Jurnal Kajian dan Terapan Matematika
Volume 12, Edisi 2, Bulan Juli 2026, 53 – 70
http://journal.student.uny.ac.id/ojs/index.php/jktm

PERBANDINGAN KINERJA METODE *LOGISTIC REGRESSION*, *RANDOM FOREST*, DAN *XGBOOST* DENGAN SMOTE DAN *HYPERPARAMETER TUNING* DALAM KLASIFIKASI INFARK MIOKARD AKUT

PERFORMANCE COMPARISON OF LOGISTIC REGRESSION, RANDOM FOREST, AND XGBOOST WITH SMOTE AND HYPERPARAMETER TUNING IN THE CLASSIFICATION OF ACUTE MYOCARDIAL INFARCTION

Annisa Nur Rohmah*, Universitas Negeri Yogyakarta
Nur Insani, Universitas Negeri Yogyakarta
*e-mail korespondensi: annisanur1043@gmail.com

Abstrak
Infark Miokard Akut (IMA) merupakan salah satu penyebab utama kematian akibat penyakit kardiovaskular di dunia. Keterlambatan penanganan dapat menyebabkan berbagai komplikasi mulai dari gagal jantung hingga kematian mendadak, sehingga diperlukan adanya deteksi dini sebagai upaya menekan mortalitas akibat IMA. Penelitian ini bertujuan melakukan studi komparatif antara model *Logistic Regression*, *Random Forest*, dan *XGBoost* dalam klasifikasi IMA. Guna menangani masalah ketidakseimbangan kelas pada data medis serta mengoptimalkan kinerja model, diterapkan teknik *Synthetic Minority Oversampling Technique* (SMOTE) serta *hyperparameter tuning* menggunakan *Randomized Search Cross-Validation*. Kinerja model dievaluasi menggunakan nilai *Area Under the Curve* (AUC), *recall*, *precision*, *F1-score*, dan *accuracy*. Hasil penelitian menunjukkan pengaplikasian SMOTE dan *hyperparameter tuning* memberikan pengaruh yang bervariasi, terutama pada *Logistic Regression* yang mengalami peningkatan signifikan. *XGBoost* menunjukkan stabilitas kinerja di setiap skenario, sedangkan *Random Forest* dengan SMOTE menghasilkan kinerja paling tinggi dan jumlah kesalahan prediksi paling minimum, yaitu mencapai *recall* 1,000, *precision* 0,992, dan ROC-AUC sebesar 0,999. Dengan demikian, *XGBoost* unggul dari segi stabilitas, sementara *Random Forest* lebih akurat dan andal dari segi minimasi kesalahan prediksi dalam klasifikasi Infark Miokard Akut.
Kata kunci: *Extreme Gradient Boosting*, *Hyperparameter Tuning*, Infark Miokard Akut (IMA), *Logistic Regression*, *Random Forest*, *Synthetic Minority Oversampling Technique* (SMOTE)

Abstract
Acute myocardial infarction (AMI) is one of the leading causes of death from cardiovascular disease worldwide. Delayed treatment can lead to various complications ranging from heart failure to sudden death; therefore, early detection is essential to reduce mortality from AMI. This study aims to conduct a comparative analysis of the Logistic Regression, Random Forest, and XGBoost models for AMI classification. To address the issue of class imbalance in medical data and optimize model performance, the Synthetic Minority Oversampling Technique (SMOTE) was applied, along with hyperparameter tuning using Randomized Search Cross-Validation. Model performance was evaluated using the Area Under the Curve (AUC), recall, precision, F1-score, and accuracy. The results of the study show that the application of SMOTE and hyperparameter tuning has varying effects, particularly on Logistic Regression, which experienced a significant improvement. XGBoost demonstrated stable performance across all scenarios, while Random Forest with SMOTE produced the highest performance and the lowest number of prediction errors, achieving a recall of 1.000, precision of 0.992, and ROC-AUC of 0.999. Thus, XGBoost excels in terms of stability, while Random Forest is more accurate and reliable in terms of minimizing prediction errors in the classification of Acute Myocardial Infarction.
Keywords*: Acute Myocardial Infarction (AMI), Extreme Gradient Boosting, Hyperparameter Tuning, Logistic Regression, Random Forest, Synthetic Minority Oversampling Technique (SMOTE)*

PENDAHULUAN

Penyakit kardiovaskular atau *cardiovascular disease* (CVD) adalah istilah umum yang mencakup semua gangguan pada jantung dan pembuluh darah, termasuk hipertensi, penyakit arteri koroner, aritmia, penyakit serebrovaskular, penyakit katup jantung, kardiomiopati, penyakit pembuluh darah perifer, dan kelainan jantung bawaan (Davidson, 1994). CVD masih menjadi masalah kesehatan global yang serius. *World Health Organization* (WHO) melaporkan bahwa CVD menyebabkan sekitar 19,8 juta kasus kematian di seluruh dunia pada tahun 2022, dengan sebagian besar kasus disebabkan oleh serangan jantung dan stroke. Beban penyakit ini banyak terjadi di negara-negara berpendapatan rendah dan menengah, termasuk Indonesia (World Health Organization, 2025). Salah satu penyebab utama kematian akibat CVD adalah Infark Miokard Akut (*Acute Myocardial Infarction*/AMI), dengan prevalensi hampir mencapai 3 juta kasus secara global (Mechanic, Gavin, & Grossman, 2023; Amaliah, Yaswir, & Prihandani, 2019).

Infark Miokard Akut (IMA) atau dikenal sebagai serangan jantung (*heart attack*), merupakan kondisi medis akut yang disebabkan oleh kurangnya pasokan oksigen dalam waktu yang cukup lama sehingga mengakibatkan kematian sel-sel otot jantung (Salman, 2019; Asshidiq, Wisudawan, & Deus, 2025). Jika tidak segera ditangani, IMA dapat menyebabkan berbagai komplikasi mulai dari gagal jantung hingga kematian mendadak (Raharjo, Magfiroh, & Rokhman, 2025). Kondisi ini dipengaruhi oleh faktor risiko metabolik dan gaya hidup yang saling berkaitan secara kompleks, sehingga kondisi ini perlu menjadi perhatian dalam upaya deteksi dini dan penanganan untuk menekan angka mortalitas.

Pada era perkembangan teknologi, deteksi dini tidak lagi hanya mengandalkan metode klasik, tetapi juga memanfaatkan *Artificial Intelligence* (AI). AI membawa perubahan besar di dunia medis, terutama dalam analisis data dan deteksi penyakit secara dini (Rhomadon, Wydyanto, Mirza, & Huda, 2025). *Machine Learning* (ML) adalah cabang dari AI yang memiliki kemampuan mempelajari interaksi antarvariabel serta pola dalam data. Selain mampu menangani data yang besar dan kompleks, ML dapat memberikan hasil kinerja yang lebih akurat dan unggul dibandingkan metode klasik (Mandias & Manoppo, 2025; Reátegui, Tandazo-Malla, Suárez, & Ramírez-Cerna, 2025). Banyak penelitian terdahulu telah membuktikan bahwa algoritma ML seperti *Logistic Regression*, *Random Forest*, dan *Extreme Gradient Boosting* (*XGBoost*) mampu mengenali pola dan hubungan kompleks pada data, serta memiliki kinerja yang baik dalam kasus klasifikasi di bidang medis termasuk IMA.

Azhar & Sari (2022) serta Rhomadon, *et al.* (2025) menemukan bahwa *Logistic Regression* (LR) efektif dan mudah diinterpretasikan dalam membangun sistem prediksi CVD. Dibandingkan *K-Nearest Neighbors* (KNN), *Random Forest*, dan *Tuned* KNN, LR unggul dengan *accuracy* mencapai 88,52% . Di sisi lain, Mandias & Manoppo (2025) menunjukkan bahwa kinerja *Random Forest* (RF) lebih baik dari *Support Vektor Machine* (SVM) dalam memprediksi IMA, dengan *accuracy* sebesar 90% dan *recall* 93%. Sementara *Extreme Gradient Boosting* (*XGBoost*) yang unggul dalam akurasi dan kecepatan, menjadi model terbaik dengan AUC 99,6% dan *F1-score* 99,5% pada klasifikasi komplikasi Infark Miokard (IM) dibandingkan SVM, RF, dan *XGBoost* (Widodo, Brawijaya, & Samudi, 2022; Tajali, *et al.*, 2024).

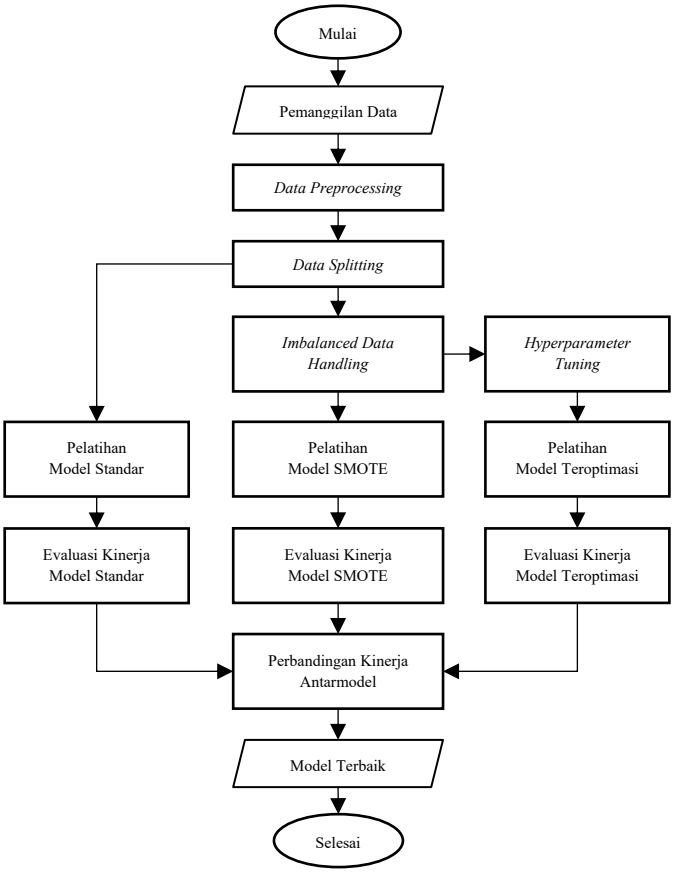
Meskipun berbagai penelitian terdahulu telah menunjukkan efektivitas ML dalam prediksi di bidang medis, masih terdapat tantangan yang memengaruhi akurasi prediksi model. Sebagian besar penelitian sebelumnya sering menemui masalah ketidakseimbangan kelas positif dan negatif pada data medis, sehingga model cenderung bias terhadap kelas mayoritas. Selain itu, tidak sedikit penelitian yang hanya membangun model dengan konfigurasi *hyperparameter* sesuai *default* tanpa melakukan optimasi atau proses *hyperparameter tuning*

(Tajali, *et al.*, 2024). Hal ini mengakibatkan potensi dari algoritma ML seperti LR, RF, dan *XGBoost* belum dieksplorasi secara maksimal.

Berdasarkan urgensi tersebut, penelitian ini bertujuan untuk melakukan studi komparatif antara model *Logistic Regression*, *Random Forest*, dan *XGBoost* dalam klasifikasi Infark Miokard Akut (IMA). Selain itu, teknik *Synthetic Minority Oversampling Technique* (SMOTE) diaplikasikan untuk menangani ketidakseimbangan data dan menerapkan *hyperparameter tuning* dengan metode *Randomized Search Cross-Validation* untuk memperoleh konfigurai *hyperparameter* terbaik, sehingga kinerja model menjadi lebih optimal. Evaluasi kinerja model masing-masing dilakukan melalui metrik *accuracy*, *recall*, *precision*, *F1-score*, dan nilai *Area Under the Curve* (AUC) guna menjamin akurasi, stabilitas, dan keandalan model dalam membedakan pasien positif dan negatif IMA.

METODE

Pada penelitian ini dilakukan klasifikasi Infark Miokard Akut menggunakan *Machine Learning* dengan tahapan metode penelitian disajikan pada Gambar 1.



Gambar 1. Tahapan Metode Penelitian

Dataset

Data yang digunakan dalam penelitian ini merupakan dataset *Heart Attack* atau Infark Miokard Akut (IMA) yang diperoleh melalui platform Mendeley Data, yaitu data rekam medis pasien dari Zheen Hospital di Erbil, Iraq, pada bulan Januari sampai Mei 2019 (Rashid & Hassan, 2022). Dataset terdiri dari 1.319 observasi dengan 9 variabel sebagai berikut:

1. *Age*: usia pasien dalam satuan tahun.

2. *Gender*: jenis kelamin pasien, yaitu laki-laki (1) dan perempuan (0).
3. *Heart rate*: frekuensi detak jantung pasien per menit (*beats per minute* atau bpm).
4. *Systolic blood pressure*: tekanan darah sistolik pasien, yaitu tekanan maksimum dalam arteri saat jantung kontraksi dan merupakan angka pertama pada tekanan darah.
5. *Diastolic blood pressure*: tekanan darah diastolik pasien, yaitu tekanan minimum dalam arteri saat jantung relaksasi dan merupakan angka kedua pada tekanan darah.
6. *Blood sugar*: kadar gula darah pasien.
7. CK-MB: kadar *Creatine Kinase*-MB (CK-MB) dalam darah pasien, yaitu enzim yang dilepaskan saat otot jantung mengalami kerusakan dan merupakan salah satu indikator utama saat mengevaluasi IMA.
8. *Troponin*: kadar *troponin* pasien, yaitu protein yang juga dilepaskan ke darah saat otot jantung mengalami kerusakan dan merupakan satu indikator utama lainnya dalam mengevaluasi IMA.
9. *Result*: variabel target berisi label status berisiko terkena IMA, yaitu positif (1) dan negatif (0).

Logistic Regression (LR)

Logistic Regression (LR) merupakan metode klasifikasi statistik untuk memodelkan probabilitas suatu kejadian biner, yaitu kejadian dengan dua kemungkinan kelas atau kategori. Metode ini sering digunakan sebagai model dasar dalam klasifikasi karena memiliki struktur yang sederhana dan mudah diinterpretasikan (Song, Liu, Liu, & Wang, 2021). LR dapat digunakan untuk menganalisis hubungan antara satu variabel biner dependen dengan satu atau lebih variabel independen, baik yang berskala nominal, ordinal, interval, maupun rasio (Pangaribuan, Tanjaya, & Kenichi, 2021).

Pada *Logistic Regression*, model terlebih dahulu menghitung kombinasi linear dari fitur *input* dengan Persamaan (1),

$$z = w^T x + b \tag{1}$$

dengan z = kombinasi linear fitur *input*,
w^T = vektor bobot,
x = fitur,
b = bias.

Namun, karena hasil dari kombinasi linear tersebut dapat bernilai sangat kecil atau sangat besar, nilai perlu diubah menjadi probabilitas. Oleh karena itu, digunakan fungsi sigmoid untuk memetakan nilai z ke dalam rentang 0 sampai 1. Probabilitas yang dihasilkan kemudian digunakan untuk menentukan apakah suatu data termasuk kelas 0 atau kelas 1. Fungsi sigmoid didefinisikan pada Persamaan (2),

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

dengan $\sigma(z)$ adalah fungsi sigmoid terhadap z.

Secara umum, proses pembelajaran pada model klasifikasi LR adalah sebagai berikut (Chaudhry, 2024):

1. Menginisialisasi vektor bobot $w^T = [w_1 \ w_2 \ \dots \ w_n]$ dengan n adalah banyak fitur, dan inisialisasi bias b dengan nilai acak, umumnya sekitar 0.
2. Menghitung nilai *loss* atau *binary cross-entropy* dengan data latih menggunakan persamaan (3),

$$L(w,b) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \ln\left(\sigma\left(w^T x^{(i)} + b\right)\right) + \left(1 - y^{(i)}\right) \ln\left(1 - \sigma\left(w^T x^{(i)} + b\right)\right) \tag{3}$$

dengan $L(w, b)$ = *loss* atau *binary cross-entropy*,
m = banyaknya data latih,
$x^{(i)}$ = data latih ke-i,
$y^{(i)}$ = label untuk data latih ke-i.

3. Menghitung gradien *loss* terhadap setiap bobot dan bias menggunakan Persamaan (4) dan Persamaan (5),

$$\frac{\partial L(w,b)}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}\left(\sigma\left(w^T x^{(i)} + b\right) - y^{(i)}\right)x_j^{(i)} \tag{4}$$

$$\frac{\partial L(w,b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\sigma\left(w^T x^{(i)} + b\right) - y^{(i)} \tag{5}$$

dengan w_j = bobot dari fitur ke-j,
$\frac{\partial L(w,b)}{\partial w_j}$ = gradien *loss* terhadap bobot ke-j,
$\frac{\partial L(w,b)}{\partial b}$ = gradien *loss* terhadap bias.

4. Memperbarui nilai bobot dan bias secara bersamaan menggunakan Persamaan (6) dan Persamaan (7),

$$w_j = w_j - \alpha\frac{\partial L(w,b)}{\partial w_j} \tag{6}$$

$$b = b - \alpha\frac{\partial L(w,b)}{\partial b} \tag{7}$$

dengan α adalah *learning rate*.

5. Kembali ke langkah kedua dan mengulangi proses perhitungan *loss*, gradien, serta pembaruan bobot dan bias hingga nilai *loss* tidak menunjukkan penurunan signifikan atau hingga mencapai jumlah iterasi yang ditentukan.
6. Melakukan prediksi pada data uji berdasarkan probabilitas yang dihasilkan oleh fungsi sigmoid. Dengan ambang batas umum sebesar 0,5, data diklasifikasikan sebagai kelas 0 apabila $\sigma(w^T x + b) \leq 0{,}5$, dan diklasifikasikan sevagai kelas 1 apabila $\sigma(w^T x + b) > 0{,}5$.

Random Forest (RF)

Random Forest (RF) merupakan algoritma *ensamble learning* berbasis pohon keputusan (*decision tree*) yang membangun banyak pohon (hutan) untuk menghasilkan prediksi yang lebih stabil. Setiap pohon dalam RF merupakan pohon yang dipartisi biner secara rekursif. Pohon dilatih menggunakan sampel data yang berbeda melalui teknik *bootstrap sampling*, serta hanya mempertimbangkan sebagian fitur secara acak pada setiap proses pemisahan simpul (Cutler, Cutler, & Stevens, 2012). Pendekatan ini membuat pohon-pohon yang terbentuk menjadi lebih beragam dan mengurangi ketergantungan antarpohon, sehingga RF mampu menangkap hubungan nonlinear antarvariabel serta mengurangi *overfitting* dibandingkan satu pohon keputusan tunggal (Rahmada & Susanto, 2024; Widodo, *et al.*, 2022).

Mekanisme kerja RF dalam membuat prediksi diilustrasikan pada Gambar 3 dan dapat dijelaskan melalui langkah-langkah berikut (Cutler, *et al.*, 2012; Kumar, 2020/2021?):

1. Membuat N sampel *bootstrap* dari data latih asli.
2. Untuk setiap sampel *bootstrap*, dibangun satu pohon keputusan dengan tahapan sebagai berikut:
   a. Menempatkan seluruh observasi dalam satu simpul.

b. Mengulangi langkah berikut secara rekursif untuk setiap simpul hinga diperoleh simpul daun atau pohon lengkap:
   1) Pada setiap simpul, memilih m fitur secara acak dari total p fitur. Secara umum, banyak fitur yang dipilih dapat ditentukan dengan $m \approx \sqrt{p}$ atau $m \approx \frac{p}{3}$ (Han, Kim, & Lee, 2020).
   2) Membagi simpul menjadi dua partisi berdasarkan fitur dan titik potong terbaik.
3. Melatih setiap pohon secara independen menggunakan sampel *bootstrap* yang berbeda.
4. Melakukan prediksi pada data uji menggunakan setiap pohon yang telah dibangun.
5. Menggabungkan hasil prediksi dari seluruh pohon. Pada kasus klasifikasi, keputusan akhir RF ditentukan berdasarkan suara terbanyak (*majority voting*).

***Extreme Gradient Boosting* (*XGBoost*)**

Extreme Gradient Boosting (*XGBoost*) merupakan algoritma *ensemble learning* berbasis *Gradient Boosting Decision Tree* yang membangun sejumlah pohon secara bertahap. Berbeda dengan *Random Forest* yang membangun banyak pohon secara independen, *XGBoost* membangun pohon secara iteratif, yaitu setiap pohon baru digunakan untuk memperbaiki kesalahan prediksi dari pohon sebelumnya. Dengan pendekatan tersebut, *XGBoost* mampu menghasilkan model yang kuat, efisien, dan sering digunakan pada data tabular karena kemampuannya menangkap hubungan nonlinear antarvariabel (Reátegui, *et al.*, 2025). Algoritma ini bekerja dengan meminimalkan fungsi objektif yang terdiri atas dua komponen utama, yaitu fungsi kerugian (*loss function*) dan regularisasi. Fungsi kerugian digunakan untuk mengukur kesalahan prediksi model, sedangkan regularisasi untuk mengontrol kompleksitas model agar tidak terjadi *overfitting*. Secara umum, fungsi objektif *XGBoost* pada iterasi ke-t dapat dituliskan seperti pada Persamaan (8),

$$obj^t = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega(f_t) \tag{8}$$

dengan n = banyaknya data latih,
l = fungsi kerugian,
y_i = label aktual data latih ke-i,
$\hat{y}_i^{t-1}$ = probabilitas prediksi dari model sebelumnya,
$f_t(x_i)$ = skor dari pohon keputusan baru pada iterasi ke-t,
$\Omega(f_t)$ = fungsi regularisasi.

Mekanisme kerja *XGBoost* dalam klasifikasi biner dapat dijelaskan melalui langkah-langkah berikut (Fajriyah, Isnandar, & Arifuddin, 2024; Omarzai, 2024):

1. Menentukan nilai prediksi awal untuk setiap data latih. Pada klasifikasi biner, nilai awal dapat dinyatakan dalam bentuk probabilitas $\hat{y}_i^{0}$, dengan nilai *default* sebesar $0{,}5$.
2. Menghitung gradien atau residual, yaitu selisih antara label aktual dengan nilai prediksi model sebelumnya. Residual pada iterasi ke-t dituliskan pada Persamaan (9),
$$Residual_i^t = y_i - \hat{y}_i^{t-1} \tag{9}$$
dengan $Residual_i^t$ = residual data latih ke-i pada iterasi ke-t,
y_i = label aktual data latih ke-i,
$\hat{y}_i^{t-1}$ = probabilitas prediksi dari model sebelumnya.
3. Membangun pohon keputusan baru untuk mempelajari residual atau kesalahan prediksi dari model sebelumnya. Pada tahap ini, *XGBoost* memilih fitur dan titik pemisahan berdasarkan nilai *gain*. Nilai *gain* menunjukkan seberasapa besar peningkatan kualitas model setelah suatu simpul dipartisi menjadi dua bagian.

4. Menghitung *similarity score* dan *gain* untuk menentukan titik pemisahan simpul terbaik. *Similarity score* dihitung menggunakan Persamaan (10), sedangkan *gain* dihitung dengan membandingkan nilai *similarity score* pada simpul kiri, simpul kanan, dan simpul akar menggunakan Persamaan (11),
$$Similarity\ Score = \frac{(\sum_{i=1}^{n} Residual_i^t)^2}{\sum_{i=1}^{n} \hat{y}_i^{t-1}\left(1-\hat{y}_i^{t-1}\right)+\lambda} \tag{10}$$
$$Gain = Left_{similarity} + Right_{similarity} - Root_{similarity} \tag{11}$$
dengan λ adalah *hyperparameter* regularisasi yang digunakan untuk membatasi jumlah daun sebagai penalti terhadap model yang terlalu kompleks, misalnya *overfitting*.
5. Melakukan pemisahan ulang simpul pada pohon dengan nilai *gain* terbesar hingga mencapai batas maksimum kedalaman pohon atau kriteria penghentian tertentu.
6. Melakukan pemangkasan atau *pruning* pada bagian pohon yang tidak memberikan peningkatan kinerja yang cukup. Proses ini dikontrol oleh *hyperparameter* γ, sehingga cabang dengan *gain* yang teralu kecil dapat dihilangkan untuk mengurangi risiko *overfitting*.
7. Menghitung nilai *output* pada setiap simpul daun. Nilai *output* pada daun ke-j dihitung menggunakan Persamaan (12),
$$Output_j^t = \frac{\sum_{i=1}^{n} Residual_i^t}{\sum_{i=1}^{n} \hat{y}_i^{t-1}\left(1-\hat{y}_i^{t-1}\right)+\lambda} \tag{12}$$
dengan $Output_j^t$ merupakan nilai *output* untuk daun ke-j pada iterasi ke-t.
8. Memperbarui nilai prediksi model dengan menambahkan kontribusi dari pohon baru yang telah dibangun. Pembaruan prediksi dikontrol oleh *learning rate* η, sehingga model tidak berubah terlalu besar pada setiap iterasi. Secara umum, pembaruan prediksi dihitung menggunakan Persamaan (13),
$$Predicted\ Value_i^t = \log\left(\frac{\hat{y}_i^{t-1}}{1-\hat{y}_i^{t-1}}\right) + \eta\ Output_j^t \tag{13}$$
dengan $Predicted\ Value_i^t$ merupakan prediksi baru data latih ke-i pada iterasi ke-t.
9. Dalam klasifikasi biner, nilai prediksi akhir diubah ke dalam bentuk probabilitas menggunakan fungsi sigmoid seperti pada Persamaan (14).
$$\hat{y}_i^{\,t} = \frac{1}{1+e^{-Predicted\ Value_i^t}} \tag{14}$$
10. Kembali ke langkah b dan mengulangi proses untuk pembentukan setiap pohon baru hingga mencapai jumlah pohon yang ditentukan atau kriteria penghentian tertentu.
11. Model akhir digunakan untuk membuat prediksi kelas pada data uji berdasarkan probabilitas yang dihasilkan. Umumnya data diklasifikasikan sevagai kelas 1 apabila probabilitas lebih besar dari 0,5 dan sebagai kelas 0 apabila probabilitas kurang dari atau sama dengan 0,5.

Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu teknik dalam menangani ketidakseimbangan data, yaitu metode *oversampling* yang menghasilkan data sintetis baru melalui interpolasi linear acak antara data kelas minoritas dengan tetangga terdekatnya tanpa mengorbankan informasi dalam kelas mayoritas. Secara umum, mekanisme kerja SMOTE dapat diuraikan dalam langkah-langkah berikut (Rahayu, *et al*., 2024):

1. Mengidentifikasi kelas minoritas.
2. Memilih 1 sampel x_i secara acak dari kelas minoritas dan k tetangga terdekatnya, secara *default* adalah 5 tetangga, berdasarkan jarak *Euclidean*.

3. Mengambil 1 tetangga secara acak $\hat{x}_i$ dan membuat sampel baru x' menggunakan Persamaan (15),
$$x' = x_i + rand(0,1) \times | x_i - \hat{x}_i | \quad (15)$$
dengan x' = data sampel baru,
x_i = data sampel yang dipilih secara acak dari kelas minoritas,
$rand(0,1)$ = bilangan acak di antara 0 dan 1,
$\hat{x}_i$ = satu tetangga yang dipilih secara acak dari k tetangga terdekat x_i.
4. Kembali ke langkah pertama dan mengulangi proses hingga diperoleh jumah data yang seimbang.

Randomized Search Cross-Validation

Randomized Search Cross-Validation adalah salah satu metode dalam *hyperparameter tuning* yang melakukan percobaan berbagai kombinasi *hyperparameter* secara acak dari ruang pencarian yang ditentukan sebelumnya, dengan cara kerja sebagai berikut (Prasetya & Rosa, 2024):

1. Menentukan ruang pencarian berisi nilai-nilai *hyperparameter* yang akan diuji.
2. Menentukan jumlah percobaan atau iterasi yang akan dilakukan *Random Search*.
3. Pada setiap iterasi:
   a. Memilih kombinasi *hyperparameter* secara acak dari nilai-nilai yang telah ditentukan di awal proses *tuning*.
   b. Mengevaluasi model dengan *Stratified K-Fold Cross-Validation* untuk memperoleh skor kinerja kombinasi *hyperparameter* tersebut.
4. Memilih kombinasi *hyperparameter* dengan skor kinerja terbaik dari semua iterasi.

Metrik Evaluasi

1. *Confusion Matrix*

		Predicted	
		Negative	*Positive*
Actual	*Negative*	TN	FP
	Positive	FN	TP

Gambar 2. *Confusion Matrix*

Confusion matrix seperti pada Gambar 2 menggambarkan perbandingan prediksi model dengan label aktual. TP (*True Positive*) menunjukkan banyak pasien positif Infark Miokard Akut (IMA) yang berhasil diprediksi model sebagai positif, TN (*True Negative*) banyak pasien negatif IMA yang berhasil diprediksi model sebagai negatif , FP (*False Positive*) banyak pasien yang diprediksi sebagai positif tetapi sebenarnya adalah pasien negatif IMA, dan FN (*False Negative*) banyak pasien yang diprediksi sebagai negatif tetapi sebenarnya adalah pasien positif IMA. Secara berturut-turut *accuracy*, *precision*, *recall*, dan *F1-score* model dihitung menggunakan persamaan-persamaan berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

$$precision = \frac{TP}{TP + FP} \quad (17)$$

$$recall = \frac{TP}{TP + FN} \quad (18)$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (19)$$

2. *Area Under the Curve* (AUC)

AUC merupakan metrik ringkasan dari kurva *Receiver Operating Characteristic* (ROC) yang menggambarkan kemampuan model dalam membedakan individu yang sakit (kelas positif) dan yang tidak sakit (kelas negatif) melalui plot rasio *True Positive Rate* (TPR) dengan *False Positive Rate* (FPR) (Kurnia, Nurdin, & Khaidar, 2025).

HASIL DAN PEMBAHASAN

Hasil

1. *Data Preprocessing*

a. Penanganan *missing value* dan duplikasi data

Pada pemeriksaan keseluruhan data tidak ditemukan adanya *missing value* maupun duplikasi data di setiap variabel, sehingga tidak perlu dilakukan imputasi, penghapusan data, atau penanganan lainnya.

b. Penanganan *outlier*

Pengecekan *outlier* atau pencilan penting untuk mengetahui adanya nilai-nilai yang menyimpang jauh dari obserbasi lainnya. Pada penelitian ini *outlier* diidentifikasi berdasarkan batas wajar fisiologis dalam data klinis seperti usia, angka CK-MB serta *troponin* harus bernilai positif, *heart rate* tidak melebihi batas maksimal 300 bpm seperti yang terjadi pada penderita *Ventricular Tachycardia* (CardioSmart, 2015), dan angka *diastolic blood pressure* harus kurang dari *systolic blood pressure*. Berdasarkan pengecekan *outlier* diketahui bahwa terdapat 3 observasi dengan *heart rate* di atas 300 bpm serta 11 observasi yang memiliki angka *diastolic blood pressure* sama dengan atau lebih dari *systolic blood pressure*. Baris data yang memuat observasi tidak wajar ini perlu dihapus untuk menjaga relevansi informasi data dalam deteksi IMA, sehingga setelah dilakukan penghapusan *outlier* diperoleh 1.305 observasi yang akan diolah dan dianalisis pada proses selanjutnya.

c. Encoding

Encoding dilakukan pada variabel kategorik, yaitu variabel *Gender* dan variabel target *Result*. Kedua variabel biner tersebut dienkode menggunakan teknik *Label Encoding*. Variabel *Gender* dengan nilai 0 merepresentasikan perempuan dan 1 adalah laki-laki, sedangkan variabel *Result* dengan nilai negatif dikodekan menjadi 0 dan positif dikodekan menjadi 1.

2. *Data Splitting*

Data splitting dilakukan dengan membagi data menjadi 70% data latih dan 30% data uji menggunakan *stratified split*. *Stratified split* digunakan agar proporsi kelas pada data latih maupun data uji tetap konsisten. Pada proses ini, diperoleh 913 observasi sebagai data latih dan 392 observasi sebagai data uji. Setelah pembagian data, dilakukan normalisasi data dengan teknik *Standardization* atau normalisasi dengan *Z-score* menggunakan *StandardScaler* dari *library scikit-learn*. Normalisasi dilakukan dengan menghitung nilai rata-rata dan standar deviasi pada data latih, kemudian digunakan untuk mentransformasi data latih dan data uji ke dalam skala standar.

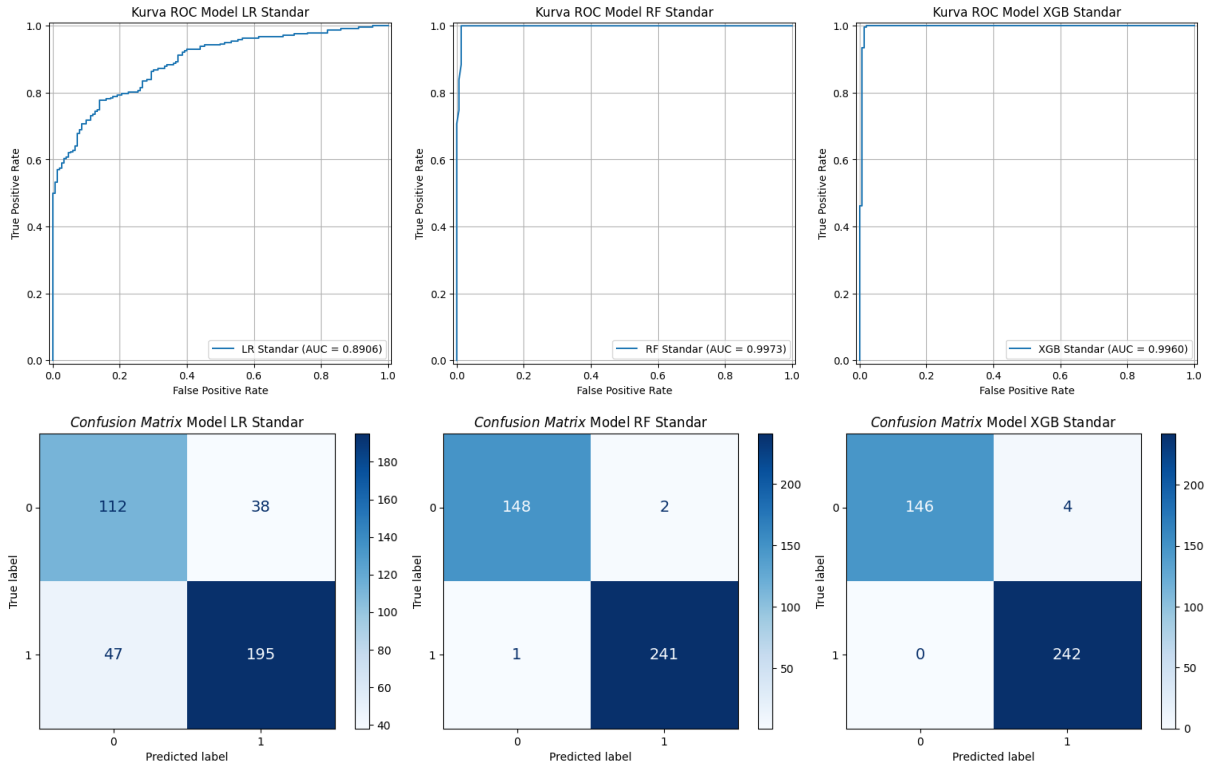
3. Pelatihan dan Evaluasi Model Standar

a. Pelatihan model standar

Pelatihan model standar menggunakan data latih awal yang berjumlah 913 observasi dengan 562 pasien positif dan 351 pasien negatif Infark Miokard Akut (IMA), serta menggunakan konfigurasi *hyperparameter* sesuai *default*. Selain itu karena distribusi label IMA tidak seimbang, masing-masing model dievaluasi menggunakan *Stratified K-Fold Cross-Validation* (SKCV) dengan 5-*fold* selama pelatihan untuk mencegah *overfitting* serta sebagai estimasi kemampuan generalisasi model.

b. Evaluasi model standar

Setelah model LR standar, RF standar, dan *XGBoost* standar dilatih, dilakukan prediksi menggunakan data uji. Model LR standar menghasilkan kinerja yang cukup baik, dengan *F1-score* 0,821, *recall* 0,806, *precision* 0,837, *accuracy* 0,783, dan nilai ROC-AUC 0,891. Di sisi lain, kesalahan prediksi yang dilakukan model LR standar masih banyak, yaitu mencapai 47 *False Negative* (FN) dan 38 *False Positive* (FP).



Gambar 3. Kurva ROC & *Confusion Matrix* Kinerja Model Standar, dari kiri ke kanan: *Logistic Regression*, *Random Forest*, *XGBoost*

Berbeda dengan LR, model RF standar dan *XGBoost* standar menghasilkan kinerja yang sangat baik, melebihi kinerja LR standar. Model RF standar memiliki *F1-score* 0,994, *recall* 0,996, *precision* 0,992, *accuracy* 0,992, dan nilai ROC-AUC 0,997. Sedangkan, model *XGBoost* standar memiliki *F1-score* 0,992, *recall* 1, *precision* 0,984, *accuracy* 0,99, dan nilai ROC-AUC 0,996. Kinerja RF lebih unggul dibandingkan *XGBoost* pada seluruh

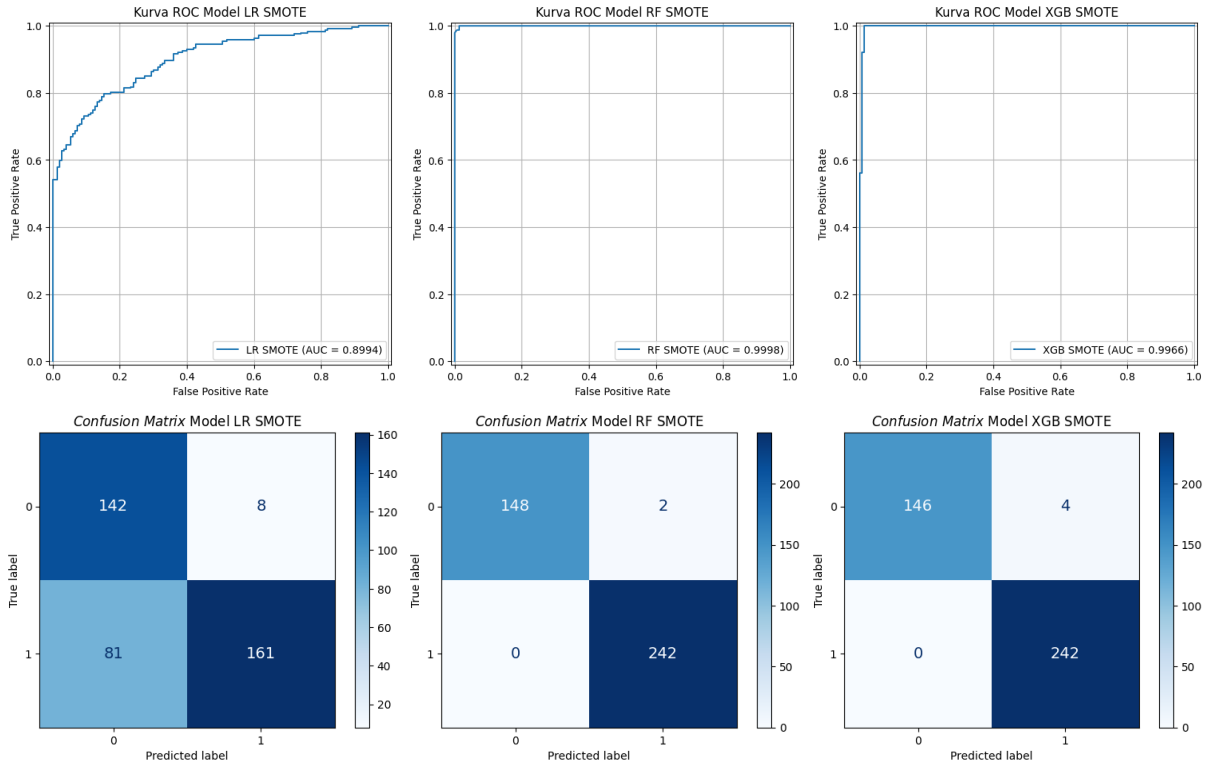
metrik kecuali *recall*. Hal ini menunjukkan adanya sedikit selisih pada jumlah kesalahan prediksi, yaitu RF dengan 1 FN dan 2 FP, sedangkan XGBoost memiliki 4 FP. Lebih lanjut, *confusion matrix* dan plot ROC-AUC dari kinerja setiap model standar disajikan pada Gambar 3.

4. Pelatihan dan Evaluasi Model dengan SMOTE

a. Pelatihan model dengan SMOTE

Model *Logistic Regression* (LR), *Random Forest* (RF), dan *Extreme Gradient Boosting* (*XGBoost*) dilatih menggunakan konfigurasi *hyperparameter* sesuai *default*. Selama pelatihan, masing-masing model dievaluasi menggunakan *Stratified K-Fold Cross-Validation* (SKCV) dengan 5-*fold* untuk mencegah *overfitting* serta sebagai estimasi kemampuan generalisasi model. Di samping itu, penanganan ketidakseimbangan distribusi kelas dengan SMOTE dilakukan di dalam *pipeline cross-validation* bersama standardisasi serta model klasifikasi. Hal ini dilakukan agar pada setiap *fold cross-validation*, SMOTE hanya diterapkan pada data latih dalam *fold* tersebut dan mencegah kebocoran data.

b. Evaluasi model dengan SMOTE



Gambar 4. Kurva ROC & *Confusion Matrix* Kinerja Model dengan SMOTE, dari kiri ke kanan: *Logistic Regression, Random Forest, XGBoost*

Setelah model LR, RF, dan *XGBoost* dilatih, dilakukan prediksi terhadap data uji. Model LR dengan SMOTE menghasilkan kinerja dengan *F1-score* 0,784, *recall* 0,665, *precision* 0,953, *accuracy* 0,773, dan nilai ROC-AUC 0,899. Meskipun dibandingkan LR standar nilai *precision* pada LR SMOTE meningkat dan jumlah FP menurun dari 38 menjadi 10, nilai *recall* justru mengalami penurunan dan jumlah FN meningkat 47 menjadi 81.

Peningkatan jumlah FN sangat berbahaya dalam kasus medis, sehingga optimasi model LR dalam klasifikasi IMA tidak cukup hanya dengan penyeimbangan distribusi kelas saja.

Model RF dan *XGBoost* menunjukkan kinerja yang sangat baik dan masih mengungguli LR. Dibandingkan RF standar, kinerja model RF dengan SMOTE meningkat dengan *F1-score* 0,996, *recall* 1, *precision* 0,992, *accuracy* 0,995, nilai ROC-AUC 0,997, dan jumlah kesalahan prediksi menurun menjadi 2 FP saja. Sebaliknya, kinerja *XGBoost* dengan SMOTE mengalami sedikit penurunan pada nilai ROC-AUC dari 0,996 menjadi 0,995 dibandingk *XGboost* standar. Kinerja model LR, RF, serta *XGBoost* dengan SMOTE disajikan dalam *confusion matrix* dan plot ROC-AUC pada Gambar 4.

5. Pelatihan dan Evaluasi Model Teroptimasi

a. Pelatihan model teroptimasi

Model *Logistic Regression* (LR), *Random Forest* (RF), dan *Extreme Gradient Boosting* (*XGBoost*) dilatih dengan data latih yang sudah seimbang melalui SMOTE serta konfigurasi *hyperparameter* hasil *tuning*. Penelitian ini menggunakan teknik *Randomized Search Cross-Validation* dengan *RandomizedSearchCV* dari *library scikit-learn* agar model dapat bekerja dengan konfigurasi *hyperparameter* yang tepat dan menghasilkan kinerja yang lebih optimal. *Randomized Search Cross-Validation* mencari kombinasi konfigurasi *hyperparameter* terbaik dari banyak kombinasi acak berdasarkan ruang pencarian berisi nilai-nilai *hyperparameter* yang ditentukan sebelumnya. Di dalam proses *tuning*, *Stratified K-Fold Cross-Validation* (SKCV) dengan 5-*fold* digunakan pada setiap iterasi, dengan *recall* sebagai metrik evaluasi yang dioptimalkan. SMOTE juga dilakukan di dalam *pipeline cross-validation* bersama standardisasi serta model klasifikasi seperti pada skenario model SMOTE sebelumnya. Dari proses *tuning* ini diperoleh kombinasi *hyperparameter* terbaik dengan kinerja terbaik sebagai model akhir yang akan digunakan. Secara berturut-turut Tabel 2, 3, dan 4 menyajikan ruang pencarian serta konfigurasi *hyperparameter* terbaik hasil *tuning* untuk model *Logistic Regression* (LR), *Random Forest* (RF), dan *Extreme Gradient Boosting* (*XGBoost*).

Tabel 1. Ruang Pencarian dan Nilai *Hyperparameter* Terbaik Hasil *Tuning* Untuk *Logistic Regression*

Hyperparameter	Rentang Nilai	Nilai Terbaik
Solver	Liblinear, saga, lbfgs	Liblinear
Penalty	L1, l2	L1
C	0,001, 0,01, 0,1, 1, 10, 100	100

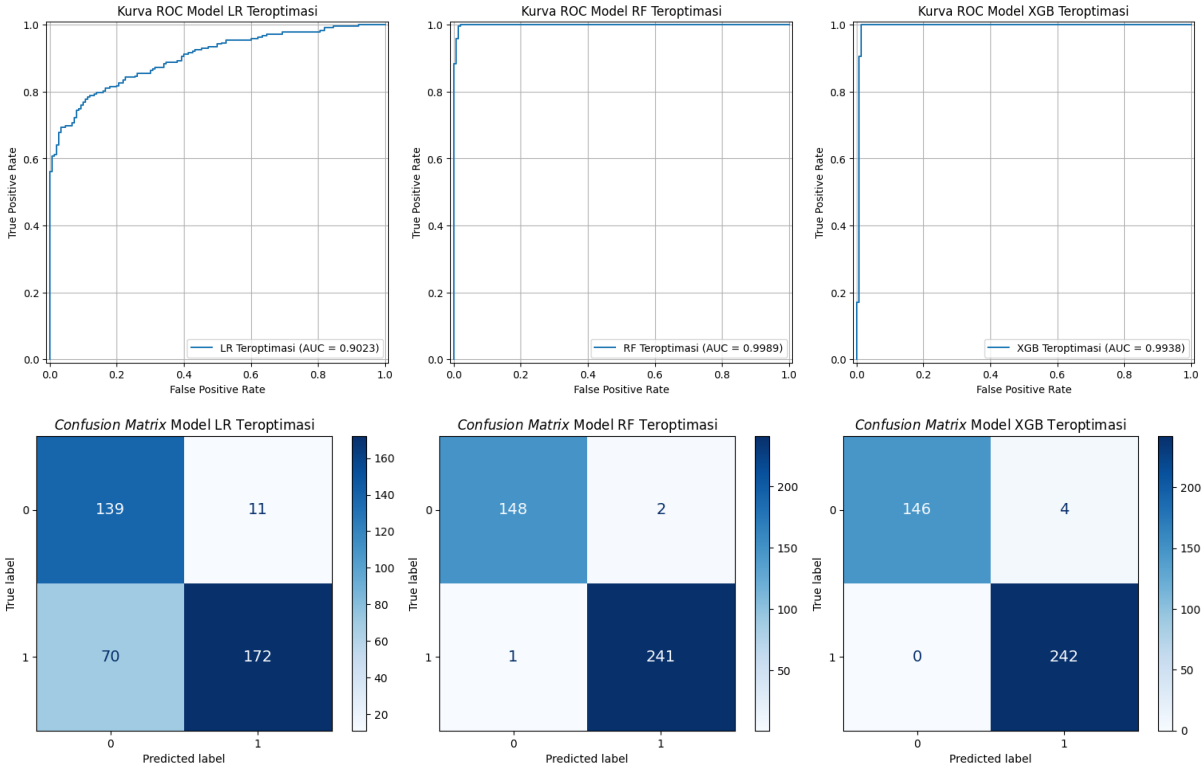
Tabel 2. Ruang Pencarian dan Nilai *Hyperparameter* Terbaik Hasil *Tuning* Untuk *Random Forest*

Hyperparameter	Rentang Nilai	Nilai Terbaik
Class_weight	*None*, *balanced*, *balanced_subsample*	*None*
Criterion	*Gini*, *entropy*	*Entropy*
Max_depth	*None*, 5, 10, 15, 20, 25, 30	*25*
Min_samples_leaf	1-10	8
Min_samples_split	2-10	6
N_estimators	100-1000	714

Tabel 3. Ruang Pencarian dan Nilai *Hyperparameter* Terbaik Hasil *Tuning* Untuk *XGBoost*

Hyperparameter	Rentang Nilai	Nilai Terbaik
Colsample_bytree	0,6-1,0	0,974
Gamma	0,1-1,0	0,807
Learning_rate	0,01-0,2	0,137
Max_depth	3-10	8
Min_child_weight	1-5	3
N_estimators	100-500	211
Reg_alpha	0,1-1,0	0,191
Reg_lambda	0,1-1,0	0,697
Subsample	0,6-0,9	0,602

b. Evaluasi model teroptimasi



Gambar 5. Kurva ROC & *Confusion Matrix* Kinerja Model Teroptimasi, dari kiri ke kanan: *Logistic Regression, Random Forest, XGBoost*

Setelah model LR, RF, dan XGBoost dilatih menggunakan konfigurasi hyperparameter terbaik hasil tuning, dilakukan prediksi menggunakan data uji. Dibanding LR SMOTE, model LR teroptimasi menunjukkan peningkatan hampir di seluruh metrik kecuali precision. F1-score meningkat dari 0,78 menjadi 0,809, recall 0,665 menjadi 0,711, accuracy 0,768 menjadi 0,793, dan ROC-AUC 0,899 menjadi 0,902, sedangkan precison sedikit menurun dari 0,942 menjadi 0,94. Peningkatan recall dan penurunan precision merepresentasikan menurunnya FN dan meningkatnya FP dari 81 FN dan 10 FP menjadi 70 FN dan 11 FP.

Berbeda dengan LR, Kinerja model RF teroptimasi sedikit menurun dari RF SMOTE, dengan *F1-score* 0,994, *recall* 0,996, *precision* 0,992, *accuracy* 0,992, serta jumlah kesalahan sebanyak 1 FN dan 2 FP. Akan tetapi apabila dibandingkan dengan RF standar, kinerja RF teroptimasi masih lebih unggul pada nilai ROC-AUC sebesar 0,999. Di sisi lain, kinerja *XGBoost* cenderung stabil baik ketika dilatih menggunakan *hyperparameter* terbaik hasil *tuning* maupun *hyperparameter* sesuai *default*. Dibanding *XGBoost* standar dan *XGBoost* SMOTE, model *XGBoost* teroptimasi mengalami sedikit penurunan pada nilai ROC-AUC menjadi 0,994 dengan jumlah kesalahan prediksi yang sama, yaitu 4 FP.

Meskipun kinerja RF dan *XGBoost* teroptimasi sedikit menurun, tetapi masih lebih unggul dari LR teroptimasi. Lebih lanjut, *confusion matrix* dan plot ROC-AUC hasil kinerja model LR, RF, serta *XGBoost* teroptimasi disajikan pada Gambar 5.

Pembahasan

Hasil penelitian menunjukkan bahwa model *Logistic Regression* (LR), *Random Forest* (RF), dan *Extreme Gradient Boosting* (*XGBoost*), mampu bekerja dengan baik dalam klasifikasi Infark Miokard Akut (IMA). Ketiga model dilatih menggunakan tiga skenario pelatihan, yaitu model standar, model dengan SMOTE, dan model teroptimasi dengan *hyperparameter tuning*. Hasil kinerja semua model dievaluasi menggunakan nilai AUC dengan 95% *Confidence Interval* (CI), *F1-score*, *recall*, *precision*, dan *accuracy* disajikan dalam Tabel 5.

Tabel 4. Perbandingan Kinerja Semua Model Klasifikasi

Metode	AUC (95% CI)	*F1-score*	*Recall*	*Precision*	*Accuracy*
LR Standar	0,874 (0,836-0,907)	0,822	0,788	0,858	0,773
RF Standar	0,992 (0,979-1,000)	0,992	0,988	0,996	0,990
XGBoost Standar	0,993 (0,983-1,000)	0,992	0,988	0,996	0,990
LR SMOTE	0,878 (0,841-0,911)	0,808	0,712	0,934	0,776
RF SMOTE	0,993 (0,982-1,000)	0,992	0,988	0,996	0,990
XGBoost SMOTE	0,992 (0,980-1,000)	0,988	0,988	0,988	0,985
LR Teroptimasi	0,971 (0,953-0,986)	0,911	0,850	0,982	0,890
RF Teroptimasi	0,993 (0,982-1,000)	0,992	0,988	0,996	0,990
XGBoost Teroptimasi	0,994 (0,986-1,000)	0,987	0,988	0,985	0,982

Pada skenario standar, kinerja model menunjukkan perbedaan karakteristik antara model linier dan model *ensemble learning* berbasis pohon. Model *Logistic Regression* (LR) standar menghasilkan *recall* 0,806 dan *precision* 0,837, serta jumlah kesalahan sebesar 47 *False Negative* (FN) dan 38 *False Positive* (FP). Angka ini berisiko tinggi secara klinis karena meloloskan pasien kritis, meskipun nilai ROC-AUC cukup baik sebesar 0,891 (95% CI: 0,860-0,919). Rentang interval kepercayaan (CI) yang lebar ini mengonfirmasi tingginya ketidakpastian prediksi model linier. Sebaliknya, model *Random Forest* (RF) standar dan *XGBoost* standar langsung mendominasi dalam menangkap hubungan nonlinear yang kompleks. RF standar mencetak *recall* 0,996 dan *precision* 0,992 (1 FN, 2 FP), sedangkan *XGBoost* standar meraih *recall* sempurna 1,000 dan *precision* 0,984 (4 FP). Nilai ROC-AUC RF dan *XGBoost* sangat tinggi, yaitu 0,997 (95% CI: 0,992–1,000) untuk RF dan 0,996 (95% CI: 0,988–1,000) untuk *XGBoost*. Rentang interval yang sangat sempit ini membuktikan stabilitas dan keandalan prediksi kedua model *ensemble* secara statistik.

Pengaplikasian teknik SMOTE memicu efek *trade-off* terutama pada LR. Pada model LR, SMOTE memicu ketidakseimbangan performa *confusion matrix* yang tajam; *precision* meningkat signifikan menjadi 0,953 (FP turun menjadi 10), namun *recall* menurun drastis hingga 0,665 dengan lonjakan kesalahan prediksi yang fatal sebanyak 81 FN. Secara sistematis, kondisi ini terjadi karena kelas minoritas dalam dataset adalah kelas negatif IMA (kelas 0). Pembuatan data sintetis pada kelas negatif menggeser batas keputusan (*decision boundary*) model linier menjadi sangat konservatif dalam menetapkan status positif IMA. Fenomena ini membuktikan bahwa peningkatan ROC-AUC (menjadi 0,899) tidak menjamin keamanan klinis model pada skenario nyata. Sebaliknya, model RF dan *XGBoost* terbukti sangat tangguh (*robust*). Skenario RF SMOTE memberikan peningkatan performa optimal dengan mencetak *Recall* sempurna 1,000, a*ccuracy* 0,995, dan ROC-AUC tertinggi 0,999 pada rentang CI paling ketat (0,999–1,000), serta memangkas jumlah kesalahan menjadi hanya 2 FP. Di sisi lain, kinerja *XGBoost* SMOTE terpantau stabil pada seluruh metrik utama dengan kenaikan tipis nilai ROC-AUC menjadi 0,997, menegaskan efektivitas SMOTE dalam memperjelas batas pemisah pada algoritma *ensemble*.

Hyperparameter tuning menggunakan *Randomized Search Cross-Validation* mampu memperbaiki kelemahan batas keputusan linier LR sebelumnya. Pada model LR teroptimasi, masalah *trade-off* berhasil diatasi, *recall* naik menjadi 0,711, *F1-score* 0,809, dan *accuracy* 0,793, dengan jumlah kesalahan menjadi 70 FN dan 11 FP. Kemampuan LR teroptimasi dalam memisahkan kelas juga meningkat dengan ROC-AUC 0,902 dan interval CI yang menyempit pada (0,872–0,929). Meskipun model LR mengalami perbaikan, kinerja akhirnya masih tidak mampu mengungguli model *ensemble*. Model RF teroptimasi menunjukkan sedikit fluktuasi minor pada jumlah kesalahan sebanyak 1 FN dan 2 FP, dengan ROC-AUC stabil di angka 0,999 (95% CI: 0,997–1,000). Hal ini mengindikasikan bahwa konfigurasi *default* bawaan RF telah mendekati titik optimal. Di sisi lain, model *XGBoost* teroptimasi menunjukkan karakteristik yang sangat kokoh dengan metrik *confusion matrix* yang identik termasuk *recall* sempurna 1,000 dengan 4 FP di seluruh skenario, meskipun nilai ROC-AUC sedikit berubah menjadi 0,994 (95% CI: 0,981–1,000). Fenomena ini menegaskan bahwa *XGBoost* memiliki ketahanan tinggi terhadap intervensi data sintetis hasil SMOTE maupun perubahan parameter.

Secara keseluruhan, model *ensemble learning* seperti RF dan *XGBoost* secara konsisten mengungguli kinerja model linear seperti LR di seluruh metrik evaluasi pada setiap skenario. Hal ini menunjukkan bahwa RF dan *XGBoost* lebih efektif dan adaptif dalam mempelajari dan menangkap pola kompleks yang nonlinear pada data medis IMA dibandingkan LR. Melalui analisis komparatif yang telah dilakukan sebelumnya, model RF terutama RF SMOTE serta *XGBoost* disimpulkan sebagai model terbaik dalam klasifikasi IMA. *XGBoost* unggul berdasarkan stabilitas kinerja dan *recall* yang sempurna, yaitu nol kesalahan prediksi pada *False Negative* (FN). Sedangkan RF SMOTE unggul berdasarkan ROC-AUC yang mencapai nilai tertinggi 0,999 dengan lebar interval CI 95% paling sempit serta jumlah kesalahan prediksi paling minimum di antara seluruh model klasifikasi, yaitu sebanyak 2 *False Positive* (FP) tanpa FN.

SIMPULAN

Berdasarkan hasil studi komparatif yang telah dilakukan, diperoleh bahwa model *ensamble learning* berbasis pohon, seperti *Random Forest* (RF) dan *Extreme Gradient Boosting* (*XGBoost*), memiliki keunggulan dibandingkan model linear seperti *Logistic Regression* (LR) dalam menangkap hubungan antavariabel yang nonlinear dan kompleks pada data klinis seperti Infark Miokard Akut (IMA). Pengaplikasian *Synthetic Minority Oversampling Technique* (SMOTE) serta *hyperparameter tuning* dengan *Randomized Search Cross-Validation* terbukti

mempengaruhi kinerja model klasifikasi IMA. Pada model LR, diperlukan tahap *hyperparameter tuning* agar tidak terdistorsi oleh data sintetis SMOTE. Sebaliknya, RF dan *XGBoost* menunjukkan ketahanan komputasional dan kestabilan yang tinggi sejak awal. Oleh karena itu, pada masalah klinis seperti klasifikasi IMA yang mengutamakan keselamatan pasien, kedua algoritma ini ditetapkan sebagai model terbaik. *XGBoost* unggul berdasarkan stabilitas kinerja, sementara *Random Forest* lebih andal dari segi minimasi kesalahan prediksi.

Model *Random Forest* sangat andal untuk diintegrasikan dalam pengembangan sistem pendukung keputusan klinis, sehingga dapat membantu tenaga medis dalam deteksi dini berbasis data rekam medis dan menekan angka mortalitas akibat keterlambatan penanganan. Pada penelitian selanjutnya, direkomendasikan untuk menggunakan dataset yang lebih besar, beragam, atau data dengan citra medis seperti CT *Scan*, MRI, dan elektrokardiogram (EKG), sehingga menghasilkan model yang lebih akurat dan kuat dalam kasus IMA atau penyakit kardiovaskular lainnya. Model dapat lebih dikembangkan dengan mengeksplorasi algoritma lain atau melakukan *hybrid* beberapa algoritma agar diperoleh kinerja model yang lebih optimal, seperti CNN, SVM, LightGBM, dll. penggunaan teknik selain SMOTE dalam menangani ketidakseimbangan data, seperti *Random Oversampling* (ROS), ROS-SMOTE, atau SMOTE dengan *Edited Nearest Neighbors* (SMOTEENN). Selain itu, menggunakan metode *hyperparameter tuning* lain seperti *Grid Search*, *Bayesian Optimization*, atau *Optuna*, serta menambahkan analisis *feature importance* untuk mengetahui variabel yang paling berpengaruh dalam prediksi IMA dan menjadi wawasan yang lebih mendalam.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing atas arahan, bimbingan, dan ilmu yang diberikan selama proses penelitian ini. Terima kasih juga kepada rekan-rekan mahasiswa yang telah memberi dukungan dan motivasi. Selain itu, terima kasih penulis tujukan kepada Mendeley Data selaku penyedia dataset yang digunakan dalam penelitian ini. Semoga seluruh kontribusi dan bantuan yang telah diberikan oleh berbagai pihak dapat memberikan manfaat dalam perkembangan ilmu pengetahuan.

DAFTAR PUSTAKA

Amaliah, R., Yaswir, R., & Prihandani, T. (2019). Gambaran homosistein pada pasien infark miokard akut di RSUP Dr. M. Djamil Padang. *Jurnal Kesehatan Andalas*, *8*(2), 351-355. https://doi.org/10.25077/jka.v8i2.1012

Asshidiq, L., Wisudawan, & Deus, T. (2025). Infark Miokard di Indonesia: Tinjauan Literatur Terbaru terhadap Faktor Risiko, Karakteristik Klinis, Manajemen Keperawatan, dan Prediktor Prognostik. *Jurnal Riset Rumpun Ilmu Kedokteran*, *4*(3), 87–95. https://doi.org/10.55606/jurrike.v4i3.6653

Azhar, I. S. B., & Sari, W. K. (2022). Penerapan Data Mining Dan Tekonologi Machine Learning Pada Klasifikasi Penyakit Jantung. *JSI: Jurnal Sistem Informasi (E-Journal)*, *14*(1), 2560-2568. https://doi.org/10.18495/jsi.v14i1.65

CardioSmart. (2015, September 1). *Ventricular Tachycardia*. Retrieved from CardioSmart American College of Cardiology: https://www.cardiosmart.org/topics/ventricular-tachycardia

Chaudhry, A. N. (2024, Juni 20). *Logistic Regression Implementation From Scratch—A Step By Step Approach*. Retrieved from Medium: https://medium.com/@chaudhryalinaeem/logistic-regression-implementation-from-scratch-a-step-by-step-approach-69e7e3ee866b

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forests. In C. Zhang & Y. Ma (Eds.). *Ensemble Machine Learning*. New York: Springer New York, pp. 157-175. https://doi.org/10.1007/978-1-4419-9326-7_5

Davidson, D. M. (1994). An introduction to cardiovascular disease. In Social support and cardiovascular disease (pp. 3-19). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4899-2572-5_1

Fajriyah, R., Isnandar, H. A., & Arifuddin, A. (2024). Gene Markers Identification Of Acute Myocardial Infarction Disease Based On Genomic Profiling Through Extreme Gradient Boosting (XGBoost). *Media Statistika*, *17*(1), 69-80. https://doi.org/10.14710/medstat.17.1.69-80

Han, S., Kim, H., & Lee, Y. S. (2020). Double random forest. *Machine Learning*, *109*(8), 1569-1586. https://doi.org/10.1007/s10994-020-05889-1

Kumar, V. (2021). Evaluation of computationally intelligent techniques for breast cancer diagnosis. *Neural Computing and Applications*, *33*(8), 3195-3208. https://doi.org/10.1007/s00521-020-05204-y

Kurnia, S., Nurdin, & Khaidar, A. (2025). Perbandingan Metode Machine Learning Menggunakan Metode Support Vector Machine dan Artificial Neural Network dalam Memprediksi Serangan Jantung. *Jurnal Informatika Kaputama (JIK)*, *9*(2), 87-94. https://doi.org/10.59697/jik.v9i2.1020

Mandias, G. F., & Manoppo, I. J. (2025). Penerapan Model Machine Learning untuk Memprediksi Serangan Jantung Dini. *Jurnal Ilmiah Matrik*, *27*(2), 193-201. https://doi.org/10.33557/2cg02a51

Mechanic, O. J., Gavin, M., Grossman, S. A. (2023). Acute myocardial infarction. In *StatPearls [Internet]*. Treasure Island (FL): StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK459269/

Omarzai, F. (2024, Juli 21). *XGBoost Classification in Depth*. Retrieved from Medium: https://medium.com/@fraidoonomarzai99/xgboost-classification-in-depth-979f11ef4bf9

Pangaribuan, J. J., Tanjaya, H., & Kenichi, K. (2021). Mendeteksi penyakit jantung menggunakan machine learning dengan algoritma logistic regression. *Journal Information System Development (ISD)*, *6*(2), 1-10.

Prasetya, F. P., & Rosa, P. P. (2024). Klasifikasi Kegagalan Pembayaran Kredit Nasabah Bank dengan Algoritma XGBoost. In *Seminar Nasional Informatika Bela Negara (SANTIKA)* (Vol. 4, No. 1, pp. 366-371).

Raharjo, F. A. A. D., Maghfiroh, I. L., & Rokhman, A. (2025). Faktor-faktor resiko serangan infark miokard akut: Risk factors for acute myocardial infarction attack. *Jurnal Ilmiah Keperawatan (Scientific Journal of Nursing)*, *11*(3), 563-572. https://doi.org/10.33023/jikep.v11i3.2769

Rahayu, W., Jollyta, D., Hajjah, A., Johan, Gusrianty, Gustientiedina, . . . Desnelita, Y. (2024). Synthetic minority oversampling technique (SMOTE) for boosting the accuracy of C4. 5 algorithm model. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, *3*(3), 624-630. https://doi.org/10.59934/jaiea.v3i3.469

Rahmada, A., & Susanto, E. R. (2024). Peningkatan akurasi prediksi penyakit jantung dengan teknik SMOTEENN pada algoritma random forest. *Jurnal Pendidikan dan Teknologi Indonesia (JPTI)*, *4*(12), 795-803. https://doi.org/10.52436/1.jpti.524

Rashid, T. A., & Hassan, B. (2022, November 23). *Heart Attack Dataset*. Mendeley Data, V1. https://doi.org/10.17632/wmhctcrt5v.1

Reátegui, R., Tandazo-Malla, C., Suárez, R., & Ramírez-Cerna, L. (2025). Cardiovascular risk prediction via ensemble machine learning and oversampling methods. *Scientific Reports*, **15**, 43576. https://doi.org/10.1038/s41598-025-30895-5

Rhomadon, M. F., Wydyanto, Mirza, A. H., & Huda, N. (2025). Penerapan Algoritma Logistic Regression Untuk Memprediksi Penyakit Jantung. *Jurnal Informatika dan Tekonologi Komputer (JITEK)*, *5*(3), 133–147. https://doi.org/10.55606/jitek.v5i3.8105

Salman, I. (2019). Heart attack mortality prediction: an application of machine learning methods. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(6), 4378-4389. https://doi.org/10.3906/elk-1811-4

Song, X., Liu, X., Liu, F., & Wang, C. (2021). Comparison of machine learning and logistic regression models in predicting acute kidney injury: a systematic review and meta-analysis. *International journal of medical informatics*, *151*, 104484. https://doi.org/10.1016/j.ijmedinf.2021.104484

Tajali, A., Saragih, T. H., Mazdadi, M. I., Budiman, I., & Farmadi, A. (2024). The Impactness of SMOTE as Imbalance Class Handling for Myocardial Infarction Complication Classification using Machine Learning Approach with Data Imputation and Hyperparameter. *Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics*, *6*(4), 227-239. https://doi.org/10.35882/ijeeemi.v6i4.13

WHO (World Health Organization). (2025, Juli 31). *Cardiovascular Diseases (CVDs)*. Retrieved from World Health Organization: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Widodo, S., Brawijaya, H., & Samudi, S. (2022). Stratified K-fold cross validation optimization on machine learning for prediction. *Sinkron: jurnal dan penelitian teknik informatika*, *6*(4), 2407-2414. https://doi.org/10.33395/sinkron.v7i4.11792