menghasilkan akurasi 86,43%, lebih tinggi dari pendekatan dengan K-Means (54,52%).
Penelitian Terdahulu 10	
Judul	Prediksi Calon Pembeli Mobil Potensial Menggunakan Algoritma Logistic Regression [20]
Penulis	Nouval Trezandy Lapatta, Abdullah Husin
Jurnal	Sistemasi: Jurnal Sistem Informasi - SINTA 4
Tahun	2024
Metode	Logistic Regression, EDA, Confusion Matrix, ROC Curve
Permasalahan	Sulitnya mengidentifikasi calon pembeli mobil secara tepat untuk mendukung strategi pemasaran industri otomotif.
Hasil Penelitian	Model Logistic Regression menghasilkan akurasi dan presisi sebesar 95%, dengan faktor-faktor signifikan seperti usia, penghasilan, kepemilikan mobil, dan status pernikahan.

Tabel 2.1 merupakan penelitian terdahulu yang terdiri dari tujuh artikel jurnal nasional dan tiga jurnal internasional yang relevan dengan topik segmentasi dan prediksi perilaku pelanggan. Ketujuh jurnal nasional diperoleh dari sumber yang terindeks SINTA (level 2 hingga 4) dan Google Scholar, sedangkan ketiga jurnal internasional telah terindeks DOAJ atau Scopus. Keseluruhan studi tersebut secara umum menggunakan metode *K-Means Clustering* untuk segmentasi pelanggan dan algoritma prediktif seperti *XGBoost* untuk memodelkan perilaku atau keputusan pembelian.

Berdasarkan hasil tinjauan dari berbagai penelitian terdahulu, sebagian besar studi telah berhasil menerapkan metode segmentasi pelanggan maupun prediksi perilaku pelanggan secara terpisah. Berbagai pendekatan seperti RFM *analysis*, *K-Means Clustering*, dan *XGBoost* telah banyak digunakan untuk menganalisis karakteristik pelanggan dalam konteks industri perbankan, ritel, asuransi, dan layanan digital lainnya. Karakteristik yang umum diidentifikasi dalam segmentasi tersebut meliputi frekuensi kunjungan, loyalitas, pola transaksi, serta preferensi waktu dan metode pembayaran. Namun, sangat sedikit penelitian yang secara khusus mengkaji perilaku pelanggan dalam industri kebugaran, terutama terkait penggunaan layanan tambahan seperti *personal trainer* (PT) yang bersifat *personal one-on-one* dan memiliki nilai strategis dalam retensi pelanggan.

Selain itu, sebagian besar studi terdahulu hanya berfokus pada segmentasi atau prediksi secara terpisah, tanpa mengintegrasikan keduanya sebagai pendekatan terpadu. Padahal, integrasi segmentasi berbasis aktivitas pelanggan dengan model prediktif berbasis *machine learning* dapat memberikan hasil yang lebih holistik dan *actionable* dalam mendukung pengambilan keputusan strategis. Dengan demikian, penelitian ini mengisi celah tersebut dengan mengembangkan segmentasi pelanggan berbasis pola aktivitas dan preferensi kunjungan menggunakan *K-Means Clustering*, serta membangun model prediksi pembelian layanan *personal trainer* menggunakan algoritma *XGBoost* yang mempertimbangkan hasil segmentasi sebagai fitur tambahan. Penelitian ini diharapkan mampu memberikan kontribusi praktis bagi pusat kebugaran dalam merancang strategi pemasaran yang lebih tepat sasaran dan berbasis data.

2.2 Teori Penelitian

2.2.1 Gym

Pusat kebugaran atau *gym* merupakan fasilitas yang menyediakan berbagai peralatan serta program latihan fisik untuk mendukung kesehatan dan kebugaran individu. Kualitas layanan dan fasilitas yang ditawarkan sangat mempengaruhi tingkat kepuasan dan loyalitas pelanggan. Persepsi terhadap layanan *gym* dipengaruhi oleh faktor-faktor seperti gender dan usia, yang turut menentukan efektivitas strategi peningkatan layanan. Preferensi pelanggan dapat berbeda-beda misalnya seperti pria cenderung lebih memperhatikan kelengkapan alat dan intensitas latihan, sedangkan wanita lebih mengutamakan aspek kebersihan, keamanan, dan kenyamanan. Selain itu, perbedaan kebutuhan juga terlihat antar kelompok usia, di mana pengguna berusia lebih tua memerlukan pendekatan latihan yang berbeda dibandingkan kelompok usia muda. Dengan memahami variasi persepsi ini, pengelola pusat kebugaran dapat merancang strategi layanan yang lebih tepat sasaran untuk meningkatkan kepuasan dan loyalitas anggota *gym* [3].

Dengan demikian, evaluasi kualitas layanan di pusat kebugaran melalui analisis kinerja dan kepentingan menunjukkan bahwa aspek

lingkungan fisik, program latihan, dan kualitas instruktur merupakan faktor utama yang mempengaruhi kepuasan pelanggan [21].

2.2.2 *Personal Trainer* (PT)

Personal Trainer (PT) merupakan profesional di bidang kebugaran yang memiliki tanggung jawab dalam merancang program latihan secara individual, serta memberikan pendampingan dan pengawasan selama proses pelatihan guna membantu klien mencapai tujuan kebugarannya secara optimal [4]. Selain itu, PT juga berperan dalam memastikan penerapan teknik latihan yang benar, mengatur tingkat intensitas beban latihan sesuai kemampuan klien, serta melakukan pemantauan berkala terhadap perkembangan yang dicapai. Peran strategis ini bertujuan untuk memaksimalkan efektivitas program latihan sekaligus meminimalisasi risiko cedera.

Seiring dengan meningkatnya pertumbuhan industri kebugaran dalam satu dekade terakhir, permintaan terhadap layanan *Personal Trainer* (PT) turut menunjukkan tren kenaikan yang signifikan. Berdasarkan laporan survei tren kebugaran global tahun 2022, pelatihan yang dilakukan di bawah pengawasan langsung PT menempati posisi teratas di Brasil, peringkat ketiga di kawasan Eropa, dan berada di urutan kedelapan di Amerika Serikat [22]. Temuan ini menggarisbawahi pentingnya peran PT dalam menyediakan pendekatan latihan yang lebih terpersonalisasi, sehingga mampu mendukung pencapaian target kebugaran individu secara lebih efektif dan terarah.

Latihan yang dilakukan di bawah pengawasan langsung *Personal Trainer* (PT) cenderung mendorong klien untuk menggunakan beban yang lebih tinggi serta meningkatkan intensitas latihan dibandingkan saat berlatih secara mandiri. Pendekatan ini terbukti memberikan kontribusi nyata terhadap peningkatan kekuatan otot dan penambahan massa tubuh bebas lemak (*lean body mass*) secara signifikan [4]. Selain itu, pengawasan teknis yang diberikan oleh PT berperan penting dalam meminimalkan risiko cedera akibat kesalahan postur atau teknik

pelaksanaan gerakan. Oleh karena itu, keterlibatan PT dalam sesi latihan tidak hanya mempercepat pencapaian target kebugaran, tetapi juga meningkatkan aspek keselamatan dan efisiensi program latihan yang dijalankan.

2.2.3 Segmentasi Pelanggan

Segmentasi pelanggan merupakan pendekatan penting dalam memahami keragaman karakteristik dan perilaku konsumen. Secara umum, segmentasi pelanggan merujuk pada proses membagi pasar atau populasi pelanggan menjadi kelompok-kelompok yang memiliki karakteristik, kebutuhan, atau perilaku serupa. Tujuannya adalah agar organisasi dapat menyusun strategi pemasaran, komunikasi, atau pelayanan yang lebih relevan dan efektif bagi masing-masing segmen pelanggan [23].

Segmentasi pelanggan dapat didasarkan pada berbagai dimensi, seperti karakteristik demografis, psikografis, perilaku, hingga preferensi dalam berinteraksi dengan merek atau layanan tertentu. Dalam teori pemasaran modern, pemahaman terhadap perbedaan individu dalam persepsi dan respons terhadap stimulus pemasaran menjadi sangat penting. Pendekatan ini memperhitungkan bahwa satu populasi pelanggan tidak selalu bersifat homogen dalam persepsi, sikap, atau intensi perilakunya terhadap suatu produk atau pesan pemasaran [23].

Seiring berkembangnya teknologi digital dan komunikasi interaktif, penting bagi organisasi untuk tidak lagi melihat pelanggan sebagai satu kesatuan tunggal, melainkan sebagai individu yang berada dalam segmen-segmen tertentu dengan kebutuhan dan orientasi berbeda. Pendekatan segmentasi memungkinkan penyampaian pesan dan strategi yang lebih personal, yang pada akhirnya meningkatkan efektivitas kampanye pemasaran dan membangun hubungan jangka panjang dengan pelanggan [23].

Selain itu, dengan menerapkan segmentasi yang tepat, perusahaan dapat mengidentifikasi kelompok pelanggan yang lebih responsif terhadap pesan pemasaran, memahami faktor-faktor kunci yang

mempengaruhi loyalitas atau konversi, serta mengoptimalkan alokasi sumber daya dalam merancang produk atau layanan sesuai dengan target pasar yang spesifik [23].

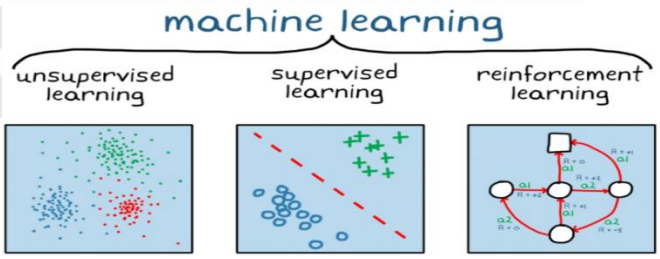
2.2.4 Prediksi Potensial Pembelian Layanan

Prediksi potensi pembelian layanan merupakan proses untuk memperkirakan kemungkinan seorang pelanggan akan membeli produk atau layanan. Proses ini penting untuk membantu perusahaan dalam perancangan strategi pemasaran yang lebih terarah, memaksimalkan konversi penjualan, dan mengelola sumber daya secara efisien. Dalam konteks ini, niat pembelian pelanggan tidak hanya dipengaruhi oleh faktor internal seperti kebutuhan pribadi, minat, atau loyalitas terhadap merek, tetapi juga oleh faktor eksternal seperti musim, tren pasar, dan aktivitas promosi. Oleh karena itu, pendekatan prediktif perlu mempertimbangkan banyak aspek yang bersifat dinamis dan kontekstual [24].

Model prediksi pembelian umumnya dibangun menggunakan data historis pelanggan, seperti riwayat transaksi, frekuensi pembelian, serta pola interaksi dengan produk atau platform digital. Data ini kemudian diolah menggunakan teknik pembelajaran mesin, khususnya model klasifikasi untuk mengidentifikasi pola-pola yang berhubungan dengan keputusan pembelian. Pemanfaatan metode prediksi ini menjadi semakin krusial di tengah persaingan pasar yang dinamis dan meningkatnya ekspektasi personalisasi dari konsumen [24].

2.3 *Framework* dan Algoritma Penelitian

2.3.1 Machine Learning



Gambar 2. 1 *Machine Learning*

Sumber: AiOps [25]

Machine Learning (ML) adalah cabang dari *Artificial Intelligence* yang berfokus pada pengembangan algoritma dan kerangka kerja untuk dapat mengenali pola serta menarik kesimpulan secara otomatis dari data. Pendekatan ini memungkinkan sistem belajar dari data historis tanpa memerlukan pemrograman eksplisit dalam setiap situasi. Dalam praktiknya, *machine learning* dibagi menjadi tiga kategori utama seperti yang terlihat pada gambar 2.1, yaitu *unsupervised learning* (pembelajaran tanpa label), *supervised learning* (pembelajaran dengan label), dan *reinforcement learning* (pembelajaran berbasis umpan balik dari lingkungan) [6].

a) Supervised Learning

Supervised learning adalah yang dilakukan menggunakan dataset yang telah dilengkapi dengan label sehingga memungkinkan algoritma untuk mempelajari keterkaitan antara variabel input dan output secara terarah. Pendekatan ini banyak digunakan dalam permasalahan klasifikasi maupun regresi, di mana model bertujuan untuk membangun pemahaman yang dapat digeneralisasikan dari pola-pola dalam data pelatihan guna melakukan prediksi terhadap data baru yang belum dilabeli.

b) Unsupervised Learning

Unsupervised learning adalah pendekatan pembelajaran mesin yang diterapkan pada data tanpa label, dengan fokus utama pada pengenalan pola tersembunyi atau struktur alami yang terkandung dalam dataset. Teknik ini bertujuan untuk mengeksplorasi dan memahami karakteristik internal data tanpa adanya panduan atau kategori yang telah ditentukan sebelumnya.

c) Reinforcement Learning

Reinforcement Learning adalah pendekatan pembelajaran mesin berbasis penguatan yang mengandalkan mekanisme sistem umpan balik (*feedback control*). Dalam metode ini, model memperoleh pengetahuan melalui proses interaktif dengan lingkungannya, di mana setiap aksi atau keputusan yang diambil

akan mempengaruhi kondisi lingkungan selanjutnya. Seiring waktu, model secara bertahap mempelajari strategi atau kebijakan optimal (*optimal policy*) untuk memaksimalkan imbal hasil (*reward*) berdasarkan pengalaman yang terkumpul dari interaksi tersebut.

2.3.2 Data Mining

Data mining merupakan proses mencari pola tersembunyi dan pengetahuan baru dari data dalam jumlah besar menggunakan teknik statistik, *machine learning*, dan sistem basis data [26]. Proses ini memungkinkan data mentah diubah menjadi *insight* untuk mendukung pengambilan keputusan yang lebih baik. Dalam ranah bisnis, *data mining* banyak dimanfaatkan untuk menganalisis perilaku historis pelanggan dan mengidentifikasi pola-pola penting yang dapat digunakan dalam strategi pemasaran dan pelayanan [27]. Dengan analisis ini, perusahaan dapat mengoptimalkan keputusan berbasis data yang lebih akurat.

Untuk mengelola proses analisis data secara sistematis, pendekatan CRISP-DM (*Cross Industry Standard Process for Data Mining*), banyak digunakan. CRISP-DM terbagi ke dalam enam tahap utama: pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan *deployment* [28]. Dalam penelitian ini, konsep *data mining* diterapkan untuk segmentasi pelanggan dan prediksi pembelian layanan *Personal Trainer* (PT) berdasarkan karakteristik historis pelanggan.

2.3.3 CRISP-DM

Gambar 2. 2 CRISP-DM

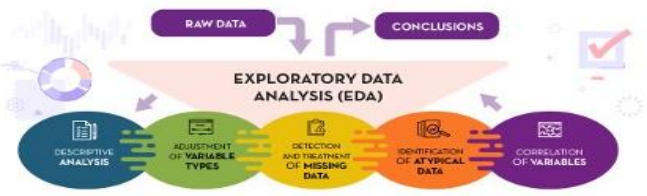
Sumber: DJKN [29]

Gambar 2.2 menggambarkan tahapan CRISP-DM (*Cross-Industry Standard Process for Data Mining*) yaitu model proses yang paling banyak diterapkan dalam proyek *data mining* lintas industri. Kerangka ini hingga saat ini tetap menjadi standar *de-facto* dalam praktik dan penelitian *data mining*. CRISP-DM membagi proses *data mining* menjadi enam fase yang saling terkait [28]:

A. *Business Understanding*: Memahami tujuan bisnis, merumuskan tujuan *data mining*, dan menetapkan kriteria keberhasilan.
B. *Data Understanding*: Mengumpulkan data awal, mengeksplorasi karakteristik data, dan menilai kualitas data.
C. *Data Preparation*: Memilih data, membersihkan data, dan membangun fitur yang relevan untuk *modeling*.
D. *Modeling*: Memilih teknik *modeling* yang sesuai, membangun model, serta mengkalibrasi parameter model.
E. *Evaluation*: Mengevaluasi hasil model terhadap tujuan bisnis yang telah ditetapkan.
F. *Deployment*: Menerapkan hasil analisis ke dalam lingkungan operasional, seperti membuat laporan atau membangun sistem berbasis model.

Keunggulan CRISP-DM terletak pada fleksibilitasnya yang dapat diterapkan di berbagai domain industri, serta strukturnya yang sistematis namun cukup adaptif untuk berbagai jenis proyek *data mining*. Dalam penelitian ini, CRISP-DM digunakan sebagai kerangka kerja metodologi untuk mengarahkan proses segmentasi pelanggan dan prediksi pembelian layanan *Personal Trainer* (PT).

2.3.4 *Exploratory Data Analysis* (EDA)



Gambar 2. 3 *Exploratory Data Analysis* (EDA)

Sumber: Binus [30]

Gambar 2.3 menggambarkan *Exploratory Data Analysis* (EDA) yang bertujuan untuk memahami struktur, pola, dan karakteristik utama dari dataset sebelum dilakukan pemodelan lebih lanjut. Melalui EDA, peneliti dapat mengidentifikasi distribusi data, mendeteksi *outlier*, serta memahami hubungan antar variabel. Teknik-teknik yang umum digunakan dalam EDA meliputi statistik deskriptif, visualisasi data seperti *histogram*, *boxplot*, *scatter plot*, dan *heatmap*, serta analisis korelasi antar variabel. Dengan melakukan EDA, peneliti dapat memperoleh wawasan awal yang penting untuk menentukan langkah-langkah selanjutnya dalam proses analisis data, seperti pemilihan fitur, penanganan data yang tidak seimbang, dan pemilihan metode pemodelan yang sesuai [31].

2.3.5 Data Preprocessing

Data preprocessing merupakan tahap krusial dalam proses analisis data dan pemodelan *machine learning*. Tujuan utamanya adalah untuk membersihkan, menstandarkan, dan mempersiapkan *raw data* agar sesuai dengan kebutuhan algoritma. Tahapan ini mencakup penanganan *missing value*, duplikat, konversi tipe data, *encoding* variabel kategorikal, hingga deteksi dan penanganan *outlier*. Tanpa proses ini, model yang dibangun berisiko mengalami bias atau *error* karena kualitas data yang rendah. *Preprocessing* yang baik akan meningkatkan akurasi model serta mempercepat proses pelatihan karena data telah disederhanakan dan direpresentasikan secara optimal untuk digunakan dalam proses pembelajaran mesin [32].

2.3.6 Winsorizing

Penanganan *outlier* merupakan salah satu tahapan penting dalam proses *preprocessing* data, mengingat *outlier* adalah nilai-nilai ekstrim yang secara signifikan berbeda dari sebagian besar data lainnya. Salah satu teknik yang dapat digunakan dalam mengatasi *outlier* adalah *winsorizing*, yaitu metode menggantikan nilai-nilai ekstrem dengan nilai batas tertentu yang biasanya ditentukan berdasarkan persentil data.

Dengan demikian, teknik ini tidak menghilangkan data, melainkan menyesuaikan nilai-nilai ekstrim agar tetap berada dalam rentang yang wajar dan representatif. Penggunaan *winsorizing* pada algoritma pohon klasifikasi, khususnya sebelum perhitungan indeks Gini yaitu sebuah metrik yang mengukur ketidakmurnian suatu *node* dalam pohon keputusan berdasarkan distribusi kelas dan digunakan untuk menentukan pemisahan data yang optimal sehingga meningkatkan akurasi model dengan menghasilkan pohon keputusan yang lebih stabil dan mengurangi kebutuhan akan proses pemangkasan (*pruning*) tambahan [33] [34]

2.3.7 Encoding

Encoding dalam data *preprocessing* adalah proses mengubah data kategorikal menjadi numerik yang dapat diolah dalam *machine learning*. Proses ini sangat penting karena sebagian besar model *machine learning* hanya dapat bekerja dengan data numerik, bukan data berbasis teks atau kategori. Beberapa teknik *encoding* dalam data *preprocessing* antara lain [35]:

a. *One Hot Encoding*: Mengubah kategori menjadi vektor biner (hanya satu elemen bernilai 1, sisanya 0). Misalnya, kategori warna [Merah, Kuning, Hijau] akan diubah menjadi tiga kolom: Merah, Kuning, Hijau
b. *Label Encoding*: Memberikan nomor unik pada setiap kategori. Misalnya, Kota: Jakarta = 0, Tangerang = 1, Bekasi = 2, Surabaya = 3. Cocok untuk data ordinal (ada urutan), tapi harus hati-hati untuk data nominal agar model tidak salah menganggap ada urutan.
c. *Binary Encoding*: Mengonversi kategori ke dalam kode biner, lalu membaginya ke beberapa kolom. Efisien untuk data dengan banyak kategori.
d. *Frequency Encoding*: Mengganti kategori dengan frekuensi kemunculannya dalam dataset.

e. *Ordinal Encoding*: Memberikan nilai numerik berdasarkan urutan kategori, cocok untuk data yang memang memiliki urutan.

2.3.8 Clustering

Clustering merupakan salah satu pendekatan *unsupervised learning* dalam proses data *mining*, dengan tujuan utama mengelompokkan data ke dalam sejumlah klaster berdasarkan kemiripan karakteristik tanpa memerlukan label awal. Teknik ini sangat berguna untuk mengungkap struktur laten dalam kumpulan data berukuran besar dan kompleks, serta sering diterapkan dalam berbagai domain seperti segmentasi pelanggan, analisis perilaku konsumen, klasifikasi dokumen, dan sistem pendukung pengambilan keputusan [33].

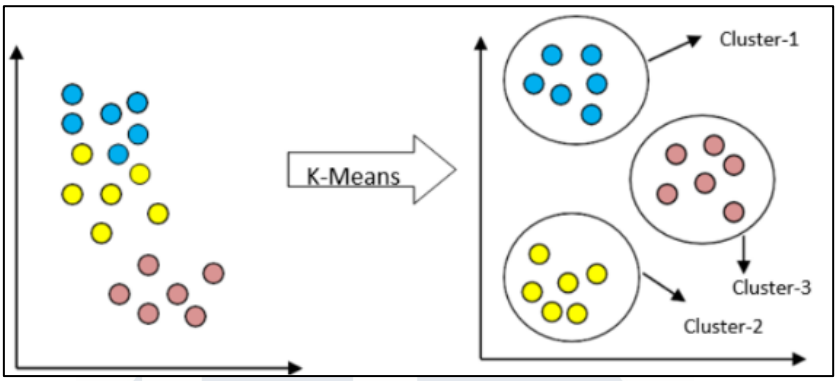
Prinsip dasar *clustering* adalah memisahkan data ke beberapa kelompok (*clusters*) sedemikian rupa agar data dalam satu klaster memiliki tingkat kemiripan yang tinggi, sementara perbedaan antar klaster menjadi signifikan. Tujuan proses ini adalah untuk meminimalkan jarak dalam klaster (*intra-cluster distance*) dan memaksimalkan jarak antar klaster (*inter-cluster distance*).

Proses pengelompokan biasanya dimulai dengan pemilihan metrik jarak tertentu sebagai dasar penilaian kedekatan antar data. Selanjutnya, algoritma *clustering* akan menyesuaikan posisi pusat klaster secara iteratif hingga mencapai kondisi konvergen atau stabil. Mengingat tidak adanya label awal, metode ini sangat sesuai untuk eksplorasi data yang belum terklasifikasi. Evaluasi terhadap hasil clustering dapat dilakukan menggunakan beberapa indikator, seperti *Silhouette Coefficient*, *Davies-Bouldin Index* (DBI), dan *Calinski-Harabasz Index* (CHI).

2.3.9 K-Means

K-Means merupakan algoritma *unsupervised learning* yang digunakan untuk mengelompokkan data ke dalam sejumlah klaster berdasarkan kemiripan karakteristik. Algoritma ini bekerja dengan menentukan jumlah klaster (k) menggunakan *Elbow Method*, lalu menempatkan setiap data ke klaster dengan pusat (*centroid*) terdekat,

kemudian memperbarui posisi *centroid* hingga pembagian klaster stabil [36].



Gambar 2. 4 *K-Means*

Sumber: Suyal M, Sharma S [36]

Prinsip utama *K-Means* adalah meminimalkan jarak antara titik data dan *centroid cluster*-nya masing-masing dan memaksimalkan perbedaan antar klaster. Algoritma ini biasanya menggunakan matrik jarak *Euclidean* untuk menentukan kedekatan, sehingga sangat efektif jika data bersifat numerik dan berada pada skala yang seragam. Proses iteratif akan terus berjalan hingga posisi *centroid* tidak lagi berubah secara signifikan atau jumlah iterasi maksimum tercapai [36].

Terdapat rumus dalam algoritma *K-Means* yaitu untuk menghitung jarak Euclidean antara setiap data dengan *centroid* dan rumus fungsi objektif (fungsi yang diminimalkan oleh *K-Means*). Berikut ini rumus dalam algoritma *K-Means*:

$$d(x_i, \mu_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - \mu_{jk})^2}$$

Rumus 2 1 Rumus Euclidean

Penjelasan Komponen Rumus 2.1:

a. xi: titik data ke-i
b. µj: *centroid* dari *cluster* ke-j
c. n: jumlah fitur

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

Rumus 2 2 Rumus *Objective Function K-Means*

Penjelasan Komponen Rumus 2.2:

a. J: total *within-cluster sum of squares* (WCSS)
b. K: jumlah *cluster*
c. Cj: himpunan data yang termasuk dalam *cluster* ke-j
d. μj: centroid dari *cluster* ke-j

Untuk memberikan pemahaman yang lebih terstruktur mengenai cara kerja algoritma *K-Means*, berikut disajikan pseudocode dari algoritma tersebut. Pseudocode adalah representasi logika algoritma dalam bentuk penulisan mirip kode program, namun tidak bergantung pada sintaksis bahasa pemrograman tertentu. Tujuannya adalah untuk menjelaskan langkah-langkah utama algoritma secara ringkas, jelas, dan mudah dipahami oleh manusia, terutama dalam konteks analisis atau dokumentasi ilmiah.

Pada algoritma *K-Means*, *pseudocode* akan menggambarkan proses inisialisasi *centroid*, penetapan keanggotaan *cluster* berdasarkan jarak terdekat, pembaruan posisi *centroid*, serta kondisi berhenti saat konvergensi tercapai. Langkah-langkah ini akan berulang hingga model mencapai hasil segmentasi pelanggan yang stabil.

Input:

1. *Dataset* D dengan n data *point*
2. Jumlah *cluster* k
3. *Maximum iteration* (*max_iter*)
4. *Tolerance* (tol) untuk konvergensi (opsional)

Output: K buah *cluster* dengan *centroid* masing-masing

Algorithm:

1. Inisialisasi k buah *centroid* secara acak dari data D

2. Untuk iterasi sebanyak *max_iter*:
    a. *Assignment Step*: Untuk setiap data *point* x di D:
        i. Hitung jarak x ke semua *centroid*
        ii. Tentukan *cluster* dengan *centroid* terdekat
        iii. *Assign* x ke *cluster* tersebut
    b. *Update Step*: Untuk setiap *cluster* j = 1 to k:
        i. Hitung rata-rata semua data *point* dalam *cluster* j
        ii. *Update centroid* j dengan nilai rata-rata tersebuT
    c. Konvergensi: Jika perpindahan *centroid* < tol untuk semua *cluster*:
        i. Hentikan iterasi (konvergen)
3. Kembalikan hasil *cluster* dan posisi *centroid*

Pseudocode di atas menjelaskan proses iteratif *K-Means* yang terdiri dari dua langkah utama: *assignment* (penentuan keanggotaan *cluster*) dan *update* (perhitungan ulang *centroid*). Algoritma terus berjalan hingga jumlah iterasi maksimum tercapai atau posisi *centroid* stabil (konvergen).

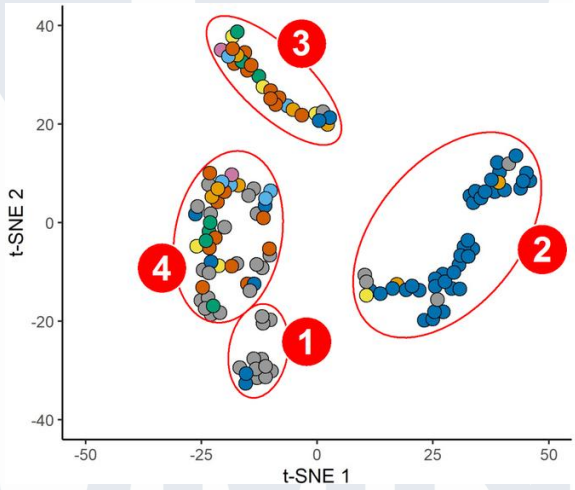
K-Means juga memiliki keterbatasan seperti keharusan menentukan jumlah klaster di awal, sensitivitas terhadap pemilihan awal *centroid*, serta kurang efektif jika bentuk klaster tidak simetris atau terdapat *outlier* yang ekstrem. Oleh karena itu, pemilihan fitur dan *preprocessing* data menjadi hal penting untuk memastikan hasil segmentasi yang optimal [36].

K-Means banyak digunakan dalam studi segmentasi pelanggan karena algoritma ini mampu mengelompokkan data berdasarkan kemiripan karakteristik tanpa memerlukan label atau target variabel. Pendekatan ini sangat sesuai untuk data pelanggan yang memiliki dimensi perilaku seperti frekuensi kunjungan, durasi keanggotaan, serta preferensi waktu, karena dapat mengungkap struktur kelompok tersembunyi yang tidak terlihat secara eksplisit. Selain itu, *K-Means* memiliki keunggulan dalam hal kecepatan komputasi dan efisiensi pada

data berukuran besar, sehingga cocok untuk kasus segmentasi dalam industri jasa seperti ritel dan kebugaran yang memiliki banyak anggota dan variabel numerik. Dengan hasil segmentasi yang lebih objektif, organisasi dapat memperoleh wawasan yang berguna untuk penyusunan strategi pemasaran yang lebih tepat sasaran.

2.3.10 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) adalah algoritma reduksi dimensi nonlinier yang dirancang untuk memetakan data berdimensi tinggi ke dalam representasi dua atau tiga dimensi. Metode ini sangat efektif dalam mengungkap struktur lokal dalam data, seperti pembentukan klaster, sehingga sering digunakan sebagai alat bantu visualisasi dalam analisis hasil *clustering* pada data yang kompleks dan berdimensi tinggi [37].



Gambar 2. 5 t-SNE

Sumber: Ha et al., 2022 [38]

Algoritma t-SNE bekerja dengan mengubah jarak antar titik dalam ruang berdimensi tinggi menjadi distribusi probabilitas yang merepresentasikan tingkat kemiripan antar titik. Semakin dekat dua titik, semakin tinggi probabilitas bahwa keduanya dianggap serupa. Selanjutnya, algoritma menghasilkan distribusi probabilitas yang sepadan di ruang berdimensi rendah, kemudian meminimalkan perbedaan antara kedua distribusi tersebut dengan menggunakan ukuran *Kullback-Leibler* (KL) *divergence* [37]

KL *divergence* sendiri merupakan ukuran yang digunakan untuk menghitung ketidaksamaan antara dua distribusi probabilitas. Dalam konteks t-SNE, nilai KL *divergence* mencerminkan seberapa besar perbedaan antara struktur data asli (dalam dimensi tinggi) dan struktur hasil proyeksi (dalam dimensi rendah). Nilai KL *divergence* yang kecil menunjukkan bahwa representasi visual yang dihasilkan berhasil mempertahankan struktur lokal data dengan baik [37].

Keunggulan utama t-SNE terletak pada kemampuannya dalam menampilkan pola lokal secara jelas dan intuitif, yang sangat berguna untuk mengevaluasi hasil segmentasi atau *clustering*. Namun demikian, algoritma ini juga memiliki sejumlah keterbatasan. t-SNE cenderung kurang mampu mempertahankan struktur global antar klaster secara konsisten dan cukup sensitif terhadap parameter seperti *perplexity* dan *learning rate*. Oleh karena itu, implementasi t-SNE memerlukan penyesuaian parameter yang hati-hati agar visualisasi yang dihasilkan dapat merefleksikan pola data secara optimal [37].

2.3.11 *Ensemble Learning*

Ensemble learning merupakan teknik dalam *machine learning* yang menggabungkan beberapa model (disebut *base learners* atau *single classifiers*) untuk menghasilkan prediksi yang lebih akurat, stabil, dan andal dibandingkan hanya mengandalkan satu model. Pendekatan ini bertujuan untuk mengurangi kesalahan, meningkatkan akurasi, serta mengurangi risiko *overfitting* dengan memanfaatkan kekuatan kolektif dari berbagai model. Model-model ini dilatih secara paralel atau berurutan pada subset data yang berbeda atau dengan pendekatan yang bervariasi, lalu hasil prediksinya digabungkan menggunakan metode seperti *majority voting, averaging*, atau *weighted voting* [39].

Beberapa teknik populer dalam *ensemble learning* antara lain *Bagging, Boosting*, dan *Stacking*. *Bagging* (*Bootstrap Aggregating*) melatih banyak model secara paralel pada subset data yang berbeda untuk mengurangi varians, contohnya adalah *Random Forest*. *Boosting*

melibatkan pelatihan model secara berurutan, di mana setiap model baru fokus memperbaiki kesalahan model sebelumnya seperti contoh umum termasuk *AdaBoost* dan *Gradient Boosting*. Sementara itu, *Stacking* menggabungkan prediksi dari beberapa model awal (level-0) dan menggunakan model tambahan (level-1) untuk menentukan prediksi akhir. Selain itu, metode sederhana seperti *voting* dan *averaging* juga digunakan untuk menggabungkan hasil prediksi beberapa model [40].

Manfaat utama dari *ensemble learning* adalah peningkatan akurasi dan keandalan prediksi, terutama pada data yang kompleks dan bervariasi. Teknik ini juga membantu mengurangi risiko *overfitting* dan *underfitting* karena kelemahan satu model dapat dikompensasi oleh model lainnya. Dengan demikian, *ensemble learning* memberikan solusi yang lebih stabil dan kuat dalam menghadapi berbagai tantangan dalam pemodelan data [40].

2.3.12 *Extreme Gradient Boosting*

Extreme Gradient Boosting (XGBoost) merupakan salah satu algoritma *machine learning* berbasis *ensemble learning* yang menggunakan teknik *boosting* secara efisien untuk meningkatkan akurasi prediksi. Algoritma ini membangun model pohon keputusan secara bertahap, di mana setiap pohon baru berupaya memperbaiki kesalahan yang dibuat oleh model sebelumnya. Keunggulan utama dari *XGBoost* terletak pada kemampuannya mengatasi masalah *overfitting* melalui mekanisme regularisasi, kecepatan komputasi yang tinggi, serta skalabilitas yang baik, sehingga sangat cocok untuk digunakan pada dataset yang besar dan kompleks [10].

Terdapat rumus dalam algoritma *XGBoost* untuk mengoptimalkan proses pembelajaran, yaitu dengan meminimalkan sebuah fungsi objektif yang menggabungkan dua komponen utama, yaitu fungsi *loss* dan fungsi regularisasi. Rumus umum dari fungsi objektif *XGBoost* adalah sebagai berikut:

$$L(\theta) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \cap (f_k)$$

Rumus 2 3 Rumus *Objective Function XGBoost*

Penjelasan Komponen Rumus 2.3:

a. L(0): fungsi objektif total yang ingin diminimalkan oleh model *XGBoost*
b. $\sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right)$: jumlah total *loss* antara prediksi dan label aktual untuk seluruh data
c. $l\left(y_i, \hat{y}_i^{(t)}\right)$: fungsi *loss*, misalnya *log loss* (klasifikasi) atau *squared error* (regresi)
d. yi: label *actual* untuk data ke-i
e. $\hat{y}_i^{(t)}$: prediksi dari model sampai iterasi *boosting* ke-t
f. $\sum_{k=1}^{t} \cap (f_k)$: penjumlahan penalti kompleksitas dari seluruh pohon fk yang telah dibangun
g. $\cap (f_k)$: fungsi regulasi untuk pohon ke-k, yang bertujuan menghindari *overfitting*
h. fk: pohon keputusan ke-k yang dibentuk pada iterasi ke-k

Melalui formulasi tersebut, *XGBoost* tidak hanya fokus pada peningkatan akurasi prediksi melalui fungsi *loss*, tetapi juga secara aktif mengendalikan kompleksitas model menggunakan fungsi regularisasi Ω(fk). Pendekatan ini memungkinkan *XGBoost* untuk menghasilkan model yang kuat namun tetap mampu generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya, sehingga sangat sesuai digunakan untuk prediksi pada kasus dengan potensi *overfitting* tinggi seperti klasifikasi pelanggan potensial.

Untuk memberikan gambaran yang lebih jelas mengenai mekanisme kerja algoritma *XGBoost*, berikut disajikan *pseudocode* dari proses *boosting* yang dilakukan. *Pseudocode* merupakan representasi logis dari algoritma yang dituliskan dalam bentuk mirip kode program, namun tidak terikat dengan aturan sintaksis bahasa pemrograman tertentu.

Pseudocode XGBoost menggambarkan proses pelatihan model secara iteratif, di mana pada setiap langkah, model baru dibentuk untuk memperbaiki kesalahan prediksi sebelumnya berdasarkan *gradien* dan *hessian* dari fungsi *loss*.

Input:

1. *Dataset* D = {(x1, y1), ..., (xn, yn)}
2. Jumlah *boosting round* T
3. Fungsi *loss* l(y, y_hat)
4. Fungsi regularisasi Ω(f)
5. Parameter *learning_rate, max_depth*, dll.

Output: Model akhir sebagai hasil penjumlahan T pohon

Algorithm:

1. Inisialisasi prediksi awal y_hat_i = 0 untuk semua data
2. Untuk t = 1 sampai T:
    a. Hitung turunan pertama (gradien) g_i = ∂l(y_i, y_hat_i)
    b. Hitung turunan kedua (hessian) h_i = ∂^2l(y_i, y_hat_i)
    c. Gunakan {g_i, h_i} untuk membangun pohon f_t:
        1) Pilih *split* dengan nilai *gain* tertinggi
        2) Bangun *tree* dengan struktur optimal
        3) Terapkan regularisasi untuk menghindari *overfitting*
    d. Update prediksi: y_hat_i ← y_hat_i + *learning_rate* × f_t(x_i)
3. Kembalikan model akhir sebagai jumlah dari semua f_t

Dari *pseudocode* di atas dapat dilihat bahwa *XGBoost* bekerja secara iteratif membangun pohon-pohon keputusan berdasarkan informasi *gradien* dan *hessian* dari fungsi *loss*. Setiap pohon bertugas memperbaiki kesalahan prediksi sebelumnya, dan hasil akhirnya merupakan penjumlahan dari seluruh pohon yang telah dibentuk. Dengan menambahkan fungsi regularisasi dan teknik seperti *early stopping*,

XGBoost mampu menghasilkan model prediktif yang akurat sekaligus tahan terhadap *overfitting*.

XGBoost merupakan salah satu algoritma klasifikasi yang sangat sesuai untuk diterapkan pada data tabular dengan jumlah fitur yang relatif banyak dan pola yang kompleks. Keunggulan utamanya terletak pada kemampuannya dalam menghasilkan prediksi yang akurat melalui teknik *boosting* yang memperbaiki kesalahan model sebelumnya, serta mekanisme regularisasi yang menjaga agar model tidak *overfitting*. Algoritma ini mampu memproses data dengan fitur yang beragam dan memanfaatkan *feature importance* untuk mengidentifikasi variabel paling berpengaruh terhadap prediksi. Selain itu, *XGBoost* juga mampu menangani data yang memiliki nilai kosong atau tidak lengkap secara langsung tanpa memerlukan imputasi tambahan [10].

Dengan struktur pohon keputusan yang terbentuk, algoritma ini juga memberikan interpretasi terhadap pentingnya setiap fitur, sehingga sangat berguna untuk mengetahui faktor-faktor yang paling mempengaruhi hasil prediksi. Oleh karena itu, *XGBoost* banyak digunakan dalam konteks klasifikasi yang memerlukan kombinasi antara performa tinggi, efisiensi waktu, dan interpretabilitas hasil.

2.3.13 Pemisahan Data

Pemisahan data (*data splitting*) merupakan langkah fundamental dalam pemodelan *machine learning* berbasis *supervised learning*. Tujuan utamanya adalah untuk memisahkan data menjadi dua subset, yaitu data latih (*training set*) dan data uji (*testing set*), agar performa model dapat dievaluasi secara objektif terhadap data yang belum pernah digunakan sebelumnya dalam pelatihan [41].

Rasio pembagian yang paling umum digunakan adalah 80:20 atau 70:30. Komposisi 80% untuk pelatihan dan 20% untuk pengujian dianggap ideal karena mampu memberikan informasi yang cukup bagi

model untuk mempelajari pola dari data, sekaligus menyediakan data yang memadai untuk proses validasi [41].

Dalam penerapannya, apabila variabel target memiliki distribusi kelas yang tidak seimbang, teknik *stratified sampling* umumnya digunakan. Strategi ini bertujuan untuk menjaga proporsi masing-masing kelas tetap konsisten di antara data latih dan data uji, sehingga distribusi kelas tetap representatif. Pendekatan ini menjadi penting untuk menghindari bias dalam evaluasi model, khususnya pada kasus klasifikasi dengan dominasi kelas mayoritas yang signifikan [41].

2.3.14 SMOTE

Ketidakseimbangan kelas (*imbalanced class*) merupakan permasalahan umum dalam pemodelan klasifikasi, di mana jumlah data pada satu kelas jauh lebih besar dibandingkan kelas lainnya. Masalah ini dapat menyebabkan model pembelajaran mesin lebih cenderung mempelajari pola dari kelas mayoritas dan mengabaikan kelas minoritas, sehingga menurunkan performa model dalam mengklasifikasikan data dari kelas minor [42].

Untuk mengatasi permasalahan tersebut, salah satu teknik yang paling banyak digunakan adalah SMOTE atau *Synthetic Minority Over-sampling Technique*. SMOTE pertama kali diperkenalkan oleh Chawla et al. pada tahun 2002 sebagai metode *oversampling* yang bertujuan untuk menyeimbangkan distribusi kelas dengan cara menghasilkan data sintetis baru dari kelas minoritas. Berbeda dengan metode *oversampling* tradisional yang hanya melakukan duplikasi data, SMOTE menciptakan contoh baru dengan memanfaatkan pendekatan berbasis tetangga terdekat (*nearest neighbor*) [42].

Cara kerja SMOTE adalah dengan memilih secara acak satu sampel dari kelas minoritas, kemudian mencari beberapa tetangga terdekat (biasanya menggunakan algoritma *k-nearest neighbors*). Selanjutnya, SMOTE akan menghasilkan sampel baru dengan cara melakukan interpolasi linier antara sampel terpilih dan tetangganya.

Hasil interpolasi ini berupa titik data baru yang memiliki karakteristik mirip dengan data kelas minoritas asli, namun bukan duplikat langsung. Dengan demikian, distribusi data menjadi lebih seimbang dan model dapat belajar dengan lebih adil terhadap kedua kelas [42].

Penggunaan SMOTE secara umum terbukti meningkatkan performa model klasifikasi dalam kondisi data tidak seimbang. Namun, perlu diperhatikan bahwa jika tidak digunakan dengan hati-hati, SMOTE juga dapat menimbulkan risiko *overfitting*, terutama ketika sampel minoritas sangat sedikit atau nilai parameter k terlalu besar, yang dapat menghasilkan data sintetis yang kurang representatif terhadap distribusi sebenarnya [42].

2.3.15 Hyperparameter Tuning

Dalam proses pelatihan model *machine learning*, *hyperparameter* adalah parameter eksternal yang nilainya tidak dipelajari dari data, melainkan ditentukan terlebih dahulu sebelum pelatihan dimulai. Contohnya antara lain jumlah estimator (n_estimators), tingkat pembelajaran (*learning rate*), dan kedalaman maksimum pohon (*max_depth*) pada model berbasis pohon. Pemilihan kombinasi *hyperparameter* yang tepat sangat penting karena dapat mempengaruhi performa, akurasi, dan kemampuan model dalam menangani data baru secara umum (*generalization*) [44].

Tuning hyperparameter dilakukan untuk mencari konfigurasi optimal yang dapat menghasilkan model terbaik. Dua pendekatan yang umum digunakan adalah *Grid Search* dan *Random Search*. *Grid Search* menguji seluruh kombinasi parameter yang mungkin dalam ruang pencarian, sedangkan *Random Search* memilih sejumlah kombinasi secara acak. Pendekatan *random* dinilai lebih efisien, terutama ketika hanya sebagian kecil parameter yang benar-benar mempengaruhi hasil model secara signifikan [44].

Secara umum, proses *tuning* memiliki sejumlah keunggulan, seperti kemampuan meningkatkan akurasi model, mencegah *overfitting*, serta

menyesuaikan model agar lebih efisien terhadap kompleksitas data. Namun, proses ini juga memiliki kekurangan, terutama pada kebutuhan komputasi yang tinggi dan waktu pelatihan yang lama, apalagi jika ruang parameter yang diuji terlalu luas [44].

2.3.16 RandomizedSearchCV dan *Cross-Validation*

RandomizedSearchCV adalah metode *tuning* yang menggabungkan pencarian acak *hyperparameter* dengan validasi silang (*cross-validation*). Dalam metode ini, sistem akan memilih kombinasi parameter secara acak dari ruang pencarian yang telah ditentukan, kemudian mengevaluasi performanya menggunakan beberapa pembagian data yang berbeda. Hal ini membuat *RandomizedSearchCV* jauh lebih efisien dibandingkan *GridSearchCV*, terutama ketika waktu pelatihan menjadi kendala [44].

Untuk proses evaluasi, digunakan teknik *k-fold cross-validation*, yaitu metode yang membagi dataset menjadi k bagian (*folds*). Model akan dilatih sebanyak k kali, di mana pada setiap iterasi satu *fold* digunakan sebagai data validasi dan sisanya sebagai data pelatihan. Skor akhir adalah rata-rata dari semua iterasi tersebut. Metode ini memberikan estimasi performa model yang lebih stabil dan tidak bergantung pada pembagian data tunggal [44].

Penggunaan *RandomizedSearchCV* dan *cross-validation* menawarkan banyak manfaat, seperti efisiensi waktu, hasil *tuning* yang lebih cepat ditemukan, serta kemampuan menghindari *overfitting* dengan evaluasi yang menyeluruh. Meski demikian, metode ini tetap memiliki keterbatasan seperti kemungkinan tidak menemukan kombinasi terbaik jika iterasi yang dilakukan terlalu sedikit atau ruang pencarian terlalu sempit [44].

2.3.17 Evaluasi Model

Evaluasi model dalam *machine learning* berperan penting untuk menilai performa sistem dalam menyelesaikan tugas prediksi atau segmentasi sesuai dengan tujuan analisis. Proses evaluasi dilakukan dengan

menggunakan sejumlah metrik yang dapat memberikan gambaran objektif terhadap kualitas hasil model, baik untuk metode yang bersifat *unsupervised* seperti *K-Means Clustering* maupun *supervised* seperti *XGBoost*.

a. Evaluasi *K-Means Clustering*

Evaluasi hasil *clustering* bertujuan untuk mengukur kualitas pengelompokan data yang dihasilkan oleh algoritma tanpa mengacu pada label kelas yang telah diketahui sebelumnya, mengingat metode *unsupervised learning* tidak memiliki target variabel eksplisit. Evaluasi ini dilakukan secara internal dengan menggunakan sejumlah metrik yang dirancang untuk menilai konsistensi dan pemisahan antar klaster. Beberapa metrik yang umum digunakan dalam evaluasi internal meliputi *Silhouette Coefficient, Davies-Bouldin Index*, dan *Calinski-Harabasz Index*. Ketiga metrik ini memberikan perspektif berbeda terhadap struktur dan kualitas klaster yang terbentuk [45].

Silhouette Coefficient merupakan ukuran yang digunakan untuk menilai seberapa sesuai suatu titik berada di dalam klaster tempatnya dikategorikan dibandingkan dengan kedekatannya terhadap klaster lain yang paling dekat. Nilai koefisien ini berada dalam rentang -1 hingga 1, di mana semakin tinggi nilainya, semakin baik tingkat pemisahan antar klaster yang terbentuk. Rumus [45]:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Rumus 2 4 Rumus *Silhouette Coefficient*

Penjelasan Komponen Rumus 2.4:

a. a(i): jarak rata-rata dari titik i ke semua titik lain dalam klaster yang sama.
b. b(i): jarak rata-rata dari titik i ke titik-titik di klaster terdekat lainnya.
c. max{a(i),b(i)}: digunakan untuk menormalisasi nilai agar hasil tetap dalam rentang -1 sampai 1.

Semakin tinggi nilai s(i), maka semakin tepat posisi suatu titik dalam klaster tempatnya berada. Apabila nilainya mendekati nol, berarti titik tersebut berada di area perbatasan antara dua klaster. Sementara itu, nilai negatif menunjukkan kemungkinan bahwa titik tersebut terkelompok dalam klaster yang kurang sesuai [45].

Davies-Bouldin Index digunakan untuk mengevaluasi kualitas hasil *clustering* dengan menilai tingkat kemiripan antar klaster. Indeks ini menghitung rasio antara sebaran dalam *cluster* dengan jarak antara pusat klaster. Semakin kecil nilai yang dihasilkan, maka kualitas pengelompokan dianggap semakin baik. Rumus [45]:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max\left(\frac{S_i + S_j}{M_{ij}}\right)$$

Rumus 2 5 Rumus *Davies-Bouldin* Index

Penjelasan Komponen Rumus 2.5:

a. k: jumlah total klaster.
b. Si: rata-rata jarak titik-titik dalam klaster i terhadap *centroid*-nya.
c. Mij: jarak antara *centroid* klaster i dan j.

Nilai indeks yang rendah mengindikasikan bahwa setiap klaster memiliki kekompakan internal yang baik (nilai S kecil) dan jarak antar klaster cukup besar (nilai M besar) [45].

Indeks *Calinski-Harabasz* digunakan untuk menilai kualitas hasil pengelompokan data dengan mengukur rasio antara variansi antar klaster dengan variansi dalam klaster. Semakin besar nilai indeks ini, maka semakin baik kualitas pengelompokan yang dihasilkan, karena menunjukkan bahwa klaster yang terbentuk memiliki pemisahan yang jelas dan kohesi internal yang kuat. Rumus [45]:

$$CHI = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{n-k}{k-1}$$

Rumus 2 6 Rumus *Calinski-Harabasz*

Penjelasan Komponen Rumus 2.6:

a. Tr(B_k): total variansi antar klaster (*between-cluster dispersion*).
b. Tr(W_k): total variansi antar klaster (*within-cluster dispersion*).
c. n: jumlah total data.
d. k: jumlah klaster.

Rasio ini merepresentasikan tingkat keterpisahan antar klaster. Nilai CHI yang lebih tinggi mengindikasikan bahwa klaster-klaster memiliki pemisahan yang lebih jelas dan kepadatan internal yang baik [45].

b. Evaluasi *XGBoost*

Evaluasi hasil prediksi dilakukan untuk menilai sejauh mana model klasifikasi, dalam hal ini *XGBoost*, mampu membedakan antara kelas target dengan akurasi dan ketepatan yang tinggi. Berbeda dengan metode *clustering* yang bersifat *unsupervised*, pendekatan *supervised* seperti *XGBoost* memiliki target variabel yang jelas, sehingga memungkinkan pengukuran performa model berdasarkan perbandingan langsung antara label prediksi dan label aktual. Evaluasi dilakukan menggunakan metrik-metrik berbasis *confusion matrix* seperti akurasi, presisi, *recall*, dan F1-*score*, yang masing-masing memberikan sudut pandang berbeda terhadap kinerja model. Selain itu, evaluasi juga dilengkapi dengan *Receiver Operating Characteristic* (ROC) dan *Area Under the Curve* (AUC) untuk mengukur kemampuan model dalam membedakan kelas secara menyeluruh, terutama pada data yang tidak seimbang [46].

Confusion matrix adalah tabel yang menggambarkan jumlah prediksi benar dan salah yang dibuat oleh model terhadap dua kelas target, yaitu positif dan negatif. Berdasarkan hasil prediksi ini, empat kategori utama digunakan [46]:

a. *True Positive* (TP): data positif yang berhasil diprediksi positif,
b. *True Negative* (TN): data negatif yang diprediksi dengan benar sebagai negatif,
c. *False Positive* (FP): data negatif yang salah diklasifikasikan sebagai positif,
d. *False Negative* (FN): data positif yang keliru diklasifikasikan sebagai negatif.

Dari *confusion matrix* tersebut, beberapa metrik evaluasi utama dapat dihitung sebagai berikut [46]:

i. Akurasi: Mengukur seberapa sering model memberikan prediksi yang benar secara keseluruhan.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Rumus 2 7 Rumus Akurasi

ii. Presisi: Mengukur proporsi prediksi positif yang benar-benar positif.

$$Presisi = \frac{TP}{TP + FP}$$

Rumus 2 8 Rumus Presisi

iii. *Recall* (*Sensitivity*): Mengukur proporsi kasus positif yang berhasil terdeteksi.

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2 9 Rumus *Recall*

iv. F1-*Score*: Merupakan rata-rata harmonis dari presisi dan recall, berguna saat distribusi kelas tidak seimbang.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

Rumus 2 10 Rumus F1-*Score*

Evaluasi ini penting karena tidak hanya menilai akurasi keseluruhan, tetapi juga menunjukkan bagaimana model menangani ketidakseimbangan data serta seberapa baik ia mengidentifikasi kelas minoritas atau kasus penting. Selain metrik berbasis *confusion matrix*, digunakan pula metode evaluasi visual dan numerik berupa ROC *Curve* (*Receiver Operating Characteristic Curve*) dan AUC (*Area Under the Curve*). ROC *Curve* menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai ambang batas klasifikasi [46].

a. *True Positive Rate* (*Recall*):

$$TPR = \frac{TP}{TP + FN}$$

Rumus 2 11 Rumus *True Positive Rate*

b. *False Positive Rate*:

$$FPR = \frac{FP}{FP + TN}$$

Rumus 2 12 Rumus *False Positive Rate*

Untuk mengukur kinerja model secara keseluruhan terhadap seluruh ambang batas, digunakan nilai *Area Under the Curve* (AUC). Nilai AUC merepresentasikan kemungkinan bahwa model memberikan skor lebih tinggi untuk contoh positif dibandingkan contoh negatif secara acak. AUC dihitung sebagai luas di bawah kurva ROC, dan secara matematis dinyatakan sebagai:

$$AUC = \int_0^1 TPR(FPR)d(FPR)$$

Rumus 2 13 Rumus AUC

Berikut merupakan kriteria interpretasi dari nilai AUC secara umum. AUC mengukur area di bawah kurva ROC. Nilai AUC berada dalam rentang 0 hingga 1, di mana semakin mendekati

1 menunjukkan performa model yang semakin baik dalam membedakan antara kelas positif dan negatif.

Tabel 2 2 Kriteria Interpretasi Nilai AUC

Sumber: Ramadhan N, 2024 [46]

Nilai AUC	Interpretasi
90%-100%	Sangat Baik (*Excellent*)
80%-90%	Baik (*Good*)
70-80%	Cukup (*Fair*)
60-70%	Buruk (*Poor*)
< 60%	Gagal (*Failure*)

Berdasarkan interpretasi pada Tabel 2.2, model yang memiliki nilai AUC di atas 0.80 dianggap memiliki kinerja yang baik dalam membedakan kelas dan model dengan AUC di atas 0.90 dapat dikategorikan sangat andal. Oleh karena itu, nilai AUC menjadi indikator penting dalam memilih model klasifikasi terbaik dalam penelitian ini.

2.4 *Tools* dan *Software* Penelitian

2.4.1 Microsoft Excel

Microsoft Excel adalah aplikasi spreadsheet yang banyak digunakan untuk pengolahan data awal. Dalam konteks data science, Excel sering digunakan untuk data preprocessing, seperti pembersihan data, transformasi data, dan analisis deskriptif sederhana. Fitur-fitur Excel memungkinkan pengguna untuk menangani data terstruktur, mengidentifikasi dan menghapus nilai yang hilang atau tidak konsisten, serta melakukan transformasi data dasar sebelum melanjutkan ke tahap analisis yang lebih kompleks [47].

2.4.2 Python

Python merupakan bahasa pemrograman tingkat tinggi yang populer di berbagai bidang seperti pendidikan, riset, dan pengembangan perangkat lunak, berkat sintaksnya yang ringkas dan fleksibel. Bahasa ini mendukung beragam paradigma pemrograman termasuk prosedural,

berorientasi objek, dan fungsional yang menjadikannya sesuai untuk berbagai kebutuhan, mulai dari pembelajaran berbasis proyek hingga analisis data. Python banyak digunakan sebagai alat utama dalam pengajaran pemrograman karena struktur bahasanya yang mudah dipahami serta ketersediaan pustaka yang luas untuk analisis dan visualisasi data [48].

2.4.3 Visual Studio Code

Visual Studio Code (VSCode) merupakan salah satu *source code editor* yang dikembangkan oleh Microsoft dan mendapatkan popularitas luas di kalangan pengembang perangkat lunak maupun praktisi data. Editor ini bersifat gratis dan *open-source*, serta mendukung berbagai sistem operasi seperti Windows, macOS, dan Linux. Keunggulan utama VSCode terletak pada fleksibilitasnya, karena dapat disesuaikan dengan berbagai bahasa pemrograman dan kebutuhan analitik melalui penggunaan *extension* atau ekstensi tambahan [49].

Dalam konteks *data science*, VSCode menjadi alternatif kuat terhadap IDE tradisional seperti RStudio. Transisi dari RStudio ke VSCode memberikan sejumlah manfaat signifikan, terutama dalam hal integrasi lintas bahasa dan manajemen proyek yang lebih baik. Salah satu fitur penting dari VSCode adalah kemampuannya untuk mengelola berbagai lingkungan kerja dalam satu antarmuka, memungkinkan pengguna untuk bekerja dengan Python, R, SQL, dan bahkan Bash secara bersamaan tanpa perlu berpindah editor [49].

Selain itu, VSCode menyediakan pengalaman kerja yang lebih *lightweight* dengan kustomisasi tinggi. Beragam ekstensi seperti Jupyter, Python *Extension Pack*, dan R *Language Support* mempermudah pengguna dalam menulis, menjalankan, serta mendokumentasikan kode secara efisien. VSCode juga memiliki integrasi Git bawaan yang memungkinkan *version control* dilakukan langsung dari editor, sehingga meningkatkan efisiensi dalam kolaborasi proyek [49].

Pengguna yang terbiasa dengan ekosistem R dan RStudio, proses adaptasi terhadap VSCode mungkin memerlukan waktu terutama dalam memahami struktur pengelolaan proyek dan navigasi file. Namun, setelah proses tersebut dilewati, VSCode memberikan fleksibilitas dan efisiensi yang lebih luas, khususnya dalam proyek yang melibatkan beragam bahasa pemrograman atau kerangka kerja (*framework*) [49].