J-Rapa (Jurnal Rekayasa Perangkat Lunak)
Vol 03, No. 01, Bulan 03 Tahun 2024, pp. 27-37
ISSN: 3046-7810

PREDIKSI PENYAKIT SERANGAN JANTUNG MENGGUNAKAN SMOTE DAN TUNING HYPERPARAMETER

Alfina Rahma Aulia [1], Zulkifli [2], Ferly Ardhy [3], Agustinus Eko Setiawan [4]
[1]*Faculty of Technology and Informatics, Universitas Aisyah Pringsewu, Jalan A Yani, 1 A Tambak Rejo, Pringsewu, Lampung, Indonesia*
[2] *Faculty of Technology and Informatics, Universitas Aisyah Pringsewu, Jalan A Yani, 1 A Tambak Rejo, Pringsewu, Lampung, Indonesia*

Article Info

Keywords:

Decision Tree;
Heart attack prediction;
Hyperparameter Optimization;
SMOTE.

Abstract

Heart attack is one of the leading causes of death worldwide, including in Indonesia, making early prediction essential for effective prevention. One of the main challenges in heart attack prediction modeling is the imbalanced dataset, where negative cases significantly outnumber positive cases, leading to biased classification results. This study aims to improve the performance of heart attack prediction models by addressing class imbalance and optimizing model parameters. The proposed methodology applies the Synthetic Minority Over Sampling Technique (SMOTE) to balance the dataset and utilizes GridSearchCV for hyperparameter optimization on three classification algorithms, namely K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Decision Tree. The dataset used in this study was obtained from Kaggle and consists of 1,319 patient records. Experimental results indicate that data balancing and parameter tuning significantly enhance model performance. Among the evaluated algorithms, the Decision Tree model achieved the best results, with accuracy, precision, recall, and F1-score reaching 98%, outperforming SVM with 92.93% accuracy and KNN with 65.15%. Confusion matrix analysis shows that the Decision Tree model effectively reduces misclassification, particularly for positive cases, which is crucial for early identification of patients at risk of heart attack. In conclusion, the integration of SMOTE and hyperparameter optimization improves prediction accuracy, and the Decision Tree algorithm is identified as the most effective model for heart attack prediction.

Kata Kunci:

Decision Tree;
Optimasi Hyperparameter;
Prediksi serangan jantung;
SMOTE.

Abstrak

Penyakit serangan jantung merupakan salah satu penyebab utama kematian di dunia, termasuk di Indonesia, sehingga prediksi dini menjadi sangat penting sebagai upaya pencegahan. Tantangan utama dalam pemodelan prediksi serangan jantung adalah distribusi data yang tidak seimbang, di mana jumlah kasus negatif jauh lebih banyak dibandingkan kasus positif, sehingga model cenderung menghasilkan prediksi yang bias. Penelitian ini bertujuan untuk meningkatkan akurasi prediksi serangan jantung dengan menangani permasalahan ketidakseimbangan data dan melakukan optimasi parameter model. Metode yang digunakan adalah penerapan Synthetic Minority Over Sampling Technique (SMOTE) untuk menyeimbangkan kelas data serta GridSearchCV untuk optimasi hyperparameter pada tiga algoritma klasifikasi, yaitu K-Nearest Neighbor (KNN), Support Vector Machine (SVM), dan Decision Tree. Dataset yang digunakan berasal dari Kaggle dengan total 1.319 data pasien. Hasil eksperimen menunjukkan bahwa setelah proses balancing dan tuning, algoritma Decision Tree memberikan performa terbaik dengan nilai akurasi, precision, recall, dan F1-score mencapai 98%, mengungguli SVM sebesar 92,93% dan KNN sebesar 65,15%. Analisis confusion matrix menunjukkan bahwa Decision Tree mampu menekan kesalahan prediksi terutama pada kelas positif, sehingga mendukung deteksi dini pasien berisiko serangan jantung. Dengan demikian, integrasi SMOTE dan optimasi hyperparameter terbukti efektif meningkatkan kinerja model prediksi serangan jantung..

Penulis Korespondensi:
Alfina Rahma Aulia
Faculty of Technology and Informatics, Universitas Aisyah Pringsewu

1. PENDAHULUAN

Serangan jantung merupakan salah satu masalah *kardiovaskular* yang sangat serius karena terjadi ketika pasokan darah ke otot jantung terhambat, sehingga mengurangi kemampuan jantung dalam mendistribusikan darah ke seluruh tubuh [1]. Kondisi ini dapat menyebabkan kerusakan permanen pada jaringan jantung dan berujung pada kematian apabila tidak ditangani secara cepat dan tepat. Berdasarkan laporan *Global Burden of Disease (GBD)* dan *Institute for Health Metrics and Evaluation (IHME)*, penyakit jantung tercatat sebagai penyebab kematian tertinggi di Indonesia pada periode 2014–2019. Data dari *World Health Organization (WHO)* juga menunjukkan bahwa penyakit *kardiovaskular* menyebabkan sekitar 17,9 juta kematian setiap tahun di seluruh dunia, atau hampir sepertiga dari total

kematian global [2]. Di Indonesia, hasil Survei Kesehatan Rumah Tangga Nasional (SKRTN) memperlihatkan peningkatan signifikan angka kematian akibat Penyakit Jantung Koroner (PJK), dari 16% pada tahun 1991 menjadi 36% pada tahun 2014, setara dengan 53,5 kematian per 100.000 penduduk [3]. Fakta tersebut menegaskan bahwa serangan jantung merupakan ancaman nyata bagi kesehatan masyarakat dan membutuhkan strategi deteksi dini serta pencegahan yang lebih efektif.

Perkembangan teknologi informasi dan komputasi, khususnya dalam bidang *data science*, memberikan peluang besar dalam mendukung proses diagnosis dan prediksi penyakit. Penerapan teknik *data mining* dan *machine learning* memungkinkan analisis data medis secara otomatis untuk menemukan pola-pola tersembunyi yang sulit diidentifikasi melalui metode konvensional [4], [5]. Pemanfaatan algoritma pembelajaran mesin dalam bidang medis tidak hanya berpotensi meningkatkan akurasi diagnosis, tetapi juga membantu mempercepat proses pengambilan keputusan klinis. Beberapa algoritma yang banyak digunakan dalam penelitian prediksi penyakit jantung antara lain *K-Nearest Neighbor (KNN), Support Vector Machine (SVM)*, dan *Decision Tree*. Algoritma *KNN* dikenal sederhana namun efektif dalam melakukan klasifikasi [6], *SVM* mampu memisahkan kelas data menggunakan *hyperplane optimal* [7], sedangkan *Decision Tree* menghasilkan model yang mudah diinterpretasikan serta memberikan gambaran visual mengenai faktor-faktor yang memengaruhi hasil klasifikasi [8].

Sejumlah penelitian terdahulu menunjukkan bahwa algoritma-algoritma tersebut memiliki performa yang cukup tinggi dalam memprediksi serangan jantung. Metode *KNN* dilaporkan mampu mencapai akurasi hingga 96,67% [9], sementara *SVM* memperoleh akurasi 100% pada data uji dan 96% pada data latih [10]. Di sisi lain, algoritma *Decision Tree* menghasilkan tingkat akurasi sebesar 93,44% [11]. Meskipun demikian, sebagian besar penelitian sebelumnya masih menghadapi permasalahan utama berupa ketidakseimbangan data *(imbalanced data)* antara kelas positif dan negatif. Kondisi ini menyebabkan model cenderung bias terhadap kelas mayoritas, sehingga kemampuan prediksi terhadap kelas minoritas menjadi kurang optimal.

Pada penelitian ini, dataset yang digunakan bersumber dari *Kaggle* dengan total 1.319 data pasien dan sembilan atribut klinis, yaitu usia, jenis kelamin, detak jantung, tekanan darah sistolik, tekanan darah diastolik, kadar gula darah, enzim CK-MB, dan Troponin. Label diagnosis dibagi ke dalam dua kelas, yaitu positif (1) untuk pasien yang mengalami serangan jantung dan negatif (0) untuk pasien yang tidak mengalaminya. Distribusi data menunjukkan adanya ketidakseimbangan kelas, dengan 810 data negatif dan 509 data positif. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan *Synthetic Minority Over-sampling Technique (SMOTE)*, yaitu teknik *oversampling* yang menghasilkan data sintetis pada kelas minoritas berdasarkan kedekatan antar data, sehingga distribusi kelas menjadi lebih seimbang dan model dapat mempelajari pola dari kedua kelas secara lebih adil [12].

Selain penyeimbangan data, penelitian ini juga menerapkan proses *tuning hyperparameter* menggunakan *GridSearchCV* pada masing-masing algoritma. *Tuning hyperparameter* bertujuan untuk memperoleh kombinasi parameter terbaik yang mampu meningkatkan performa model. *GridSearchCV* bekerja dengan menguji berbagai kombinasi parameter dan mengevaluasinya menggunakan teknik *cross-validation*, sehingga risiko *overfitting* dapat diminimalkan dan model yang dihasilkan lebih general terhadap data baru [13].

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk membandingkan performa algoritma *KNN, SVM*, dan *Decision Tree* dalam memprediksi risiko serangan jantung dengan memanfaatkan teknik *SMOTE* dan *tuning hyperparameter* sebagai upaya peningkatan kinerja model. Kebaruan penelitian ini terletak pada analisis komprehensif perbandingan performa ketiga algoritma setelah penyeimbangan data dan optimasi parameter, serta evaluasi yang menekankan kesesuaian antara hasil pada data latih dan data uji. Hasil penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem prediksi medis yang lebih akurat dan aplikatif, sehingga dapat digunakan sebagai alat bantu bagi tenaga medis dalam mendeteksi risiko serangan jantung secara dini.

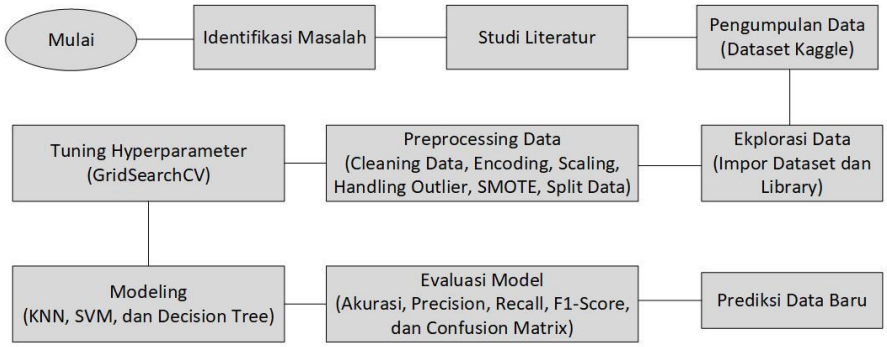
2. METODE

Penelitian ini menggunakan metode dokumentatif dengan data sekunder berupa *Heart Attack Dataset* dari *Kaggle* yang berisi 1.319 data pasien dengan sembilan atribut klinis, termasuk CK-MB dan Troponin sebagai penanda biokimia utama serangan jantung. Proses penelitian meliputi eksplorasi, *preprocessing*, pembagian data latih dan uji, penerapan *SMOTE* untuk menyeimbangkan kelas, serta *tuning hyperparameter* menggunakan *GridSearchCV*. Analisis dilakukan menggunakan *Python* di *Google Colab* dengan dukungan library *Pandas, NumPy, Scikit-learn*, dan *Matplotlib* untuk pemodelan serta visualisasi hasil.

A. Tahapan Penelitian

Penelitian ini menggunakan *Heart Attack Dataset* dari *Kaggle* yang diproses melalui *tahapan preprocessing (handling missing values, outlier, SMOTE*, dan normalisasi) serta *tuning hyperparameter* dengan *GridSearchCV*. Tiga algoritma klasifikasi, yaitu *KNN, SVM*, dan *Decision Tree*, kemudian diuji dan dievaluasi menggunakan akurasi, presisi,

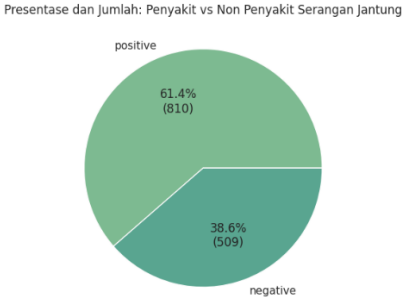
recall, F1-score, serta *confusion matrix*, di mana model terbaik digunakan untuk prediksi data baru. Rancangan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahap Penelitian

B. *Dataset*

Data yang digunakan bersumber dari *Kaggle* dengan judul *Heart Attack Dataset* yang terdiri dari 1.319 data pasien dengan sembilan atribut klinis, antara lain usia, jenis kelamin, detak jantung, tekanan darah, kadar gula darah, enzim CK-MB, Troponin, serta label diagnosis (positif = 1, negatif = 0). Dataset ini memiliki ketidakseimbangan kelas, yaitu 61,4% (810) pasien negatif dan 38,6% (509) positif, sehingga diperlukan *SMOTE* untuk menyeimbangkan data. Data ditunjukkan pada Gambar 6.



Gambar 2. Visualisasi Data

C. *Preprocessing*

Tahapan preprocessing meliputi:

1. *Handling Missing Values Dan Data Duplikat*

Hasilnya menunjukkan bahwa dataset tidak memiliki nilai kosong pada seluruh atribut sehingga dapat langsung digunakan untuk analisis. Tampilan pada Gambar 3.

```
df.isnull().sum() #done
```

	0
Age	0
Gender	0
Heart rate	0
Systolic blood pressure	0
Diastolic blood pressure	0
Blood sugar	0
CK-MB	0
Troponin	0
Result	0

Gambar 3. Cek Missing Value

Selain itu, dilakukan juga pengecekan data duplikat dan diperoleh hasil bahwa tidak terdapat data ganda dalam dataset, sehingga kualitas data terjaga dan layak digunakan dalam proses penelitian.

```
[ ]  df.duplicated().sum()

     np.int64(0)
```

Gambar 4. Cek Data Duplikat

2. *Encoding Variabel Kategorikal*

Proses *encoding* untuk mengubah data kategorikal menjadi bentuk numerik agar dapat diproses oleh algoritma *machine learning*. Pada penelitian ini digunakan *LabelEncoder* dengan cara membuat objek *encoder* dan menerapkannya pada kolom *Result*, sehingga label kategori pada atribut target berhasil diubah menjadi representasi angka.

```
encoder = LabelEncoder()
df['Result'] = encoder.fit_transform(df['Result'])
```

Gambar 5. Label Encoder

3. *Normalisasi Dengan Standard Scaler*

Normalisasi data dilakukan dengan *StandardScaler* agar semua fitur memiliki rata-rata nol dan standar deviasi satu, diterapkan pada data latih *(fit_transform)* dan data uji *(transform)* untuk menjaga konsistensi skala. Normalisasi data dilakukan menggunakan *Standard Scaler* untuk menyamakan skala fitur dengan *mean* = 0 dan *standar deviasi* = 1. Parameter dihitung dari data latih hasil *SMOTE*, lalu digunakan untuk mentransformasi data latih dan uji agar seluruh fitur berada pada skala yang sama dan model dapat belajar lebih optimal.

```
[19] scaler = StandardScaler()
     X_train_scaled = scaler.fit_transform(X_train_smote)
     X_test_scaled = scaler.transform(X_test)
```

Gambar 6. Normalisasi Data

4. *Penanganan Outlier Menggunakan Z-Score*

Kemudian dilakukan tahap *Handling Outlier* dilakukan untuk mengurangi pengaruh data ekstrem yang dapat mengganggu kinerja model. Outlier diidentifikasi menggunakan metode *Z-Score* (>3), dan hasilnya terdapat 114 data terdeteksi sebagai *outlier*. Gambar 8. menunjukkan hasil pengecekkan data *outlier*.

```
from scipy.stats import zscore
import pandas as pd
X_df = pd.DataFrame(X)
z_scores = np.abs(zscore(X_df))
outlier_mask = (z_scores > 3)
X_outliers = X_df[(outlier_mask).any(axis=1)]
print(f"Jumlah outlier: {X_outliers.shape[0]}")

Jumlah outlier: 114
```

Gambar 7. Handling Outlier

Setelah proses deteksi, data *outlier* yang ditemukan kemudian dihapus agar tidak memengaruhi hasil pemodelan. Setelah tahap ini, jumlah data berkurang dari 1.319 baris menjadi 1.205 baris dengan 8 kolom, yang berarti sebanyak 114 baris data *outlier* telah berhasil dihapus.

```
mask_bukan_outlier = ~(outlier_mask.any(axis=1))
X_clean = X_df[mask_bukan_outlier]
y_clean = y[mask_bukan_outlier]
print(f"Data setelah hapus outlier: {X_clean.shape}")

Data setelah hapus outlier: (1205, 8)
```

Gambar 8. Hapus Outlier

5. *Balancing Data Dengan SMOTE*

Setelah *handling outlier*, dataset dibagi 70% latih dan 30% uji. Pada data latih, distribusi kelas tidak seimbang dengan 505 data positif dan 338 data negatif. Untuk mengatasi hal ini diterapkan *SMOTE* sehingga jumlah data latih meningkat dari 843 menjadi 1.406 dengan distribusi seimbang, masing-masing 505 positif dan 505 negatif. Dengan proporsi yang seimbang, model dapat mempelajari pola dari kedua kelas secara lebih optimal.

D. Algoritma

Tiga algoritma utama digunakan, yaitu *KNN, SVM*, dan *Decision Tree. KNN* dipilih karena kesederhanaannya, *SVM* karena performanya dalam klasifikasi dua kelas, dan *Decision Tree* karena kemudahan interpretasi. Untuk memperoleh hasil optimal, dilakukan tuning hyperparameter menggunakan *GridSearchCV.*

1. *K-Nearest Neighbor (KNN)* merupakan algoritma yang bekerja dengan mencari kesamaan antara data baru dan data yang sudah ada sebelumnya. Setelah menemukan kemiripan tersebut, *KNN* akan menempatkan data baru ke dalam kelompok yang paling mendekati kategori dari data yang sudah tersedia [14]. *KNN* relatif sederhana, tetapi performanya sangat dipengaruhi oleh pemilihan nilai k dan metode pengukuran jarak. *K-Nearest Neighbor (KNN)* adalah algoritma *non-parametrik* yang bekerja dengan membandingkan data baru terhadap data latih berdasarkan jarak terdekat.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

Dimana:
x adalah data yang digunakan untuk pengujian model
y adalah data yang digunakan untuk melatih model
n adalah total jumlah fitur atau atribut pada dataset

2. *Support Vector Machines (SVM)* merupakan algoritma yang berfungsi untuk membedakan dua kelas dengan memilih *hyperplane* yang memiliki margin paling lebar terhadap data poin paling dekat dari tiap kelas. Banyaknya fitur dalam data input menentukan apakah *hyperplane* tersebut berbentuk garis dalam ruang dua dimensi atau bidang dalam ruang berdimensi lebih tinggi [15]. *SVM* efektif pada data berdimensi tinggi dan mampu menangani pemisahan linier maupun non-linier. Rumus dasar pemisahan *hyperplane* dapat dituliskan sebagai berikut:

$$f(x) = w^T x + b \qquad (2)$$

Dimana:
x adalah vektor fitur (fitur input)
w adalah vektor bobot
b adalah bias
f(x) = 0 adalah hyperplane

3. Algoritma *Decision Tree* memanfaatkan struktur pohon untuk menganalisis data dan mempermudah pengambilan keputusan. Setiap *node* merepresentasikan pengujian fitur, cabang menunjukkan hasil pengujian, dan leaf node menggambarkan kelas akhir. *Root node* berfungsi sebagai *classifier* utama yang mengorganisir kelas-kelas dalam data [16]. Keunggulannya mudah dipahami dan mampu menggambarkan hubungan antarvariabel. Salah satu metode pemisahan dalam *Decision Tree* adalah menggunakan *Information Gain*, yaitu pengurangan entropi sebelum dan sesudah suatu atribut digunakan untuk membagi data:

$$\text{Information Gain (IG)} = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (3)$$

Dimana:
S adalah seluruh dataset,
Si adalah subset ke-i setelah pemisahan berdasarkan atribut tertentu,
Entropy(S) adalah ukuran ketidakpastian pada dataset.

4. *Tuning hyperparameter* merupakan proses pengaturan parameter sebelum pelatihan agar kinerja model optimal. Pada penelitian ini digunakan metode *GridSearchCV* yang secara sistematis mencari kombinasi parameter terbaik melalui *cross-validation*, sehingga mampu menghindari *overfitting* dan menghasilkan performa akurat [17].

E. Evaluasi

Model dievaluasi dengan metrik akurasi, *precision, recall, F1-Score*, serta *confusion matrix*. Pengujian dilakukan pada data *training* dan data *testing* untuk memastikan performa model stabil.

1. Akurasi

Akurasi adalah metrik penilaian yang menunjukkan tingkat ketepatan model dalam menghasilkan prediksi yang benar. Hasilnya menggambarkan persentase prediksi benar dari keseluruhan data yang diuji, data kelas positif dan negatif [18].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Keterangan:

- TP (*True Positive*) : Data positif yang berhasil diprediksi dengan benar.
- TN (*True Negative*) : Data negatif yang tepat diprediksi sebagai negatif.
- FP (*False Positive*) : Data negatif namun diprediksi sebagai positif.
- FN (*False Negative*) : Data positif tetapi salah diprediksi negatif oleh model.

2. *Precision*

Precision merupakan ukuran yang menunjukkan seberapa tepat model dalam memprediksi data positif. *Precision* adalah perbandingan antara jumlah rasio prediksi *true positive* dengan keseluruhan prediksi positif yang diklasifikasikan sebagai positif oleh model (*true positive + false positive)* [19].

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

Keterangan:

- TP: Prediksi benar terhadap kasus yang sebenarnya positif
- FP: Prediksi positif terhadap kasus yang sebenarnya negatif

3. *F1-Score*

F1 Score adalah rata-rata harmonis dari *precision* dan *recall* yang digunakan untuk menilai keseimbangan antara ketepatan prediksi positif dan kemampuan menemukan seluruh data positif [20].

Rumus F1-Score:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (6)$$

Dimana:

- *Precision* : Kecakapan model dalam memprediksi seluruh kasus positif
- *Recall* : Kecakapan model dalam mengenali seluruh kasus positif.

4. *Confusion Matrix*

Confusion Matrix merupakan metriks berbentuk tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan memperlihatkan jumlah prediksi yang benar dan salah terhadap kelas positif dan negatif. Matriks terdiri dari empat komponen utama yaitu *True Positive, True Negative, False Positive*, dan *False Negative*. Setiap kombinasi merepresentasikan keluaran klasifikasi yang benar maupun salah. *Confusion matrix* memperlihatkan secara spesifik kesalahan model, baik dalam memprediksi kelas positif maupun negatif [21].

F. Data Baru

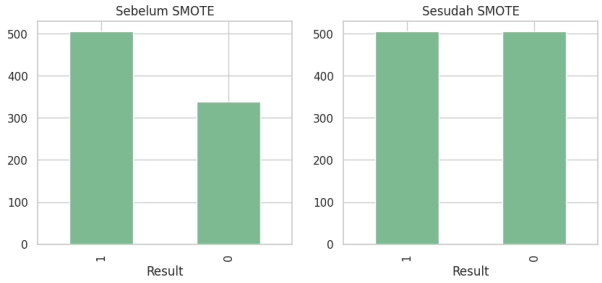
Melakukan prediksi data baru menggunakan model terbaik yang telah diperoleh, yaitu *Decision Tree*. Data pasien baru yang berisi atribut medis seperti usia, jenis kelamin, tekanan darah, kolesterol, detak jantung, CK-MB, dan Troponin diproses terlebih dahulu melalui tahapan *preprocessing* yang sama dengan data latih, termasuk normalisasi dan encoding variabel kategorikal. Setelah itu, data dimasukkan ke dalam model untuk diklasifikasikan ke dalam dua kategori, yaitu positif (berisiko serangan jantung) atau negatif (tidak berisiko serangan jantung).

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan membandingkan performa tiga algoritma *machine learning*, yaitu *K-Nearest Neighbor (KNN), Support Vector Machine (SVM),* dan *Decision Tree* dalam prediksi penyakit serangan jantung dengan penerapan *SMOTE* untuk penyeimbangan data dan *GridSearchCV* untuk *tuning hyperparameter*.

Pada tahap awal, data menunjukkan adanya ketidakseimbangan kelas, yaitu 810 data negatif dan 509 data positif. Ketidakseimbangan ini berpotensi menyebabkan bias model, terutama dalam mendeteksi kasus positif (pasien serangan

jantung). Oleh karena itu, dilakukan penerapan *SMOTE* sehingga jumlah data menjadi seimbang. Hasil *balancing* ini terbukti meningkatkan sensitivitas model terhadap kelas minoritas dengan menurunkan nilai *false negative* yang sangat penting dalam konteks medis. Hasil ditunjukkan pada Gambar 10.



Gambar 9. Hasil SMOTE

SMOTE diterapkan untuk mengatasi ketidakseimbangan kelas, dimana distribusi awal *data training* terdiri dari 843 sampel dengan 61,4% (810) data negatif dan 38,6% (509) data positif. Setelah *SMOTE*, jumlah sampel meningkat menjadi 1.406 dengan distribusi seimbang antara kelas positif (505) dan negatif (505). Teknik ini berhasil menyeimbangkan kelas sehingga model dapat belajar secara proporsional dari kedua kelas dan meningkatkan kinerja prediksi terhadap kelas minoritas. Tabel 2 menujukkan jumlah *data training* sebelum dan sesudah *SMOTE*.

TABEL 2
HASIL SMOTE

Keterangan	Positif (1)	Negatif (0)
Imbalance	505	338
Balancing	505	505

Setelah dilakukan *balancing*, tahap selanjutnya adalah *tuning hyperparameter* menggunakan *GridSearchCV*. Proses ini dilakukan untuk mencari parameter terbaik pada masing-masing algoritma. Hasil *tuning* menunjukkan adanya peningkatan performa, khususnya pada *SVM* dan *Decision Tree*, di mana kombinasi parameter yang tepat memberikan akurasi lebih stabil serta menekan kesalahan prediksi. Hasil *tuning hyperparameter* pada algoritma *KNN, SVM*, dan *Decision Tree* ditunjukkan pada Tabel 3

TABEL 3
TUNING HYPERPARAMETER

Algoritma	Parameter Optimal	Akurasi Cross-Validation
KNN	k = 11, Metode jarak = Manhattan	67,25%
SVM	C = 100, Kernel = Linear	84,38%
Decision Tree	max_depth = 4, min_samples_split = 2, min_samples_leaf = 5	Tinggi & stabil (tanpa overfitting)

Tahap modeling dilakukan untuk membangun model prediksi penyakit serangan jantung menggunakan *Decision Tree, KNN,* dan *SVM* dengan konfigurasi terbaik hasil *tuning hyperparameter*, kemudian dievaluasi untuk menilai akurasi dan efektivitasnya.

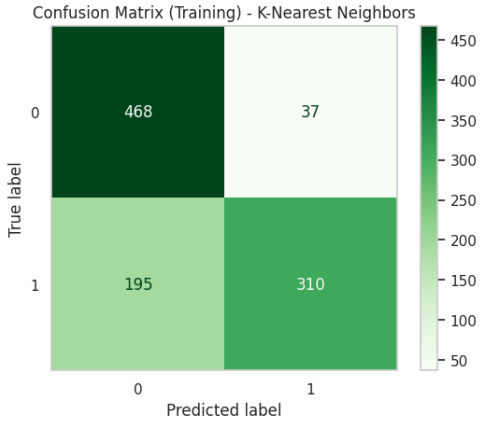
Evaluasi kinerja model dilakukan menggunakan akurasi, *precision, recall, f1-score*, dan *confusion matrix*. Pada *data training*, *Decision Tree* mencapai akurasi 98% dengan nilai *precision, recall*, dan *f1-score* yang konsisten tinggi. Sementara itu, *SVM* memperoleh akurasi sebesar 92,93% dan *KNN* hanya 65,15%. Hasil ini juga konsisten pada data testing, di mana *Decision Tree* tetap unggul, terutama dalam mendeteksi kelas positif. Analisis *confusion matrix* memperlihatkan bahwa *Decision Tree* mampu mengurangi kesalahan klasifikasi pada pasien positif, sehingga lebih andal untuk deteksi dini serangan jantung.

TABEL 4
HASIL EVALUASI MODEL

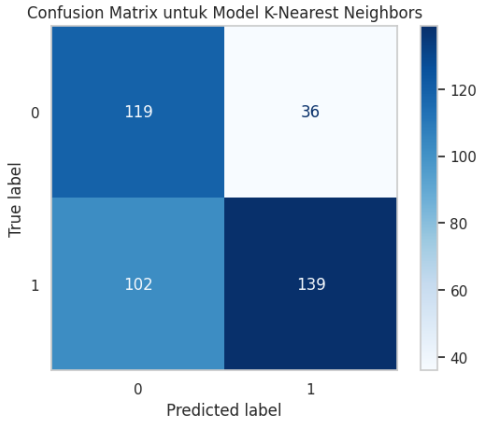
Model	Precision		Recall		F1-Score		Akurasi	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
KNN	0.7996	0.6799	0.7703	0.6271	0.7645	0.6269	77.03%	62.71%
SVM	0.9548	0.9448	0.9535	0.9392	0.9534	0.9397	95.35%	93.92%
Decision Tree	0.9960	0.9864	0.9960	0.9862	0.9960	0.9862	99.60%	98.62%

Tabel 4 merupakan tahap evaluasi model *machine learning* menggunakan akurasi, *precision, recall*, dan *f1-score* yang menunjukkan bahwa algoritma *Decision Tree* memiliki akurasi tertinggi baik pada *data training* maupun *testing*, diikuti oleh *Support Vector Machine (SVM),* sedangkan *K-Nearest Neighbor (KNN)* memiliki akurasi terendah.

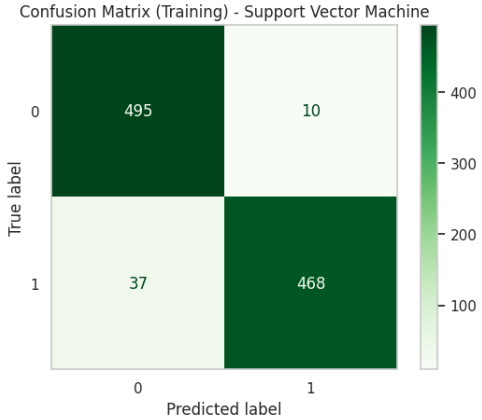
Confusion matrix digunakan untuk mengevaluasi kemampuan model dalam membedakan kelas positif dan negatif secara detail. Pada penelitian ini, *confusion matrix* disajikan baik pada *data training* maupun *testing* untuk memberikan gambaran menyeluruh terkait kinerja model. Pada *data training, confusion matrix* menunjukkan seberapa baik algoritma mengenali pola dari data yang digunakan untuk melatih model. Sedangkan pada *data testing, confusion matrix* digunakan untuk mengukur kemampuan generalisasi model terhadap data baru yang belum pernah dilihat. Berikut merupakan hasil dari perbandingan *confusion matrix* antara *data training* dan *data testing.*



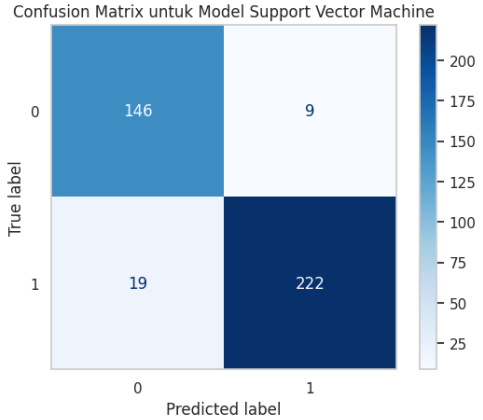
Gambar 10. Confusion Matrix KNN (Training)



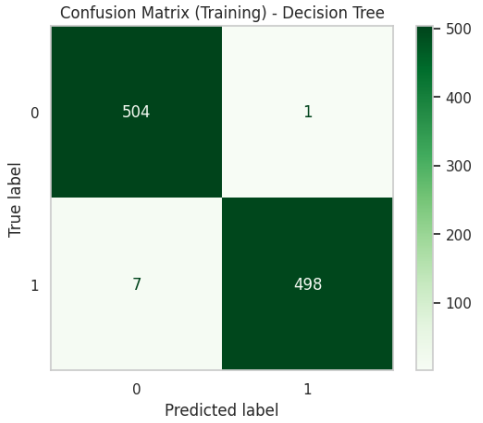
Gambar 11. Confusion Matrix KNN (Testing)



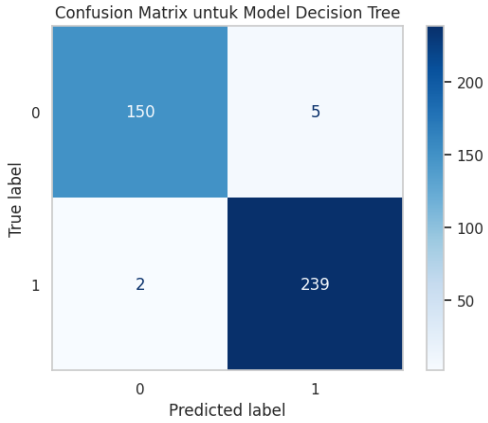
Gambar 12. Confusion Matrix SVM (Training)



Gambar 13. Confusion Matrix SVM (Training)



Gambar 14. Confusion Matrix DT (Training)



Gambar 15. Confusion Matrix DT (Testing)

Tabel 4 menunjukkan nilai *confusion matrix* dari masing-masing model pada *data training* dan *testing*. Nilai ini menunjukkan jumlah prediksi benar dan salah model.

TABEL 4
CONFUSION MATRIX

Model	True Positive		False Positive		True Negative		False Negative	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
KNN	310	113	37	31	468	114	195	104
SVM	468	197	10	2	495	143	37	20
Decision Tree	504	213	3	1	502	144	1	4

Hasil *confusion matrix* pada Tabel 4 menunjukkan bahwa *KNN* masih menghasilkan banyak *false negative*, sehingga akurasinya rendah. *SVM* lebih stabil dengan *true positive* dan *true negative* tinggi serta kesalahan prediksi rendah. *Decision Tree* tampil paling unggul dengan hasil hampir sempurna pada *data training* dan tetap konsisten terbaik pada *data testing*.

Tahap akhir penelitian adalah prediksi pada data baru dengan menggunakan model terbaik, yaitu *Decision Tree*. Hasil prediksi menunjukkan bahwa model mampu memberikan klasifikasi dengan akurasi tinggi serta konsistensi dalam mendeteksi pasien berisiko. Hal ini memperkuat potensi penerapan model sebagai sistem pendukung keputusan dalam diagnosis medis.

Dengan demikian, temuan ini menegaskan bahwa kombinasi *SMOTE* dan *tuning hyperparameter* terbukti mampu meningkatkan akurasi model prediksi serangan jantung. Selain itu, hasil penelitian juga menunjukkan bahwa *Decision Tree* merupakan algoritma yang paling efektif dan aplikatif untuk mendukung deteksi dini pasien berisiko serangan jantung, mengingat keunggulannya dalam hal akurasi, interpretabilitas, serta kemampuan menangani data klinis yang kompleks.

TABEL 5
CONFUSION MATRIX

Model	True Positive		False Positive		True Negative		False Negative	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
KNN	310	113	37	31	468	114	195	104
SVM	468	197	10	2	495	143	37	20
Decision Tree	504	213	3	1	502	144	1	4

Tabel 5 menunjukkan nilai *confusion matrix* dari masing-masing model pada *data training* dan *testing*. Nilai ini menunjukkan jumlah prediksi benar dan salah model.

Perbandingan ketiga Algoritma menghasilkan hasil bahwa algoritma *Decision Tree* unggul. Mulai dari akurasi tinggi, stabil, mudah diinterpretasi, dan efektif pada data medis dengan atribut numerik dan kategorikal. Sedangkan *SVM* cukup baik dengan akurasi tinggi, namun lebih kompleks dan masih kalah dibanding *Decision Tree*. Sebaliknya, algoritma *KNN* kurang optimal, dengan performa rendah pada *data testing*, sensitif terhadap *outlier* dan distribusi data.

Prediksi Data Baru menggunakan model terbaik yaitu *Decision Tree* digunakan untuk menguji data pasien baru. Hasil menunjukkan bahwa model mampu memprediksi dengan baik apakah pasien masuk kategori positif (berisiko serangan jantung) atau negatif (sehat). Pada uji coba, data pasien baru dengan nilai fitur tertentu diklasifikasikan sebagai positif, sesuai dengan kondisi klinis yang diharapkan.

```
new_data = pd.DataFrame({
    'Age': [65],
    'Gender': [1],
    'Heart rate': [95],
    'Systolic blood pressure': [160],
    'Diastolic blood pressure': [90],
    'Blood sugar': [180],
    'CK-MB': [6.5],
    'Troponin': [0.8]})
new_data #done
```

	Age	Gender	Heart rate	Systolic blood pressure	Diastolic blood pressure	Blood sugar	CK-MB	Troponin
0	65	1	95	160	90	180	6.5000	0.8000

Gambar 16. Data Baru

Pada Gambar 12. menunjukkan proses input data baru untuk dilakukan tahap prediksi data baru dengan membuat sebuah sampel pasien (usia 60 tahun, laki-laki, detak jantung 90 bpm, tekanan darah 160/90 mmHg, gula darah 180 mg/dL, CK-MB 6.5, Troponin 0.002).

```
new_data_scaled = scaler.transform(new_data)
dt_model = models['Decision Tree']
prediction = dt_model.predict(new_data_scaled)
hasil_prediksi = "negative" if prediction[0] == 0 else "positive"
print(f"Prediksi Penyakit oleh Decision Tree: {hasil_prediksi}")
```

```
Prediksi Penyakit oleh Decision Tree: positive
```

Gambar 17. Prediksi Data Baru

Kemudian data dinormalisasi menggunakan *StandardScaler* dan diprediksi dengan *Decision Tree* sebagai model terbaik pada penelitian ini. Prediksi data baru menghasilkan hasil prediksi positif penyakit serangan jantung pada pasien dengan kriteria tersebut.

I. SIMPULAN

Penelitian ini membandingkan kinerja tiga algoritma klasifikasi *KNN, SVM,* dan *Decision Tree* dalam memprediksi penyakit serangan jantung menggunakan dataset 1.319 pasien. Data diproses melalui normalisasi, *handling outlier*, dan penerapan *SMOTE* untuk menyeimbangkan kelas, sehingga distribusi positif dan negatif menjadi 505:505. Sebelum *SMOTE, data training* tidak seimbang (810 negatif, 509 positif), sedangkan setelah *SMOTE* jumlah *data training* menjadi 1.010 sampel dengan distribusi seimbang. *Tuning hyperparameter* menggunakan *GridSearchCV* dilakukan untuk menentukan nilai optimal, seperti k pada *KNN, kernel* pada *SVM,* dan *max_depth* pada *Decision Tree*. Hasil pengujian menunjukkan *Decision Tree* unggul pada data testing dengan akurasi, *precision, recall, dan F1-Score* 98%. *SVM* mencapai 92–93%, sedangkan *KNN* hanya 65–69% dan sensitif terhadap *outlier*. Dengan akurasi tinggi, interpretasi mudah, dan efektif untuk data medis numerik dan kategorikal, *Decision Tree* menjadi algoritma yang paling direkomendasikan. Untuk penelitian selanjutnya, disarankan memperluas variasi dataset, menambahkan fitur klinis tambahan, melakukan optimasi *hyperparameter* lebih mendalam, menggunakan teknik validasi silang seperti *Stratified K-Fold*, serta mencoba algoritma lain seperti *Random Forest, XGBoost*, atau *Gradient Boosting*. Dengan pengembangan tersebut, diharapkan model prediksi penyakit serangan jantung dapat semakin akurat, *robust*, dan aplikatif dalam mendukung diagnosis medis.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan dalam pelaksanaan penelitian ini. Ucapan terima kasih disampaikan kepada institusi yang telah menyediakan fasilitas dan sarana pendukung penelitian sehingga penelitian ini dapat dilaksanakan dengan baik. Selain itu, penulis juga mengucapkan terima kasih kepada semua pihak yang telah memberikan masukan, bimbingan, dan dukungan moral selama proses penelitian hingga penulisan artikel ini selesai.

DAFTAR PUSTAKA

[1] N. Yudistira and A. F. Putra, "Algoritma Decision Tree Dan Smote Untuk Klasifikasi Serangan Jantung Miokarditis Yang Imbalance," *J. Litbang Edusaintech*, vol. 2, no. 2, pp. 112–122, 2021, doi: 10.51402/jle.v2i2.48.

[2] Z. Maisat, E. Darmawan, and A. Fauzan, "Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM Implementation of GridSearchCV Hyperparameter Optimization in Heart Attack Prediction System Using SVM," *Unipdu*, vol. 13, no. 1, pp. 8–15, 2023.

[3] R. Apriyatmoko and F. Aini, "Remaja Mengenali Serangan Jantung Koroner," *Indones. J. Community Empower.*, vol. 2, no. 2, pp. 115–122, 2020, doi: 10.35473/ijce.v2i2.758.

[4] M. G. Pradana, P. H. Saputro, and D. P. Wijaya, "Komparasi Metode Support Vector Machine Dan Naïve Bayes Dalam Klasifikasi Peluang Penyakit Serangan Jantung," *Indones. J. Bus. Intell.*, vol. 5, no. 2, p. 87, 2022, doi: 10.21927/ijubi.v5i2.2659.

[5] N. Nuraeni, "Klasifikasi Data Mining untuk Prediksi Penyakit Kardiovaskular," *J. Tek. Inf. dan Komput.*, vol. 7, no. 1, pp. 161–170, 2024, doi: 10.37600/tekinkom.v7i1.

[6] A. Yogianto, A. Homaidi, and Z. Fatah, "Implementasi Metode K-Nearest Neighbors (KNN) untuk Klasifikasi Penyakit Jantung," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1720–1728, 2024, doi: 10.33379/gtech.v8i3.4495.

[7] M. D. F. Tino, Herliyani Hasanah, and Tri Djoko Santosa, "Perbandingan Algoritma Support Vector Machines (Svm) Dan Neural Network Untuk Klasifikasi Penyakit Jantung," *INFOTECH J.*, vol. 9, no. 1, pp. 232–235, 2023, doi: 10.31949/infotech.v9i1.5432.

[8] C. Andini Bahri *et al.*, "Analisis Faktor Risiko Pemicu Serangan Jantung Di Indonesia, Menggunakan Metode Klasifikasi (Decision Tree, Naive Bayes, Dan Random Forest)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 4, pp. 6155–6160, 2025, doi: 10.36040/jati.v9i4.13945.

[9] D. A. Ryfai, N. Hidayat, and E. Santoso, "Klasifikasi Tingkat Resiko Serangan Penyakit Jantung Menggunakan Metode K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 10, pp. 4701–4707, 2022, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11662.

[10] G. Ariyanto Pamungkas, R. Rahmadewi, and I. Purwita Sary, "Klasifikasi Risiko Penyakit Jantung Menggunakan Algoritma Vector Machine (Svm)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 3, pp. 5438–5443, 2025, doi: 10.36040/jati.v9i3.14243.

[11] R. Antika, A. Rifa'I, F. Dikananda, D. Indriya Efendi, and R. Narasati, "Penerapan Algoritma Decision Tree Berbasis Pohon Keputusan Dalam Klasifikasi Penyakit Jantung," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3688–3692, 2024, doi: 10.36040/jati.v7i6.8264.

[12] A. E. Setiawan, S. Rustad, A. Syukur, M. A. Soeleman, M. Akrom, and A. W. Setiawan, "SMOTE-ENN Resampling to Optimize Diabetes Prediction in Imbalanced Data," *Ing. des Syst. d'Information*, vol. 30, no. 5, pp. 1163–1176, 2025, doi: 10.18280/isi.300505.

[13] K. Alemerien, S. Alsarayreh, and E. Altarawneh, "Diagnosing Cardiovascular Diseases using Optimized Machine Learning Algorithms with GridSearchCV," *J. Appl. Data Sci.*, vol. 5, no. 4, pp. 1539–1552, 2024, doi: 10.47738/jads.v5i4.280.

[14] L. Nafi'ah and Z. Fatah, "Implementasi Algoritma Decision Tree Untuk Pendeteksian Penyakit Jantung," *JUSIFOR J. Sist. Inf. dan Inform.*, vol. 3, no. 2, pp. 160–165, 2024, doi: 10.70609/jusifor.v3i2.5729.

[15] A. Arfian, J. Siregar, and D. Wijayanti, "Integrasi Model Neural Network Dengan Svm Dan Knn Untuk Deteksi Dini Penyakita Jantung Pada Penderita Hypertensi," *Technol. J. Ilm.*, vol. 16, no. 1, p. 104, 2025, doi: 10.31602/tji.v16i1.17046.

[16] E. F. Laili, Z. Alawi, R. Rohmah, and M. A. Barata, "Komparasi Algoritma Decision Tree Dan Support Vector Machine (Svm) Dalam Klasifikasi Serangan Jantung," *J. Sist. Inf. dan Inform.*, vol. 8, no. 1, pp. 67–76, 2025, doi: 10.47080/simika.v8i1.3683.

[17] S. F. M. Radzi, M. K. A. Karim, M. I. Saripan, M. A. A. Rahman, I. N. C. Isa, and M. J. Ibahim, "Hyperparameter tuning and pipeline

optimization via grid search method and tree-based autoML in breast cancer prediction," *J. Pers. Med.*, vol. 11, no. 10, 2021, doi: 10.3390/jpm11100978.
[18] Rina, "Memahami Confusion Matrix: Accuracy, Precision, Recall, Specificity, dan F1-Score untuk Evaluasi Model Klasifikasi," *Medium*, 2023. https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf (accessed Apr. 14, 2025).
[19] R. Mulyawan, "Pengertian Precision and Recall: Definisi, Rumus, dan Contohnya!," *rifqimulyawan.com*, 2021. https://rifqimulyawan.com/kamus/precision-and-recall/ (accessed Apr. 14, 2025).
[20] C. AlKahfi, "Metrik Evaluasi untuk Model Klasifikasi," *SainsData.id*, 2023. https://sainsdata.id/machine-learning/8871/metrik-evaluasi-untuk-model-klasifikasi/ (accessed Apr. 14, 2025).
[21] M. K. Dr. Maria Susan Anggreany, S.KOM., "Confusion Matrix," *BINUS UNIVERSITY*. https://socs.binus.ac.id/2020/11/01/confusion-matrix/.