BAB 2
TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Berbagai penelitian sebelumnya telah mengkaji pengaruh *preprocessing* terhadap kinerja model klasifikasi penyakit jantung, namun sebagian besar masih terbatas pada penerapan teknik tertentu secara terpisah.

Tabel 2.1. Penelitian Terdahulu

No	Peneliti	Fokus Penelitian	*Preprocessing*	Algoritma
1	Baihaqi et al. (2023)	Imputasi data	Imputasi, *Normalisasi*	*Linear Regression*
2	Rahim et al. (2023)	*Normalisasi* dan *Balancing*	*Normalisasi*, *SMOTE*	*Random Forest*
3	Abror (2023)	Seleksi fitur	*Information Gain*	*SVM*
4	Afghoni et al. (2025)	Penyeimbangan data	*SMOTE*	*Decision Tree*

Tabel 2.1. merangkum penelitian-penelitian terdahulu beserta teknik *preprocessing* yang diterapkan. Peneliti pertama meneliti efektivitas metode imputasi *mean* dan menemukan bahwa metode ini mampu memperbaiki kualitas data dengan hasil yang lebih stabil dibandingkan tanpa imputasi data [7]. Peneliti kedua mengkaji pengaruh *Normalisasi* dan *balancing* menggunakan *SMOTE*, dan menunjukkan bahwa kombinasi keduanya meningkatkan akurasi model hingga 2% [8]. Peneliti ketiga mengevaluasi penerapan teknik *information Gain* dalam seleksi fitur dan menemukan bahwa metode ini mampu meningkatkan akurasi dan presisi

algoritma svm [9]. Peneliti keempat meneliti penggunaan *SMOTE* untuk mengatasi ketidakseimbangan kelas dan melaporkan peningkatan akurasi signifikan hingga 4% setelah dilakukan *balancing* [10]. Keempat penelitian tersebut menunjukkan bahwa tahapan *preprocessing* dapat memberikan dampak signifikan terhadap performa model, namun masing-masing masih terbatas pada teknik tertentu secara terpisah tanpa menguji kombinasi *preprocessing* yang lebih komprehensif. Dengan demikian, dapat disimpulkan bahwa masih terdapat celah penelitian untuk mengevaluasi kombinasi tahapan *preprocessing* dalam satu eksperimen terpadu. Penelitian ini hadir untuk mengisi kekosongan tersebut melalui pendekatan yang terstruktur berbasis metodologi *CRISP-DM*, serta dengan memanfaatkan algoritma KNN sebagai model klasifikasi.

2.2. Landasan Teori

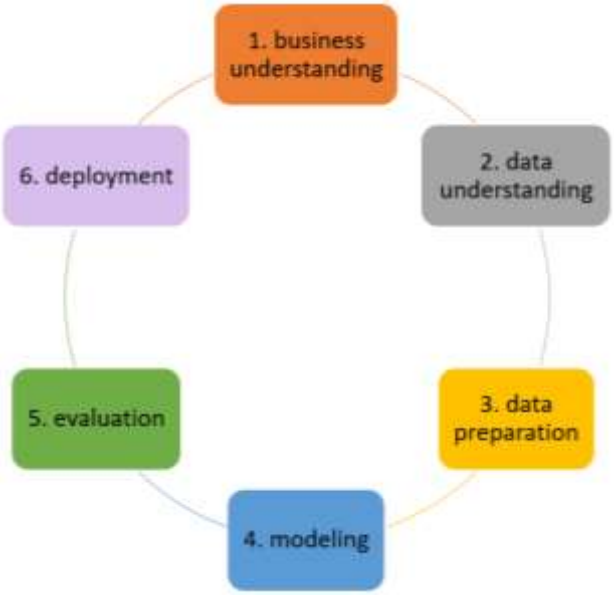
2.2.1. *Data mining* dalam Bidang Kesehatan

Pemanfaatan data dalam jumlah besar telah menjadi bagian penting dalam upaya peningkatan layanan kesehatan, terutama melalui pendekatan analitis seperti *data mining*. Secara umum, *data mining* dapat didefinisikan sebagai proses eksplorasi pengetahuan yang tersembunyi dalam kumpulan data besar dengan tujuan menemukan pola yang bermanfaat untuk pengambilan keputusan. Dalam bidang kesehatan, teknologi ini banyak digunakan dalam berbagai aplikasi, seperti membantu proses diagnosis, mengklasifikasikan kondisi pasien, serta mendukung keputusan klinis berbasis data [11]. Salah satu keunggulan utama dari penerapan klasifikasi dalam *data mining* adalah kemampuannya untuk memprediksi kondisi kesehatan pasien secara dini melalui pola yang ditemukan dari data historis.

2.2.2. Metodologi *CRISP-DM*

Dalam praktik *data mining*, dibutuhkan kerangka kerja metodologi yang sistematis agar proses analisis berjalan terarah. Salah satu metodologi yang paling banyak digunakan secara luas adalah *CRISP-DM* (*Cross Industry Standard Process for Data mining*), yang menawarkan tahapan komprehensif mulai dari pemahaman bisnis hingga implementasi hasil. *CRISP-DM* terdiri atas enam tahap utama, yaitu: *business understanding, data understanding, Data Preparation, Modeling,*

Evaluation, dan *Deployment* [12]. Keunggulan dari metodologi ini terletak pada fleksibilitasnya, yang memungkinkan penerapan di berbagai bidang industri, termasuk sektor kesehatan. Selain itu, pendekatan tahapan yang iteratif membantu peneliti untuk menyesuaikan alur kerja berdasarkan kebutuhan dan dinamika data. Namun demikian, *CRISP-DM* memiliki kekurangan, antara lain tidak memberikan panduan teknis yang spesifik terkait pemilihan algoritma atau teknik *preprocessing* yang optimal. Selain itu, tahap *Deployment* dalam *CRISP-DM* cenderung kurang rinci, terutama dalam konteks implementasi berbasis sistem modern seperti *machine learning* berbasis cloud atau integrasi API.



Gambar 2.1. Metodologi *CRISP-DM*

Gambar 2.1. merupakan alur metodologi *CRISP-DM* yang terdiri dari enam tahapan utama yang bersifat iteratif dan fleksibel. Tahapan pertama *adalah business understanding*, yaitu memahami secara mendalam tujuan bisnis atau permasalahan yang ingin diselesaikan melalui *data mining*. Setelah itu, tahap *data understanding* dilakukan untuk mengenali karakteristik data yang tersedia, seperti tipe atribut, kualitas data, serta potensi masalah seperti *missing value* atau *outlier*. Tahap berikutnya adalah *Data Preparation*, yang mencakup berbagai proses pengolahan data seperti pembersihan, transformasi, *Encoding*, *Normalisasi*, *balancing*, dan seleksi *fitur*, agar data siap digunakan untuk *Modeling*. Pada tahap *Modeling,*

algoritma pembelajaran mesin seperti *K-Nearest Neighbors* diterapkan untuk membangun model klasifikasi berdasarkan data yang telah dipersiapkan. Selanjutnya, tahap *Evaluation* dilakukan untuk menilai performa model menggunakan metrik evaluasi seperti akurasi, *Precision, Recall*, dan *F1-score*, serta memastikan bahwa hasil model relevan dengan tujuan awal. Terakhir, tahap *Deployment* berfokus pada implementasi hasil analisis, baik dalam bentuk dokumentasi, rekomendasi, maupun integrasi ke dalam sistem operasional yang dapat digunakan oleh pengguna akhir. Keseluruhan proses ini bersifat siklus, artinya dapat kembali ke tahap sebelumnya bila diperlukan untuk mencapai hasil yang optimal.

2.3.3. *Preprocessing Data*

Preprocessing data berada didalam tahapan *Data Preparation*, yang bertujuan untuk mempersiapkan data mentah agar layak dan optimal sebelum dimanfaatkan untuk membangun model prediksi. *Preprocessing* adalah serangkaian tahapan yang bertujuan untuk membersihkan, menyatukan, mengubah, dan mereduksi data agar siap digunakan dalam proses *Modeling*.

Secara umum, *preprocessing* terbagi menjadi empat tahap utama yaitu *Cleaning* / pembersihan data bertujuan memperbaiki data yang tidak lengkap atau tidak konsisten seperti nilai kosong atau ekstrim [13]. Data *Transformation* / transformasi mengubah format data agar sesuai untuk analisis, misalnya melalui *Encoding*, *Normalisasi* atau teknik *balancing* seperti *SMOTE* [14]. *Data integration* menggabungkan data dari berbagai sumber agar menjadi satu kesatuan yang konsisten, meskipun tidak selalu diperlukan jika data berasal dari satu sumber. Terakhir, *data Reduction* digunakan untuk menyederhanakan jumlah fitur dengan teknik seperti seleksi atribut menggunakan *information Gain* guna meningkatkan efisiensi tanpa kehilangan informasi penting [15]. Keempat tahapan ini saling mendukung dalam meningkatkan kualitas data untuk proses klasifikasi yang lebih akurat.

Fungsi utama dari *preprocessing* adalah untuk meningkatkan kualitas data dan mengurangi gangguan dari *noise* atau ketidakkonsistenan, sehingga model yang

dibangun dapat memberikan hasil yang lebih akurat dan dapat diandalkan. Keunggulan dari *preprocessing* adalah kemampuannya dalam meningkatkan performa model secara signifikan, mengurangi risiko *Overfitting*, serta memastikan bahwa data dalam format yang sesuai dengan algoritma *machine learning* yang digunakan. Namun, *preprocessing* juga memiliki kekurangan, seperti waktu pemrosesan yang cenderung lama, terutama pada *Dataset* besar, serta potensi hilangnya informasi penting jika tidak dilakukan dengan hati-hati. Selain itu, pemilihan teknik *preprocessing* yang kurang tepat dapat menyebabkan bias atau menurunnya kinerja model. Oleh karena itu, pemahaman yang mendalam terhadap karakteristik data dan tujuan analisis sangat diperlukan agar *preprocessing* dapat dilakukan secara efektif dan efisien.

2.3.4. *K-Nearest Neighbors (KNN)*

Salah satu algoritma klasifikasi yang cukup populer dalam penelitian *data mining* adalah *K-Nearest Neighbors (KNN)*. Algoritma ini bersifat *instance-based* dan bekerja berdasarkan prinsip kemiripan jarak antara data uji dan data latih. Secara umum, KNN menentukan kelas suatu instance baru berdasarkan mayoritas kelas dari K tetangga terdekatnya dalam ruang fitur. Karena bersifat non-parametrik, algoritma ini tidak memerlukan pelatihan model secara eksplisit dan hanya bergantung pada kedekatan data. Namun demikian, performa KNN sangat sensitif terhadap skala fitur yang digunakan, sehingga *preprocessing* seperti *Normalisasi* menjadi sangat penting untuk mencegah bias terhadap fitur dengan rentang nilai yang besar [16]. Pemilihan algoritma KNN didasarkan pada sifatnya yang sederhana, namun efektif untuk *Dataset* dengan ukuran sedang dan fitur numerik maupun kategorikal setelah dilakukan *Encoding*. Kelebihan KNN terletak pada kemudahan implementasi dan fleksibilitasnya dalam menangani data dengan distribusi yang tidak diketahui, tanpa perlu asumsi khusus terhadap struktur data. Namun di sisi lain, algoritma ini memiliki kekurangan yaitu rentan terhadap *noise* dan data outlier, serta memiliki biaya komputasi yang cukup tinggi saat proses prediksi, terutama pada *Dataset* berukuran besar, karena seluruh data latih harus dihitung jaraknya setiap kali klasifikasi dilakukan.