BAB 2
LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Penyakit Jantung

Penyakit jantung termasuk dalam kelompok penyakit kardiovaskular yang terjadi akibat kelainan pada struktur dan fungsi jantung. Jenis-jenisnya meliputi penyakit jantung koroner, gagal jantung, gangguan irama jantung (aritmia), hingga serangan jantung. Penyebab utama kondisi ini umumnya berkaitan dengan adanya penyempitan atau sumbatan pada pembuluh darah koroner yang mengurangi suplai darah ke jaringan otot jantung [9].

Risiko terjadinya penyakit jantung dipengaruhi oleh dua kategori faktor, yaitu faktor yang bersifat tetap dan faktor yang dapat diubah. Faktor tetap meliputi usia, jenis kelamin, dan riwayat keluarga, sedangkan faktor yang dapat dikendalikan antara lain tekanan darah tinggi, kadar kolesterol, diabetes, obesitas, kebiasaan merokok, serta kurangnya aktivitas fisik. Kombinasi dari faktor-faktor tersebut berkontribusi dalam meningkatkan risiko terjadinya penyakit jantung [10].

2.1.2 Klasifikasi

Klasifikasi adalah salah satu teknik dalam *supervised learning* yang bertujuan untuk mengelompokkan data ke dalam kategori tertentu berdasarkan karakteristik yang dimiliki. Pada kasus klasifikasi biner, data diklasifikasikan ke dalam dua kelompok, misalnya kelas positif dan negatif. Secara umum, proses ini terdiri dari dua tahap, yaitu tahap pelatihan (*training*) untuk membangun model dari data latih, serta tahap pengujian (*testing*) untuk mengevaluasi kemampuan model menggunakan data baru yang belum pernah dikenali sebelumnya [11].

$$f : X \rightarrow Y \tag{2.1}$$

dengan $X = \{x_1, x_2, \ldots, x_n\}$ sebagai fitur dan $Y \in \{0, 1\}$ sebagai label kelas.

Pada kasus penyakit jantung, metode klasifikasi dimanfaatkan untuk menentukan status pasien, yaitu apakah termasuk dalam kelompok penderita penyakit jantung (1) atau bukan (0), dengan mengacu pada data klinis yang dimiliki.

2.1.3 Logistic Regression

Logistic Regression adalah metode klasifikasi yang bertujuan untuk memprediksi kemungkinan suatu peristiwa terjadi dengan keluaran dalam bentuk dua kategori, yaitu positif dan negatif. Prosesnya menggunakan fungsi logistik (sigmoid) untuk mengubah hasil kombinasi linier dari variabel masukan menjadi nilai probabilitas antara 0 dan 1, yang selanjutnya dijadikan acuan dalam pengambilan keputusan klasifikasi [12].

Logistic Regression dikenal memiliki keunggulan dari segi kesederhanaan model dan kemudahan interpretasi. Nilai koefisien yang dihasilkan dapat menunjukkan kontribusi masing-masing variabel independen terhadap kemungkinan terjadinya suatu kejadian, seperti penyakit jantung. Hal ini membuat metode tersebut banyak digunakan dalam analisis di bidang medis. Meskipun demikian, metode ini kurang optimal dalam menangani hubungan non-linear dan interaksi fitur yang kompleks, sehingga kinerjanya dapat berkurang ketika data tidak mengikuti pola linier [13].

Dalam pendekatan matematis, *Logistic Regression* menggambarkan keterkaitan antara variabel masukan dan probabilitas kelas target dengan menggunakan fungsi logistik (sigmoid), yang dinyatakan dalam persamaan berikut:

$$P(Y = 1|X) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

di mana:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{2.3}$$

di mana:

- $P(Y = 1|X)$ adalah probabilitas data termasuk ke dalam kelas positif (penyakit jantung),
- e adalah bilangan eksponensial (sekitar 2,718),
- β_0 adalah *intercept* atau bias model,

- β_i adalah koefisien regresi untuk variabel ke-i,
- x_i adalah nilai dari variabel input ke-i,
- n adalah jumlah variabel independen.

Probabilitas yang dihasilkan kemudian dievaluasi menggunakan nilai ambang (*threshold*), yang biasanya ditentukan sebesar 0,5. Data akan dimasukkan ke dalam kelas positif jika nilai probabilitasnya sama dengan atau melebihi ambang tersebut, sedangkan jika berada di bawahnya, data akan diklasifikasikan sebagai kelas negatif [14].

2.1.4 Random Forest

Random Forest adalah algoritma berbasis *ensemble* yang memanfaatkan kumpulan pohon keputusan (*decision tree*) untuk menghasilkan prediksi yang lebih akurat dan stabil. Setiap pohon dibangun secara terpisah menggunakan metode *bootstrap sampling* serta pemilihan subset fitur secara acak pada setiap percabangan node. Strategi ini digunakan untuk mengurangi kemungkinan *overfitting* yang umumnya terjadi pada satu pohon keputusan [15].

Pada tahap awal, *Random Forest* menerapkan teknik *bootstrap sampling* untuk menghasilkan beberapa subset data dari dataset latih melalui pengambilan sampel acak dengan pengembalian. Setiap subset tersebut selanjutnya digunakan untuk membentuk sebuah pohon keputusan. Dalam proses pembentukannya, data dipisahkan berdasarkan atribut yang paling optimal dengan menggunakan kriteria tertentu, seperti *Gini Index* atau *Entropy* [16].

Dalam formulasi matematis, *Gini Index* digunakan sebagai ukuran tingkat ketidakmurnian suatu node, yang dapat dirumuskan sebagai berikut:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{2.4}$$

di mana:

- S adalah himpunan data pada suatu *node*,
- c adalah jumlah kelas,
- p_i adalah proporsi data yang termasuk ke dalam kelas ke-i.

Selain *Gini Index*, *Random Forest* juga dapat menggunakan *Entropy* sebagai kriteria pemisahan node, yang dirumuskan sebagai berikut:

$$Entropy(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2.5}$$

di mana:

- S adalah himpunan data pada suatu node,
- c adalah jumlah kelas,
- p_i adalah proporsi data pada kelas ke-i.

Pada tahap pembentukan node, ditentukan nilai ambang (*threshold*) yang berfungsi untuk membagi data menjadi dua *subset*, yakni cabang kiri dan cabang kanan. Proses pembagian data berdasarkan ambang ini dapat dirumuskan secara matematis sebagai berikut:

$$x_j \leq t \quad \text{atau} \quad x_j > t \tag{2.6}$$

di mana:

- x_j adalah nilai atribut ke-j,
- t adalah nilai ambang (*threshold*) yang digunakan sebagai pemisahan data.

Tahap pengambilan dataset dan pembentukan pohon keputusan diulang sebanyak n kali sehingga dihasilkan sejumlah pohon keputusan yang saling independen. Setelah seluruh pohon terbentuk, setiap data uji akan diklasifikasikan oleh seluruh pohon, sehingga diperoleh sejumlah hasil prediksi.

Hasil prediksi akhir pada *Random Forest* diperoleh dengan menerapkan metode *majority voting*, yang secara matematis dinyatakan sebagai berikut:

$$\hat{y} = \text{mode}\left(h_1(x), h_2(x), \ldots, h_n(x)\right) \tag{2.7}$$

di mana:

- $\hat{y}$ adalah hasil prediksi akhir,
- $h_i(x)$ adalah hasil prediksi dari pohon keputusan ke-i,
- n adalah jumlah pohon keputusan dalam *Random Forest*.

2.1.5 Ketidakseimbangan Data

Imbalanced data adalah kondisi ketika distribusi jumlah data antar kelas dalam sebuah dataset tidak seimbang, di mana salah satu kelas memiliki jumlah data yang jauh lebih besar dibandingkan kelas lainnya. Kondisi ini kerap terjadi dalam kasus klasifikasi di bidang medis, karena jumlah pasien yang sehat biasanya lebih dominan dibandingkan pasien yang memiliki penyakit tertentu [17].

Ketidakseimbangan distribusi kelas dapat menyebabkan masalah serius dalam proses klasifikasi. Model yang dibangun dari dataset dengan komposisi kelas yang tidak seimbang umumnya lebih akurat dalam memprediksi kelas mayoritas, namun kurang mampu mengenali kelas minoritas. Hal ini menjadi krusial, terutama pada kasus seperti deteksi penyakit, di mana kelas minoritas justru memiliki peran penting karena berhubungan dengan kondisi kesehatan yang berisiko tinggi [18].

Pada kasus penyakit jantung, ketidakseimbangan data tercermin dari jumlah data pasien sehat yang jauh lebih besar dibandingkan pasien yang menderita penyakit jantung. Situasi ini dapat menyebabkan model klasifikasi cenderung bias terhadap kelas mayoritas, sehingga kemampuan model dalam mendeteksi kasus positif menjadi kurang optimal.

Untuk mengukur tingkat ketidakseimbangan data, digunakan rasio *imbalance* yang secara matematis dapat dirumuskan sebagai berikut:

$$IR = \frac{N_{majority}}{N_{minority}} \tag{2.8}$$

di mana:

- IR adalah rasio ketidakseimbangan (*Imbalance Ratio*),
- $N_{majority}$ adalah jumlah data pada kelas mayoritas,
- $N_{minority}$ adalah jumlah data pada kelas minoritas.

Nilai rasio *imbalance* yang semakin besar menunjukkan tingkat ketidakseimbangan data yang semakin tinggi pada suatu dataset. Dengan demikian, dibutuhkan metode penanganan yang tepat untuk mengatasi permasalahan tersebut, sehingga model klasifikasi dapat memahami pola dari masing-masing kelas secara lebih proporsional.

2.1.6 K-Means SMOTE

K-Means SMOTE adalah metode *oversampling* yang mengintegrasikan algoritma *K-Means* dengan teknik SMOTE untuk mengatasi ketidakseimbangan data secara lebih efektif. Metode ini menghasilkan data sintetis dengan mempertimbangkan pola klaster dan kepadatan distribusi data minoritas, sehingga proses pembentukan data baru tidak dilakukan secara seragam pada seluruh ruang fitur [19].

Metode *K-Means* SMOTE memiliki tiga tahapan utama, yaitu *clustering*, *filtering*, dan *oversampling*. Pada tahap *clustering*, data dibagi ke dalam sejumlah klaster dengan menggunakan algoritma *K-Means* guna memahami pola distribusi dan kepadatan data. Proses ini membantu mengidentifikasi area yang padat maupun jarang dalam ruang fitur, khususnya pada kelas minoritas [20].

Tahap selanjutnya adalah *filtering*, yaitu pemilihan klaster yang layak untuk dilakukan oversampling. Klaster yang memiliki proporsi kelas minoritas yang signifikan akan diprioritaskan, sedangkan klaster yang didominasi oleh kelas mayoritas atau memiliki kepadatan minoritas yang sangat tinggi dapat dikecualikan. Pada tahap ini juga ditentukan alokasi jumlah sampel sintetis untuk setiap klaster, dimana klaster dengan kepadatan data minoritas yang rendah akan memperoleh alokasi sampel sintetis yang lebih besar dibandingkan klaster yang padat.

Untuk menentukan alokasi tersebut, terlebih dahulu dihitung kepadatan (*density*) data minoritas pada setiap klaster. Kepadatan data minoritas pada klaster ke- k dirumuskan sebagai berikut:

$$D_k = \frac{N^k_{minority}}{V_k} \tag{2.9}$$

di mana:

- D_k adalah kepadatan data minoritas pada klaster ke-k,

- $N^k_{minority}$ adalah jumlah data minoritas dalam klaster ke-k,
- V_k adalah volume atau ukuran klaster ke-k.

Berdasarkan nilai kepadatan tersebut, bobot *sampling* untuk setiap klaster dihitung menggunakan *invers* dari kepadatan data minoritas, sehingga klaster dengan kepadatan yang lebih rendah akan memperoleh bobot yang lebih besar. Bobot *sampling* dirumuskan sebagai berikut:

$$w_k = \frac{1/D_k}{\sum_{j=1}^{K}(1/D_j)} \tag{2.10}$$

di mana:

- w_k adalah bobot sampling klaster ke-k,
- D_k adalah kepadatan data minoritas pada klaster ke-k,
- K adalah jumlah klaster yang terpilih.

Jumlah sampel sintetis yang akan dihasilkan pada setiap klaster kemudian ditentukan berdasarkan bobot sampling tersebut, yang dirumuskan sebagai berikut:

$$N^k_{synthetic} = w_k \times N_{synthetic} \tag{2.11}$$

di mana:

- $N^k_{synthetic}$ adalah jumlah sampel sintetis yang dihasilkan pada klaster ke-k,
- w_k adalah bobot *sampling* klaster ke-k,
- $N_{synthetic}$ adalah total jumlah sampel sintetis yang akan dihasilkan.

Pada tahap terakhir, yaitu *oversampling*, teknik SMOTE diaplikasikan pada masing-masing klaster terpilih berdasarkan jumlah sampel sintetis yang telah ditetapkan. Pendekatan ini memungkinkan *K-Means SMOTE* menghasilkan distribusi data baru yang lebih sesuai dengan karakteristik data, meminimalkan pembentukan sampel pada area yang tidak relevan, serta meningkatkan kinerja model klasifikasi pada data yang mengalami ketidakseimbangan [21].

2.1.7 Borderline SMOTE

Borderline SMOTE adalah variasi dari metode SMOTE yang dikembangkan untuk menangani masalah *overlapping* antara kelas minoritas dan mayoritas. Permasalahan ini sering muncul ketika posisi data minoritas berada berdekatan dengan data mayoritas dalam ruang fitur, yang dapat menyebabkan meningkatnya kesalahan klasifikasi [22].

Tidak seperti SMOTE standar yang melakukan pembangkitan data sintetis secara menyeluruh pada kelas minoritas, *Borderline SMOTE* lebih berfokus pada sampel minoritas yang terletak di sekitar *decision boundary*. Data di area ini cenderung lebih rawan mengalami kesalahan klasifikasi dan memiliki kontribusi besar dalam menentukan batas pemisahan antar kelas.

Tahapan awal dalam *Borderline SMOTE* adalah menentukan m tetangga terdekat (*nearest neighbors*) untuk setiap sampel pada kelas minoritas. Setelah itu, dilakukan evaluasi terhadap distribusi kelas dari tetangga tersebut. Apabila mayoritas tetangga terdekat berasal dari kelas mayoritas, maka sampel minoritas tersebut diklasifikasikan sebagai *borderline instance* dan digunakan sebagai fokus utama dalam proses pembentukan data sintetis [23].

Secara matematis, suatu data minoritas xi diklasifikasikan sebagai *borderline* apabila memenuhi kondisi berikut:

$$\frac{M_i}{m} > 0.5 \tag{2.12}$$

di mana:

- M_i adalah jumlah tetangga terdekat dari kelas mayoritas untuk data x_i,
- m adalah jumlah total tetangga terdekat yang digunakan.

Setelah data minoritas yang berada pada area *borderline* teridentifikasi, pembentukan data sintetis dilakukan dengan mekanisme interpolasi linier yang serupa dengan metode SMOTE. Persamaan pembentukan data sintetis pada *Borderline SMOTE* dirumuskan sebagai berikut:

$$x_{synthetic} = x_i + \lambda \times (x_{nn} - x_i) \tag{2.13}$$

di mana:

- $x_{synthetic}$ adalah data sintetis yang dihasilkan,
- x_i adalah data minoritas yang berada pada area *borderline*,
- x_{nn} adalah salah satu tetangga terdekat dari kelas minoritas,
- λ adalah bilangan acak dalam interval $[0, 1]$.

Melalui pemusatan proses *oversampling* pada sampel minoritas yang berada di sekitar *decision boundary*, *Borderline SMOTE* dapat memperkuat representasi kelas minoritas pada area penting. Pendekatan ini membantu meningkatkan kemampuan model dalam mengklasifikasikan dan membedakan antara kelas mayoritas dan minoritas dengan lebih tepat [24].

2.1.8 Confusion Matrix

Confusion matrix adalah metode evaluasi yang digunakan untuk menilai performa model klasifikasi dengan cara membandingkan hasil prediksi dengan label aktual. Hasil evaluasi ini disajikan dalam bentuk tabel yang memudahkan analisis terhadap tingkat akurasi serta kesalahan prediksi yang dihasilkan oleh model [25].

Dalam klasifikasi biner, *confusion matrix* berbentuk matriks 2×2 yang terdiri dari empat elemen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen tersebut memberikan informasi detail mengenai hasil klasifikasi serta memungkinkan evaluasi model secara lebih mendalam [26].

True Positive (TP) merupakan jumlah data positif yang diprediksi dengan benar, sedangkan *True Negative* (TN) adalah jumlah data negatif yang juga berhasil diklasifikasikan dengan tepat. Sebaliknya, *False Positive* (FP) menunjukkan kesalahan ketika data negatif diprediksi sebagai positif, dan *False Negative* (FN) menunjukkan kesalahan ketika data positif diprediksi sebagai negatif. Informasi ini sangat penting dalam mengevaluasi kelemahan model, khususnya pada kasus klasifikasi dengan tingkat risiko yang tinggi.

Secara matematis, *confusion matrix* untuk klasifikasi biner dapat dituliskan dalam bentuk matriks sebagai berikut:

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{2.14}$$

di mana:

- TP (*True Positive*) adalah jumlah data positif yang diprediksi benar,
- TN (*True Negative*) adalah jumlah data negatif yang diprediksi benar,
- FP (*False Positive*) adalah jumlah data negatif yang salah diprediksi sebagai positif,
- FN (*False Negative*) adalah jumlah data positif yang salah diprediksi sebagai negatif.

2.1.9 Precision

Precision adalah metrik evaluasi yang digunakan untuk menilai seberapa akurat model dalam memprediksi kelas positif. Metrik ini menggambarkan proporsi prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dihasilkan oleh model. Dengan kata lain, *precision* berfokus pada kemampuan model dalam mengurangi kesalahan prediksi positif [27].

Nilai *precision* dihitung dengan membandingkan jumlah *True Positive* terhadap total prediksi positif, yaitu kombinasi antara *True Positive* dan *False Positive*. Semakin tinggi nilai *precision*, semakin besar tingkat keakuratan model dalam menghasilkan prediksi positif yang benar.

Secara matematis, *precision* dirumuskan sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \tag{2.15}$$

di mana:

- TP (*True Positive*) adalah jumlah data positif yang diprediksi dengan benar,
- FP (*False Positive*) adalah jumlah data negatif yang salah diprediksi sebagai positif.

2.1.10 Recall

Recall adalah metrik evaluasi yang digunakan untuk mengukur tingkat keberhasilan model dalam menemukan data yang benar-benar termasuk dalam kelas positif. Metrik ini menggambarkan perbandingan antara jumlah prediksi positif yang benar dengan total data positif yang sebenarnya. Dengan kata lain, *recall* menekankan pada kemampuan model dalam mengurangi kesalahan akibat tidak terdeteksinya data positif [28].

Nilai *recall* diperoleh dari perbandingan antara *True Positive* dan total data positif aktual, yaitu *True Positive* dan *False Negative*. Semakin tinggi nilai *recall*, semakin baik kemampuan model dalam mengidentifikasi data pada kelas positif.

Secara matematis, *recall* dirumuskan sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \tag{2.16}$$

di mana:

- TP (*True Positive*) adalah jumlah data positif yang diprediksi dengan benar,
- FN (*False Negative*) adalah jumlah data positif yang salah diprediksi sebagai negatif.

2.1.11 F1-Score

F1-Score merupakan metrik evaluasi yang mengintegrasikan *precision* dan *recall* ke dalam satu nilai untuk memberikan penilaian yang lebih seimbang terhadap kinerja model. Nilai ini diperoleh melalui rata-rata harmonis dari *precision* dan *recall*, sehingga mampu menggambarkan kompromi antara akurasi prediksi positif dan kemampuan model dalam menemukan seluruh data positif [29].

Metrik ini sangat penting digunakan pada dataset dengan distribusi kelas yang tidak merata, karena mempertimbangkan baik prediksi yang benar maupun kesalahan yang terjadi. Oleh sebab itu, *F1-Score* sering digunakan sebagai indikator performa yang lebih representatif dalam evaluasi model klasifikasi.

Secara matematis, *F1-Score* dirumuskan sebagai berikut:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.17}$$

di mana:

- *Precision* adalah nilai ketepatan prediksi kelas positif,
- *Recall* adalah kemampuan model dalam mendeteksi seluruh data kelas positif.

Selain *F1-Score* standar, terdapat beberapa varian yang sering digunakan dalam evaluasi model klasifikasi multikelas, yaitu *F1-Weighted* dan *F1-Macro*.

F1-Weighted adalah metrik *F1-Score* yang memperhitungkan proporsi jumlah data pada setiap kelas. Dalam metode ini, kontribusi masing-masing kelas terhadap nilai akhir disesuaikan dengan banyaknya data yang dimiliki, sehingga kelas mayoritas akan memiliki pengaruh yang lebih dominan. Pendekatan ini memungkinkan evaluasi yang lebih representatif terhadap kondisi distribusi data yang tidak seimbang.

Secara matematis, *F1-Weighted* dirumuskan sebagai berikut:

$$F1_{weighted} = \sum_{i=1}^{C} \frac{n_i}{N} \times F1_i \tag{2.18}$$

di mana:

- $F1_i$ adalah nilai *F1-Score* untuk kelas ke-i,
- n_i adalah jumlah data pada kelas ke-i,
- N adalah jumlah total data,
- C adalah jumlah kelas.

Di sisi lain, *F1-Macro* adalah metrik yang diperoleh dengan menghitung rata-rata *F1-Score* dari seluruh kelas tanpa mempertimbangkan proporsi jumlah data pada masing-masing kelas. Pendekatan ini memberikan perlakuan yang setara bagi setiap kelas, sehingga mampu mencerminkan kinerja model secara lebih adil, terutama dalam kondisi ketidakseimbangan data.

Secara matematis, *F1-Macro* dirumuskan sebagai berikut:

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{C} F1_i \tag{2.19}$$

di mana:

- $F1_i$ adalah nilai *F1-Score* untuk kelas ke-i,
- C adalah jumlah kelas.

2.1.12 Accuracy

Accuracy adalah metrik evaluasi yang digunakan untuk mengukur proporsi prediksi yang benar terhadap keseluruhan data yang diuji. Metrik ini mempertimbangkan jumlah *True Positive* dan *True Negative* sebagai prediksi yang tepat dibandingkan dengan total data.

Meskipun *accuracy* mampu memberikan gambaran umum mengenai kinerja model, penggunaannya pada dataset yang tidak seimbang perlu disertai dengan metrik lain seperti *precision*, *recall*, dan *F1-Score*. Hal ini penting agar evaluasi model tidak bias terhadap kelas mayoritas dan tetap mampu mencerminkan performa pada seluruh kelas secara lebih akurat [30].

Secara matematis, *accuracy* dirumuskan sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.20}$$

di mana:

- TP (*True Positive*) adalah jumlah data positif yang diprediksi dengan benar,
- TN (*True Negative*) adalah jumlah data negatif yang diprediksi dengan benar,
- FP (*False Positive*) adalah jumlah data negatif yang salah diprediksi sebagai positif,
- FN (*False Negative*) adalah jumlah data positif yang salah diprediksi sebagai negatif.

2.1.13 ROC dan AUC

ROC *Curve* (*Receiver Operating Characteristic*) adalah teknik evaluasi visual yang digunakan untuk mengukur kemampuan model klasifikasi biner dalam membedakan kelas positif dan negatif. Kurva ini diperoleh dengan memplot nilai *True Positive Rate* (TPR) terhadap *False Positive Rate* (FPR) pada berbagai nilai ambang (*threshold*).

TPR menggambarkan tingkat keberhasilan model dalam mendeteksi data positif, sedangkan FPR menunjukkan tingkat kesalahan model dalam mengklasifikasikan data negatif sebagai positif. Perubahan nilai *threshold* akan menghasilkan kombinasi nilai TPR dan FPR yang membentuk kurva ROC. Model dengan performa yang baik akan menghasilkan kurva yang mendekati titik kiri atas pada grafik [31].

Secara matematis, *True Positive Rate* dan *False Positive Rate* dirumuskan sebagai berikut:

$$TPR = \frac{TP}{TP + FN} \tag{2.21}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.22}$$

di mana:

- TP (*True Positive*) adalah jumlah data positif yang diprediksi dengan benar,
- FN (*False Negative*) adalah jumlah data positif yang salah diprediksi sebagai negatif,
- FP (*False Positive*) adalah jumlah data negatif yang salah diprediksi sebagai positif,
- TN (*True Negative*) adalah jumlah data negatif yang diprediksi dengan benar.

AUC (*Area Under the Curve*) adalah metrik evaluasi yang digunakan untuk mengukur luas area di bawah kurva ROC sebagai indikator performa model klasifikasi. Nilai ini menggambarkan kemampuan model dalam membedakan

kelas positif dan negatif secara menyeluruh tanpa bergantung pada nilai *threshold* tertentu.

Rentang nilai AUC adalah antara 0 hingga 1. Semakin tinggi nilai AUC, semakin baik kemampuan model dalam melakukan klasifikasi. Nilai sebesar 0,5 menunjukkan bahwa model tidak memiliki kemampuan diskriminatif dan hanya setara dengan tebakan acak, sedangkan nilai di bawah 0,5 menunjukkan bahwa model memiliki performa yang kurang baik.

Secara konseptual, AUC dapat dinyatakan sebagai integral dari kurva ROC, yaitu:

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \tag{2.23}$$

di mana:

- TPR (*True Positive Rate*) adalah tingkat prediksi positif yang benar,
- FPR (*False Positive Rate*) adalah tingkat kesalahan prediksi positif,
- $TPR(FPR)$ menunjukkan hubungan antara TPR terhadap FPR pada berbagai threshold.

2.2 Penelitian Terdahulu

Tabel 2.1. Ringkasan Penelitian Terkait

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
1	Fachriz K	*Application of Random Forest for Heart Disease Classification with SMOTE Approach to Balance Data*, 2024	Ketidak-seimbang-an data penyakit jantung	SMOTE + *Random Forest*	Meningkat-kan performa klasifikasi dengan data seimbang	Akurasi 94,7%, AUC 0,983
2	Islami-yatul Adde-wiyah, Zaehol Fatah	Penerapan Algoritma *Random Forest* dan Teknik SMOTE untuk Prediksi Kematian Akibat Gagal Jantung Menggunakan *RapidMiner*, 2025	Data gagal jantung bersifat *im-balanced*	SMOTE + *Random Forest*	Evaluasi lengkap metrik pada data medis	Akurasi 84,60%, AUC 0,916

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
3	Abdul Mizwar A. Rahim dkk	Klasifikasi Penyakit Jantung Menggunakan Metode *Synthetic Minority Over-Sampling Technique* Dan *Random Forest Clasifier*, 2023	Ketidak-seimbang-an kelas pada dataset jantung	SMOTE + *Random Forest*	Mengurangi bias kelas mayoritas	Akurasi 92%
4	Erliyan R. Susanto, Akbar E. Pranajaya	Optimasi *Random Forest* untuk Prediksi Penyakit Jantung Menggunakan SMOTEENN dan *Grid Search*, 2025	Data *imbalance* dan parameter model	SMOTE-ENN + RF + *Grid Search*	Optimasi performa klasifikasi	Akurasi 94%, AUC 0,99

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
5	Rahmad Firdaus dkk	Klasifikasi *Multi-Class* Penyakit Jantung Dengan SMOTE dan *Pearson's Correlation* menggunakan MLP, 2024	Klasifikasi lebih dari dua kelas	SMOTE + MLP	Pendekatan *multi-class* pada data jantung	Akurasi meningkat dibanding tanpa SMOTE
6	Silmi Ath Thahirah Al Azhima dkk	*Hybrid Machine Learning Model* untuk memprediksi Penyakit Jantung dengan Metode *Logistic Regression* dan *Random Forest*, 2025	Kebutuhan model yang stabil dan akurat	*Logistic Regression* + *Random Forest*	Perbandingan dan kombinasi model	RF lebih unggul dari LR

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
7	Nurliana Nasution dkk	*Predicting Heart Disease Using Machine Learning: An Evaluation of Logistic Regression, Random Forest, SVM, and KNN Models on the UCI Heart Disease Dataset*, 2025	Pemilihan algoritma terbaik	LR, RF, SVM, KNN	*Benchmark* model klasifikasi	RF: 89,7%, LR: 84,2%
8	Ahmad Ubai Dullah dkk	*Extreme Gradient Boosting Model with SMOTE for Heart Disease Classification Authors*, 2025	Akurasi diagnosis penyakit jantung	SMOTE + *XGBoost*	Perbandingan *boosting* dan model klasik	*XGBoost* unggul dari RF

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
9	Sabrina Putri Aulia dkk	*Enhancing Heart Disease Prediction through SMOTE-ENN Balancing and RFECV Feature Selection*, 2025	Data tidak seimbang dan fitur tidak relevan	SMOTE-ENN + RFECV + RF	Integrasi *balancing* dan *feature selection*	*Recall* 0,984, F1 0,938
10	Bilal Ahmad dkk	*Feature selection strategies for optimized heart disease diagnosis using ML and DL models*, 2025	Optimasi fitur prediktif	LR, RF + *Feature Selection*	Evaluasi dampak seleksi fitur	Akurasi hingga 82,3%

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
11	Ali Azimi Lamir dkk	*A Comprehensive Machine Learning Framework for Heart Disease Prediction: Performance Evaluation and Future Perspectives*, 2025	Evaluasi model ML	LR, RF, KNN	*Framework* evaluasi menyeluruh	RF akurasi 91%
12	Uswatun Hasanah dkk	*Effect of Random Under sampling, Oversampling, and SMOTE on the Performance of Cardiovascular Disease Prediction Models*, 2024	Pengaruh teknik *balancing*	*Oversampling*, *Undersampling*, SMOTE	Analisis perbandingan teknik	SMOTE paling stabil

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
13	Rahul Karmakar dkk	*A data balancing approach towards design of an expert system for Heart Disease Prediction*, 2024	Data minoritas sulit diprediksi	*K-Means* SMOTE + RF	Distribusi sintetik berbasis *cluster*	Akurasi hingga 99,83%
14	Yian Mao dkk	*Machine learning algorithms for heart disease diagnosis: A systematic review*, 2025	Ringkasan tren ML medis	*Review* sistematis	Identifikasi metode populer	RF dan LR paling sering digunakan

No	Penulis	Judul, Tahun	Masalah	Metode	Kontribusi	Hasil
15	Salma N. Khofiyah, A.P. Kuncoro	Prediksi Penyakit Jantung Mengguna-kan Algoritma *Random Forest* Dan *Synthetic Minority Oversampling Technique* (SMOTE), 2025	Prediksi penyakit jantung berbasis *web*	SMOTE + *Random Forest*	Implementasi sistem prediksi	Akurasi 86,58%

Penelitian terkait klasifikasi penyakit jantung berbasis *machine learning* menunjukkan perkembangan yang pesat, khususnya dalam penerapan algoritma *supervised learning* untuk mendukung diagnosis medis. Berbagai metode klasifikasi telah digunakan guna meningkatkan akurasi dan keandalan prediksi berdasarkan data klinis. Namun, tantangan utama yang sering dihadapi adalah ketidakseimbangan data, di mana jumlah pasien tanpa penyakit jantung lebih dominan dibandingkan pasien yang menderita penyakit jantung. Kondisi ini dapat menyebabkan penurunan performa model dalam mendeteksi kelas minoritas apabila tidak ditangani secara tepat.

Sejumlah penelitian menunjukkan bahwa algoritma *Random Forest* memiliki performa yang baik dalam menangani data medis yang kompleks dan berdimensi tinggi. Ketika dikombinasikan dengan teknik penyeimbangan data, khususnya metode *oversampling* seperti SMOTE, algoritma ini mampu meningkatkan kinerja klasifikasi secara signifikan. Penerapan metode tersebut terbukti dapat mengurangi bias terhadap kelas mayoritas serta meningkatkan kemampuan model dalam mendeteksi kasus positif, yang ditunjukkan melalui peningkatan nilai akurasi, *recall*, dan AUC.

Selain SMOTE konvensional, beberapa studi juga mengembangkan

pendekatan yang lebih lanjut, seperti kombinasi teknik *oversampling* dan *undersampling*, serta optimasi parameter model. Pendekatan ini bertujuan untuk menghasilkan distribusi data yang lebih representatif sekaligus meminimalkan kesalahan klasifikasi. Hasil penelitian menunjukkan bahwa pemilihan teknik penyeimbangan data yang tepat berpengaruh signifikan terhadap stabilitas dan performa model, terutama pada dataset medis yang tidak seimbang.

Di sisi lain, *Logistic Regression* masih banyak digunakan dalam penelitian klasifikasi penyakit jantung karena kesederhanaan dan kemudahan interpretasi modelnya. Algoritma ini dinilai sesuai untuk kebutuhan analisis medis yang memerlukan transparansi dalam pengambilan keputusan. Meskipun performanya cenderung lebih rendah dibandingkan metode *ensemble* seperti *Random Forest*, *Logistic Regression* tetap memberikan hasil yang kompetitif, terutama ketika dikombinasikan dengan teknik penyeimbangan data.

Penelitian terbaru mulai mengembangkan metode *oversampling* berbasis *clustering*, seperti *K-Means*-SMOTE, yang bertujuan untuk meningkatkan kualitas data sintetik yang dihasilkan. Teknik ini memanfaatkan struktur klaster dalam data untuk menghasilkan sampel minoritas yang lebih representatif dan mengurangi risiko tumpang tindih antar kelas. Penerapan metode ini dilaporkan mampu meningkatkan performa model secara signifikan, baik dari sisi akurasi maupun metrik evaluasi lainnya.

Berdasarkan kajian terhadap penelitian terdahulu, terlihat bahwa penggunaan algoritma *machine learning* seperti *Random Forest* dan *Logistic Regression* mampu menghasilkan kinerja yang baik dalam klasifikasi penyakit jantung, khususnya ketika dikombinasikan dengan teknik penyeimbangan data. *Random Forest* cenderung menghasilkan kinerja yang lebih unggul pada dataset tidak seimbang, sementara *Logistic Regression* tetap relevan karena kemudahan interpretasinya dalam konteks medis. Meskipun demikian, sebagian besar penelitian masih berfokus pada penggunaan teknik SMOTE konvensional atau kombinasi *oversampling–undersampling*, tanpa melakukan perbandingan yang komprehensif terhadap varian SMOTE yang lebih lanjut, seperti *Borderline SMOTE* dan *K-Means SMOTE*. Selain itu, evaluasi performa model pada beberapa penelitian masih terbatas pada metrik tertentu, sehingga belum memberikan gambaran menyeluruh terhadap kemampuan model dalam menangani ketidakseimbangan data. Oleh karena itu, diperlukan penelitian yang membandingkan secara sistematis kinerja *Random Forest* dan *Logistic Regression* dengan penerapan berbagai teknik *oversampling*, khususnya *Borderline SMOTE*, dan *K-Means SMOTE*, serta

mengevaluasinya menggunakan metrik performa yang lebih komprehensif untuk mengidentifikasi pendekatan yang paling optimal dalam klasifikasi penyakit jantung.