

# STUDI KOMPARASI ALGORITMA *RANDOM FOREST CLASSIFIER* DAN *SUPPORT VECTOR MACHINE* DALAM PREDIKSI PENYAKIT JANTUNG

Nur Halizah Alfajr<sup>1\*</sup>, Garno<sup>2</sup>, Dadang Yusup<sup>3</sup>

<sup>1,2,3</sup>Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361, Telp. (0267) 641177

## Keywords:

Studi Komparasi, Prediksi, Jantung, *Random Forest Classifier*, *Support Vector Machine*.

## Correspondent Email:

nurhlzh2403@gmail.com

**Abstrak.** Penyakit jantung merupakan penyakit tidak menular dengan tingkat kematian tinggi secara global dan nasional. WHO mencatat sekitar 17,9 juta kematian setiap tahun akibat penyakit kardiovaskular. Prediksi dini menjadi sangat penting untuk menurunkan angka kematian dan meningkatkan harapan hidup. Penelitian ini membandingkan kinerja algoritma machine learning *Random Forest Classifier* dan *Support Vector Machine* dalam memprediksi penyakit jantung. Data yang digunakan merupakan data rekam medis pasien poli jantung RSUD Kabupaten Bekasi tahun 2024, terdiri dari 5432 data dengan dua kelas (Label 1 (jantung) = 3068 dan Label 0 (non-jantung) = 2364). Model dibangun menggunakan pendekatan *Knowledge Discovery in Database* (KDD). Hasil evaluasi menunjukkan bahwa model *Support Vector Machine* memberikan performa terbaik dibandingkan *Random Forest Classifier* dengan akurasi 65%, precision 70%, recall 68%, dan f-measure 64%. Pengujian dengan cross-validation dan analisis ROC juga menunjukkan nilai AUC tertinggi pada model *Support Vector Machine*, yaitu 0,67 – 0,68 yang dikategorikan sebagai *poor classification*.

**Abstract.** Heart disease is a non-communicable disease with a high mortality rate both globally and in Indonesia. According to WHO, around 17.9 million deaths occur each year due to cardiovascular diseases. Early prediction is crucial to reducing mortality and improving life expectancy. This study compares the performance of machine learning algorithms *Random Forest Classifier* and *Support Vector Machine* in predicting heart disease. The dataset consists of 5432 medical records from cardiac outpatients at RSUD Kabupaten Bekasi in 2024, with two classes (labeled 1 (heart disease) = 3068 and labeled 0 (non-heart disease) = 2364). Models were developed using the *Knowledge Discovery in Databases* (KDD) approach. Evaluation results show that the *Support Vector Machine* model achieved the best performance compared to *Random Forest Classifier* with 65% accuracy, 70% precision, 68% recall, and 64% f-measure. Cross-validation and ROC analysis also indicated that *Support Vector Machine* obtained the highest AUC score, ranging from 0.67 to 0.68, which is categorized as *poor classification*.

## 1. PENDAHULUAN

Penyakit jantung merupakan salah satu penyakit tidak menular yang menjadi penyebab kematian tertinggi di dunia, termasuk Indonesia [1]. WHO mencatat bahwa sekitar 17,9 juta orang meninggal setiap tahunnya akibat

penyakit kardiovaskular [2]. Di Indonesia sendiri, prevalensi penyakit jantung lebih tinggi pada perempuan dibandingkan laki – laki. Faktor risiko seperti usia produktif, gaya hidup tidak sehat, kurangnya aktivitas fisik, serta

akses terbatas ke fasilitas kesehatan turut memperburuk kondisi ini [3].

Diagnosis dini terhadap penyakit jantung sangat penting untuk mencegah komplikasi serius dan menurunkan angka kematian. Namun, proses diagnosa sering kali lambat dan mahal karena melibatkan serangkaian pemeriksaan seperti ECG, CT scan, dan MRI [4].

Penggunaan teknologi *machine learning* menjadi solusi potensial dalam mempercepat proses prediksi penyakit jantung. Berbagai algoritma telah diterapkan, seperti *Naïve Bayes*, KNN, *Decision Tree*, *Logistic Regression*, *Random Forest*, dan *Support Vector Machine* [5]. Metode *supervised learning* banyak digunakan karena dapat mengklasifikasikan data berdasarkan label yang sudah ditentukan. Di antara algoritma tersebut, *Random Forest* dan *SVM* sering digunakan karena performanya yang baik dalam klasifikasi data medis [6].

Meskipun keduanya populer, setiap algoritma memiliki kelebihan dan kekurangannya masing – masing. *Random Forest* unggul dalam menangani dataset besar dan kompleks, serta cenderung stabil terhadap *overfitting* [7], meskipun interpretasi hasilnya cukup sulit. Di sisi lain, *SVM* menghasilkan akurasi tinggi untuk data berdimensi besar, namun sensitif terhadap parameter dan memerlukan waktu komputasi yang lebih tinggi [8]. Maka dari itu, studi komparatif diperlukan untuk mengetahui algoritma yang paling optimal dalam konteks data spesifik.

Penelitian ini bertujuan untuk membandingkan performa algoritma *Random Forest Classifier* dan *Support Vector Machine* dalam memprediksi penyakit jantung. Dataset yang digunakan berasal dari data rekam medis kunjungan pasien poli jantung di RSUD Kabupaten Bekasi tahun 2024, yang terdiri atas sekitar 5000 baris data dan 15 atribut. Dengan menggunakan data nyata dari rumah sakit, diharapkan hasil penelitian ini dapat memberikan kontribusi nyata dalam penerapan teknologi *machine learning* di sektor kesehatan.

## 2. TINJAUAN PUSTAKA

Penyakit jantung adalah ketika bagian jantung, yang terdiri dari pembuluh darah jantung, selaput jantung, katup jantung, dan otot jantung, bekerja dengan cara yang tidak normal. Faktor risiko lain seperti, tekanan darah tinggi,

diabetes, obesitas, gaya hidup tidak sehat, merokok, dan gen dapat menyebabkan penyakit jantung [9]. Penelitian komparatif membandingkan kemunculan satu atau lebih variabel pada dua atau lebih sampel pada waktu yang berbeda. Bertujuan untuk menemukan alasan dasar dari sebab dan akibat dari suatu peristiwa tertentu [10].

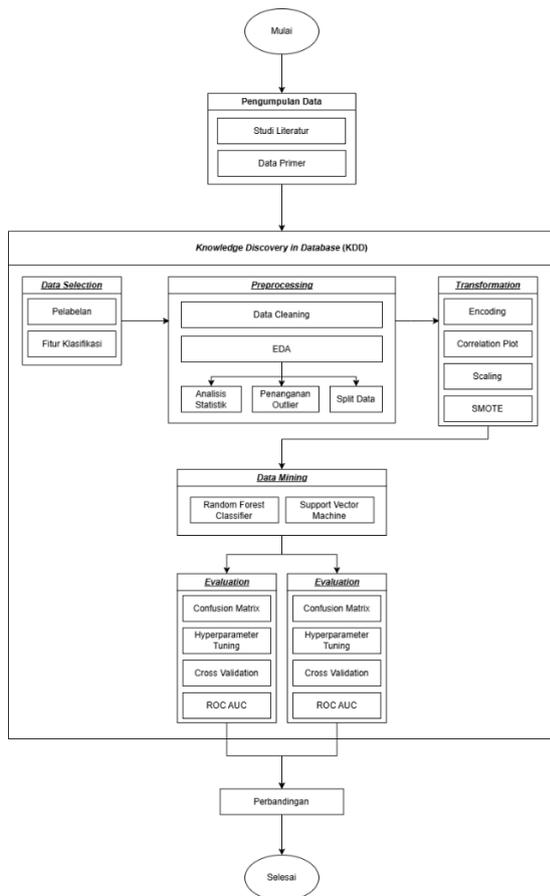
Berbagai studi sebelumnya menunjukkan bahwa baik *Random Forest* maupun *SVM* mampu memberikan hasil prediksi yang kompetitif. Misalnya, penelitian [11] memperoleh akurasi 92,63% menggunakan *Random Forest* dalam prediksi penyakit jantung. Penelitian oleh [12] mencatat akurasi hingga 98% menggunakan algoritma yang sama. Sementara itu, [13] menunjukkan bahwa *SVM* mampu mencapai precision 0,93 dan f1-score 0,83 dalam klasifikasi penyakit jantung. Studi – studi ini menunjukkan bahwa kedua algoritma memiliki potensi besar dalam dunia medis, namun perbandingan langsung berdasarkan data lokal seperti data rekam medis dari RSUD Kabupaten Bekasi masing jarang dilakukan, sehingga diperlukan studi komparatif yang lebih kontekstual.

## 3. METODE PENELITIAN

Penelitian ini dimulai dengan pengumpulan data dengan melihat literatur terkait dengan topik penelitian, seperti buku, jurnal, dan website. Lalu melakukan pengumpulan data dan informasi menggunakan data primer rekam medis kunjungan pasien poli jantung tahun 2024 dari RSUD Kabupaten Bekasi.

Metode penelitian yang digunakan dalam penelitian ini adalah *Knowledge Discovery in Database* yang merupakan proses penerapan teknik data mining untuk menemukan informasi berharga serta pola dalam data, dengan memanfaatkan algoritma guna mengidentifikasi pola tersebut [14].

KDD sendiri memiliki 5 tahapan, yakni *Data Selection*, *Preprocessing*, *Transformation*, *Data Mining*, *Evaluation*. Agar lebih mudah dipahami, alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

### 3.1. Pengumpulan Data

Langkah pertama adalah tahap pengumpulan data yaitu studi literatur dan pengumpulan data primer. Studi literatur dilakukan penulis dengan mengumpulkan data yang relevan dengan penelitian mengenai *Data Mining*, *Random Forest Classifier*, *Support Vector Machine*, penelitian sejenis dan mempelajari metode yang digunakan. Sedangkan pengumpulan data primer, peneliti berkesempatan mendapatkan data rekam medis kunjungan pasien poli jantung tahun 2024 dari RSUD Kabupaten Bekasi. Data yang mencakup informasi pasien yang relevan serta memperhatikan aspek etika, seperti memperoleh izin yang diperlukan untuk penggunaan data medis, sehingga data yang diperoleh valid dan sesuai dengan regulasi yang berlaku.

### 3.2. Knowledge Discovery in Database (KDD)

#### 3.2.1. Data Selection

Tahapan pertama KDD yaitu tahap *data selection* yang akan melakukan pelabelan data

untuk memberikan label pada setiap data pasien (kategori jantung atau non-jantung) dan akan melakukan pemilihan fitur – fitur yang relevan untuk proses klasifikasi sehingga *dataset* siap digunakan untuk membangun model prediksi.

#### 3.2.2. Preprocessing

Tahap *preprocessing* adalah tahap di mana akan melakukan pembersihan (*cleaning*) data untuk mengatasi masalah seperti data hilang atau *outlier* dan eksplorasi data untuk memahami antar fitur. Setelah itu, akan melakukan pembagian (*split*) data menjadi data pelatihan (*training*) dan pengujian (*testing*) untuk digunakan pada proses data mining.

#### 3.2.3. Transformation

Tahap selanjutnya adalah tahap *transformation* di mana akan melakukan tindakan transformasi pada data seperti *encoding* untuk mengubah data kategorik menjadi data numerik yang dapat digunakan dalam pemodelan, *scaling*, dan *resampling*.

#### 3.2.4. Data Mining

Tahap keempat adalah tahap *data mining* atau pemodelan di mana akan melakukan klasifikasi data mining untuk pemodelan, yang bertujuan mengubah data mentah menjadi *intelligen* yang berguna.

#### 3.2.5. Evaluation

Tahap terakhir KDD yaitu *evaluation* di mana akan melakukan penilaian kinerja model yang telah selesai diproses untuk mengevaluasi kemampuan prediktif model terhadap penyakit jantung. *Confusion Matrix* memungkinkan analisis di empat dimensi untuk menyajikan pandangan kinerja yang lebih jelas :

1. *Akurasi* : untuk mengetahui seberapa dekat nilai yang diantisipasi dan nilai aktual satu sama lain.
2. *Presisi* : berdasarkan prediksi positif, menentukan seberapa tepat atau presisi suatu model. Presisi adalah metrik yang bermanfaat saat model memiliki nilai positif palsu yang tinggi.
3. *Recall* : untuk menghitung jumlah nilai Positif Aktual yang telah diidentifikasi sebagai True Positif oleh model menggunakan label.
4. *F-Measure* : menunjukkan perbandingan antara presisi dan hasil rata-rata tertimbang.

Selain menggunakan *confusion matrix*, pengujian juga akan menggunakan

*hyperparameter tuning*, *cross validation* dan ROC AUC untuk mengukur dan meningkatkan performa model untuk kemampuan prediksi.

### 3.3. Perbandingan

Langkah terakhir dalam penelitian ini adalah akan melakukan perbandingan hasil kinerja model antara *Random Forest Classifier* dan *Support Vector Machine* untuk melihat model yang lebih efektif dalam memprediksi penyakit jantung.

## 4. HASIL DAN PEMBAHASAN

Hasil dan pembahasan untuk studi komparasi algoritma *Random Forest Classifier* dan *Support Vector Machine* dalam prediksi penyakit jantung yang diolah oleh peneliti menggunakan Visual Studio Code dan bahasa *python*, untuk masing – masing algoritma disajikan pada sub bab selanjutnya.

### 4.1. Hasil Pengumpulan Data

Hasil dari tahap pengumpulan data yang dilakukan dengan melakukan penelitian literatur yang mendukung topik penelitian, seperti penyakit jantung, *Random Forest Classifier*, *Support Vector Machine*, *Data Mining*, *Confusion Matrix*, dan *Dataset*. Selain itu hasil pengumpulan data dan informasi yang dihasilkan menggunakan data primer dari rekam medis kunjungan pasien poli jantung tahun 2024 dari RSUD Kabupaten Bekasi. Total data yang didapat berjumlah 5921 dan memiliki 15 atribut.

### 4.2. Hasil Knowledge Discovery in Database (KDD)

#### 4.2.1. Hasil Data Selection

Hasil dari data *selection* yang dilakukan dengan cara pengumpulan atribut yang akan digunakan dalam proses data mining dari data rekam medis kunjungan pasien poli jantung 2024 yang memiliki 15 fitur. Memberi label untuk data pasien yang terdiagnosa penyakit jantung atau tidak terdiagnosa penyakit jantung. Membuat kolom UMUR berdasarkan jenis kelamin pada fitur P dan L. Setelah melakukan pelabelan, atribut yang tidak digunakan dalam pemodelan akan dihapus dan dipilih 5 fitur yang akan digunakan dalam proses pemodelan data mining yaitu JKEL, KUNJUNGAN, KASUS, UMUR, LABEL. Hasil sampel pemilihan fitur dapat dilihat pada gambar 2 berikut.

	JKEL	KUNJUNGAN	KASUS	UMUR	LABEL
0	P	Lama	1. LAMA	12 Th	1
1	P	Lama	1. LAMA	14 Th	0
2	P	Lama	1. BARU	17 Th	0
3	P	Lama	1. LAMA	20 Th	0
4	P	Lama	1. BARU	21 Th	1

Gambar 2. Hasil Sampel Pelabelan

Penjelasan dari masing – masing atribut terbaru tersebut dapat dilihat pada tabel 1 berikut.

Tabel 1. Keterangan Atribut

No	Atribut	Tipe Data	Penjelasan
1	JKEL	Kategorikal	Jenis kelamin pasien poli jantung (P = Perempuan, L = Laki-Laki).
2	KUNJUNGAN	Kategorikal	Jenis kunjungan pasien poli jantung (Lama, Baru).
3	KASUS	Kategorikal	Jenis kasus pasien poli jantung (Lama, Baru)
4	UMUR	Kategorikal	Umur dari pasien poli jantung (0 – 90 thn)
5	LABEL	Numerik	Klasifikasi pasien yang terdiagnosa penyakit jantung (1 : Penyakit Jantung, 0 : Bukan Penyakit Jantung)

Berdasarkan tabel 1 hasil dari tahap data selection adalah terdapat 5921 data dan 5 atribut yang dipilih dengan 3 tipe kategorikal dan 1 tipe numerik.

#### 4.2.2. Hasil Preprocessing

Proses *cleaning* merupakan proses pembersihan data dari karakter – karakter, spasi, noise dan outlier yang tidak diperlukan. Hasil penerapan tahap *cleaning*, bisa dilihat pada tabel 2.

Tabel 2. Hasil Tahap Cleaning

Sebelum						Setelah					
JKEL	KUNJUNGAN	KASUS	UMUR	LABEL		JKEL	KUNJUNGAN	KASUS	UMUR	LABEL	
0	P	Lama	1. LAMA	12 Th	1	0	P	Lama	Lama	12.0	1
1	P	Lama	1. LAMA	14 Th	0	1	P	Lama	Lama	14.0	0
2	P	Lama	1. BARU	17 Th	0	2	P	Lama	Baru	17.0	0
3	P	Lama	1. LAMA	20 Th	0	3	P	Lama	Lama	20.0	0
4	P	Lama	1. BARU	21 Th	1	4	P	Lama	Baru	21.0	1

Setelah dilakukannya proses *cleaning*, data tersebut kemudian dilakukan proses pengecekan *missing value* di mana akan melihat jumlah *missing value* di setiap fitur dalam bentuk tabel. Tabel 3 adalah hasil pengecekan *missing value*.

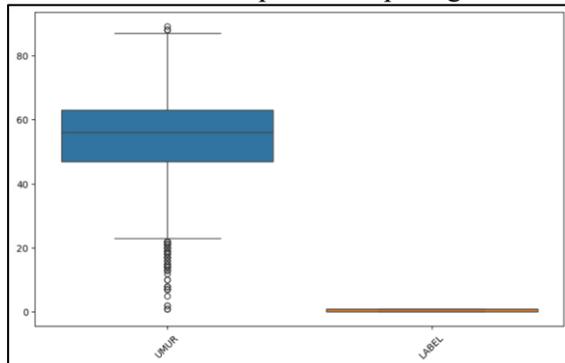
Tabel 3. Hasil Pengecekan Missing Value

Fitur	Jumlah Missing Value
JKEL	0
KUNJUNGAN	0
KASUS	353

UMUR	15
LABEL	0

Setelah dilakukan tahap penanganan *missing value*, jumlah data menjadi 5555. Data menjadi bersih dari data kosong (NaN), data tidak diisi atau data hilang karena pola tertentu.

Setelah itu, dilakukan tahap penanganan *outlier* dengan mengidentifikasi data yang tidak normal dan perlu ditangani. Hasil dari identifikasi *outlier* dapat dilihat pada gambar 3.



Gambar 3. Hasil Pengecekan Outlier

Pada gambar 3 dapat dilihat bahwa fitur UMUR memiliki *outlier* yang harus ditangani. *Outlier* terlihat sebagai titik yang jauh di luar whiskers (garis batas bawah dan atas). Selanjutnya adalah melakukan penghapusan *outlier* dengan metode IQR (*Interquartile Range*).

Tabel 4. Source Code Hapus Outlier

```
Source Code Hapus Outlier
Q1 = df_jantung['UMUR'].quantile(0.25)
Q3 = df_jantung['UMUR'].quantile(0.75)
IQR = Q3 - Q1

batas_bawah = Q1 - 1.5 * IQR
batas_atas = Q3 + 1.5 * IQR

df_no_outliers =
df_jantung[(df_jantung['UMUR'] >=
batas_bawah) & (df_jantung['UMUR'] <=
batas_atas)]

print(f'Jumlah data sebelum menghapus outlier:
{df_jantung.shape[0]}")
print(f'Jumlah data setelah menghapus outlier:
{df_no_outliers.shape[0]}")

plt.figure(figsize=(10, 6))
sns.boxplot(data=df_no_outliers)
plt.xticks(rotation=45)
plt.show()
```

Setelah menangani *outlier*, data yang ada berjumlah 5432 dan fitur UMUR tidak lagi

terdapat banyak *outlier*. Tabel 5 adalah jumlah pasien dalam label 1 dan 0 setelah tahap penanganan *outlier*.

Tabel 5. Jumlah Pasien Label 1 dan 0

Label	Jumlah Pasien
Label 1	3068
Label 0	2364
Jumlah	5432

Tahap selanjutnya adalah melakukan tahap pembagian (*split*) data. Dataset dibagi menjadi dua bagian selama proses pemodelan: data pelatihan (*train*) dan data pengujian (*test*). Data pelatihan digunakan untuk melatih model agar dapat memahami pola data, sedangkan data pengujian digunakan untuk mengevaluasi kinerja model yang telah dilatih.

Tabel 6. Hasil Split Data

Data Train	Data Test
3259	2173

Pembagian data dilakukan dengan rasio 60% untuk data pelatihan dan 40% untuk data pengujian. Berdasarkan hasil pembagian, diperoleh sebanyak 3259 data untuk pelatihan (*train*) dan 2173 data untuk pengujian (*test*).

#### 4.2.3. Hasil Transformation

Pada tahap ini dilakukan *transformasi* data di mana *dataset train* dan *test* akan di *encoding* secara terpisah menjadi tipe numerik agar mudah digunakan dalam proses pemodelan. Hasil dari penerapan *encoding* adalah sebagai berikut.

	JKEL	KUNJUNGAN	KASUS	UMUR	
	445	0	1	0	55.0
	4682	0	1	0	50.0
	741	1	1	0	56.0
	3539	0	1	1	43.0
	3889	0	1	0	48.0

Gambar 4. Hasil Encoding Data Train

Gambar 4 adalah hasil *encoding* dari data *train* yang berjumlah 3259 data dan menggunakan *label encoder*.

	JKEL	KUNJUNGAN	KASUS	UMUR
1631	1	1	0	46.0
4672	1	1	1	57.0
4698	1	1	1	48.0
3433	1	1	1	63.0
4481	1	1	0	43.0

Gambar 5. Hasil Encoding Data Test

Sedangkan gambar 5 adalah hasil *encoding* dari data *test* yang berjumlah 2173 data dan juga menggunakan *label encoder*.

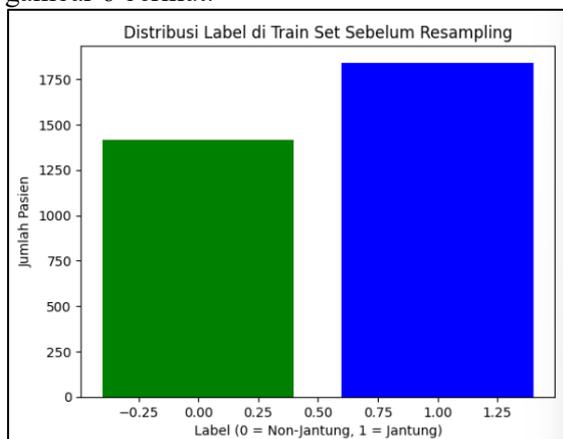
Selanjutnya, akan dilakukan tahap normalisasi data terhadap fitur tertentu untuk memastikan bahwa model nantinya akan bekerja secara optimal tanpa dipengaruhi oleh perbedaan skala antar variabel. Normalisasi dilakukan menggunakan metode *StandardScaler* yang berfungsi untuk mengubah distribusi data menjadi memiliki rata – rata 0 dan *standar deviasi* 1.

Tabel 7. Source Code Scaling

```
Source Code Scaling
scaler = StandardScaler()

features_to_scale = ['UMUR']
X_train[features_to_scale] =
scaler.fit_transform(X_train[features_to_scale])
X_test[features_to_scale] =
scaler.transform(X_test[features_to_scale])
```

Setelah melakukan tahap *scaling*, terlihat bahwa masih ada kelas pada data pelatihan (*train*) yang tidak seimbang. Ketidakseimbangan data dapat memengaruhi performa model *machine learning*, terutama jika salah satu kelas mendominasi. Hasil pengecekan *imbalance class* dapat dilihat pada gambar 6 berikut.



Gambar 6. Hasil Pengecekan Imbalance Class

Setelah mengetahui bahwa ada ketidakseimbangan kelas (*imbalance class*) pada data latih, maka selanjutnya akan diatasi dengan menerapkan teknik *resampling* yaitu menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). Teknik SMOTE digunakan untuk menyeimbangkan distribusi kelas dengan mensintesis data baru untuk kelas minoritas, sehingga model dapat belajar dengan lebih baik tanpa bias terhadap kelas mayoritas. Hasil dari tahap *resampling* terdapat pada gambar 7.

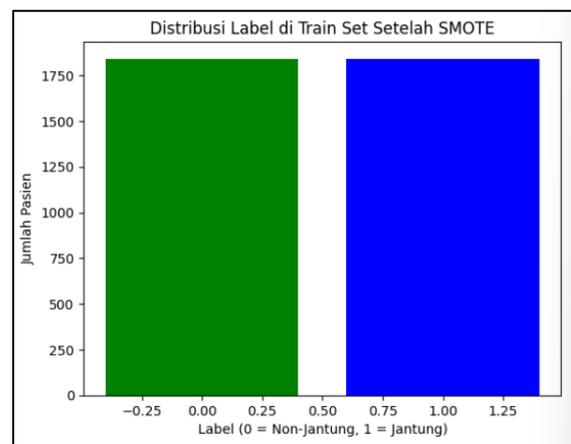
Tabel 8. Source Code SMOTE

```
Source Code SMOTE
smote = SMOTE(sampling_strategy='auto',
random_state=42)

X_train_resampled, y_train_resampled =
smote.fit_resample(X_train, y_train)

counter_resampled =
Counter(y_train_resampled)
print("Distribusi Label Setelah SMOTE:",
counter_resampled)

plt.bar(counter_resampled.keys(),
counter_resampled.values(), color=['blue',
'green'])
plt.xlabel("Label (0 = Non-Jantung, 1 =
Jantung)")
plt.ylabel("Jumlah Pasien")
plt.title("Distribusi Label di Train Set Setelah
SMOTE")
plt.show()
```



Gambar 7. Hasil Teknik SMOTE

Setelah melakukan *resampling* maka fitur label akan dipisahkan kembali.

**Tabel 9.** Hasil Split Data Setelah SMOTE

Data Train setelah SMOTE	Data Test
3682	2173

Berdasarkan tabel 9, hasil dari tahap SMOTE adalah pembagian data menjadi 3682 pada data *training* dan 2173 pada *testing*.

#### 4.2.4. Hasil Data Mining

Pada tahap klasifikasi, pengajaran mesin digunakan untuk melakukan *cross-validation* dan menghasilkan nilai prediksi untuk ketepatan. Hasil dari tahap klasifikasi digambarkan di bawah ini menggunakan script dari algoritma *Random Forest Classifier* dan *Support Vector Machine*.

**Tabel 10.** Source Code RFC

Algoritma <i>Random Forest Classifier</i>
<pre>rf_non = RandomForestClassifier(random_state=42) rf_non.fit(X_train_final, y_train_final)  y_pred rf_non = rf_non.predict(X_test)</pre>

**Tabel 11.** Source Code SVM

Algoritma <i>Support Vector Machine</i>
<pre>svm = SVC(probability=True, random_state=42) svm.fit(X_train_final, y_train_final)  y_pred svm = svm.predict(X_test)</pre>

#### 4.2.5. Hasil Evaluation

Hasil evaluasi dari output script diatas dapat dilihat pada gambar dibawah ini.

Random Forest:	precision	recall	f1-score	support
0	0.56	0.58	0.57	946
1	0.66	0.65	0.66	1227
accuracy			0.62	2173
macro avg	0.61	0.61	0.61	2173
weighted avg	0.62	0.62	0.62	2173
Random Forest Accuracy : 0.6161988034974689				
Random Forest Precision : 0.6109268351095678				
Random Forest Recall : 0.6117293937843207				
Random Forest F1-Score : 0.6111750469864319				

**Gambar 8.** Hasil Evaluasi RFC

Pada gambar 8, merupakan hasil evaluasi dari penerapan algoritma *Random Forest Classifier* pada prediksi penyakit jantung menunjukkan bahwa tingkat *accuracy* 62%, *presisi* 61%,

*recall* 61%, dan *f1-score* 61%, data *testing* yang digunakan sebanyak 2173 data atau 40% dari 5855 data, dari jumlah data yang digunakan dengan metode random pada saat testing.

SVM:	precision	recall	f1-score	support
0	0.56	0.88	0.69	946
1	0.84	0.47	0.60	1227
accuracy			0.65	2173
macro avg	0.70	0.68	0.64	2173
weighted avg	0.72	0.65	0.64	2173
SVM Accuracy : 0.6484123331799355				
SVM Precision : 0.6995560591790284				
SVM Recall : 0.6752357543709110				
SVM F1-Score : 0.6432659018163190				

**Gambar 9.** Hasil Evaluasi SVM

Pada gambar 9, merupakan hasil evaluasi dari penerapan algoritma *Support Vector Machine* pada prediksi penyakit jantung menunjukkan bahwa tingkat *accuracy* 65%, *presisi* 70%, *recall* 68%, dan *f1-score* 64%, data *testing* yang digunakan sebanyak 2173 data atau 40% dari 5855 data, dari jumlah data yang digunakan dengan metode random pada saat testing.

#### 4.3. Hasil Perbandingan

Setelah melakukan evaluasi pada model RFC dan SVM dengan pembagian 60:40, didapatkan hasil dari beberapa teknik evaluasi. Maka, tahap selanjutnya adalah membandingkan semua hasil model RFC dan SVM, menganalisis model mana yang memiliki performa terbaik dan optimal dalam memprediksi penyakit jantung. Tabel 12 merupakan hasil perbandingan dari tahap evaluasi *classification report*.

**Tabel 12.** Perbandingan Classification Report

Evaluasi Classification Report	Accuracy	Precision	Recall	F-measure
RFC	0.62	0.61	0.61	0.61
SVM	0.65	0.70	0.68	0.64
RFC_Resampling	0.64	0.65	0.65	0.64
SVM_Resampling	0.65	0.70	0.68	0.64
RFC_Tuning	0.63	0.63	0.63	0.63
SVM_Tuning	0.65	0.70	0.68	0.64
RFC_Resampling_Tuning	0.64	0.65	0.65	0.63
SVM_Resampling_Tuning	0.65	0.70	0.68	0.64

Tabel 13 merupakan hasil perbandingan dari tahap evaluasi *confusion matrix* pada model RFC dan SVM yang sudah *di-resampling* dan *tuning*.

**Tabel 13.** Perbandingan Confusion Matrix

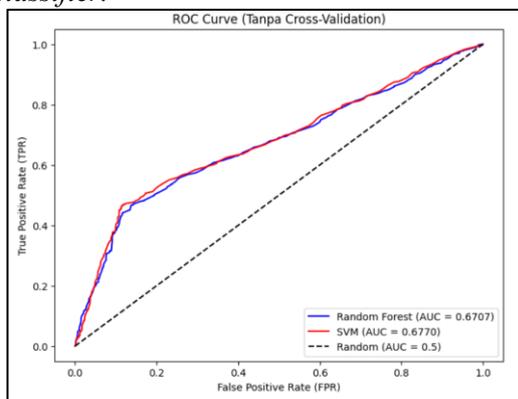
Model	Confusion Matrix		
		Prediksi 0	Prediksi 1
RFC	Aktual 0	721	225
	Aktual 1	568	659
SVM		Prediksi 0	Prediksi 1
	Aktual 0	835	111
	Aktual 1	653	574

Tabel 14 merupakan hasil perbandingan dari tahap evaluasi ROC AUC *train* dan *test* pada semua model RFC dan SVM.

**Tabel 14.** Perbandingan ROC AUC

Model	Recall Train	ROC AUC Train	Recall Test	ROC AUC Test
RFC	0.695274	0.746779	0.646292	0.662281
RFC_Resampling	0.567083	0.759549	0.556642	0.663915
SVM	0.440521	0.664873	0.467808	0.674872
SVM_Resampling	0.439978	0.651352	0.467808	0.677050
RFC_Tuning	0.622488	0.735458	0.596577	0.673014
RFC_Resampling_Tuning	0.534492	0.740892	0.537082	0.670670
SVM_Tuning	0.440521	0.664873	0.467808	0.674872
SVM_Resampling_Tuning	0.439978	0.651352	0.467808	0.677050

Setelah itu, dilakukan juga perbandingan hasil evaluasi kurva ROC AUC dari model *Random Forest Classifier* dan *Support Vector Machine* dan hasilnya model *Support Vector Machine* menghasilkan nilai AUC terbaik sebesar 68% dibandingkan dengan model *Random Forest Classifier*.



**Gambar 10.** Kurva ROC AUC

Penelitian ini membandingkan kinerja algoritma *Random Forest Classifier* (RFC) dan *Support Vector Machine* (SVM) dalam memprediksi penyakit jantung menggunakan pendekatan *Knowledge Discovery in Database* (KDD). Data rekam medis poli jantung tahun 2024 dari RSUD Kabupaten Bekasi sebanyak 5921 data digunakan, lalu melalui tahap preprocessing hingga tersisa 5432 data yang telah diberi label (1 = jantung, 0 = non-jantung). Setelah dilakukan pembagian data (60% train

dan 40% test), transformasi dilakukan melalui encoding, normalisasi menggunakan *StandardScaler*, dan penanganan *class imbalance* dengan SMOTE. Proses data mining menggunakan RFC dan SVM dilengkapi dengan evaluasi model melalui confusion matrix, cross-validation, ROC AUC, serta tuning parameter.

Hasil evaluasi menunjukkan bahwa SVM memiliki performa lebih unggul dibanding RFC dengan akurasi tertinggi 65%, sementara RFC maksimal 64%. Meski demikian, nilai AUC seluruh model berkisar antara 0,66–0,68 yang tergolong *Poor Classification*, mengindikasikan keterbatasan model dalam membedakan pasien dengan dan tanpa penyakit jantung. RFC lebih baik dalam mendeteksi kasus positif (recall lebih tinggi), sedangkan SVM cenderung memberikan akurasi keseluruhan lebih tinggi, namun berisiko menyebabkan *underdiagnosis* karena jumlah *false negative* (FN) yang lebih besar.

**5. KESIMPULAN**

- a. Hasil klasifikasi algoritma *Random Forest Classifier* dan *Support Vector Machine* dalam melakukan klasifikasi terhadap data penyakit jantung RSUD Kabupaten Bekasi tahun 2024 menggunakan metodologi KDD menghasilkan 3068 pasien kategori jantung (1) dan 2364 non-jantung (0), dengan jumlah kasus kategori jantung lebih dominan.
- b. Perbandingan performa *Random Forest Classifier* dan *Support Vector Machine* dalam memprediksi penyakit jantung dilakukan dengan *split data* 60 : 40 menunjukkan bahwa SVM unggul dalam skor performa meskipun AUC masih rendah (0.67 – 0.68), sedangkan RFC lebih banyak memprediksi kasus positif dan SVM lebih akurat memprediksi pasien sehat, namun dengan risiko false negative yang lebih tinggi.

**UCAPAN TERIMA KASIH**

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

## DAFTAR PUSTAKA

- [1] A. Fitri *et al.*, “Pengukuran Kinerja Model Klasifikasi dengan Data Oversampling pada Algoritma Supervised Learning untuk Penyakit Jantung,” *Computer Science (CO-SCIENCE)*, vol. 4, no. 1, pp. 62–70, Jan. 2024.
- [2] S. N. N. Arif, A. M. Siregar, S. Faisal, and A. R. Juwita, “Klasifikasi Penyakit Serangan Jantung Menggunakan Metode Machine Learning K-Nearest Neighbors (Knn) Dan Support Vector Machine (Svm),” *Jurnal Media Informatika Budidarma*, Vol. 8, No. 3, Pp. 1617–1626, Jul. 2024.
- [3] I. S. Sholihah, N. Wirawan, Y. Rohyadi, And M. I. A. Kusuma, “Hubungan Jarak Tempat Tinggal, Alat Transportasi, Serta Persepsi Pasien Terhadap Keterlambatan Pasien Ke Instalasi Gawat Darurat Pada Pasien Penyakit Jantung Koroner,” *Jurnal Keperawatan 'Aisyiyah*, vol. 8, no. 1, pp. 7–15, 2021.
- [4] A. S. Aulia, “Implementasi Ensemble Learning Menggunakan Algoritma SVM dan ADABOOST Untuk Memprediksi Penyakit Jantung,” Bachelor’s thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, 2024.
- [5] S. Das, K. Bhattacharyya, and S. Sarkar, “Performance Analysis of Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest and SVM on Hate Speech Detection from Twitter,” *International Research Journal of Innovations in Engineering and Technology*, vol. 07, no. 03, pp. 24–28, 2023.
- [6] D. Kusumaningrum and E. M. Imah, “Studi Komparasi Algoritma Klasifikasi Mental Workload Berdasarkan Sinyal Eeg,” *Jurnal Sistem Cerdas*, Vol. 3, no. 2, pp. 133–143, 2020.
- [7] M. Salsabil, N. L. Azizah, and A. Eviyanti, “Implementasi Data Mining Dalam Melakukan Prediksi Penyakit Diabetes Menggunakan Metode Random Forest Dan Xgboost,” *Jurnal Ilmiah Komputasi*, vol. 23, no. 1, pp. 51–58, Mar. 2024.
- [8] M. R. Haditama, “Analisis dan pembuatan dashboard prediksi kelulusan mahasiswa menggunakan metode random forest, naïve bayes dan support vector machine,” *Bachelor’s thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta*, 2023.
- [9] N. H. Alfajr and S. Defiyanti, “Prediksi Penyakit Jantung Menggunakan Metode Random Forest Dan Penerapan Principal Component Analysis (Pca),” *Jitet (Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 12, no. 3S1, pp. 3457–3464, Oct. 2024.
- [10] E. Amrin, R. Rismawati, G. Gosro, and A. Asriany, “Studi Komparasi Layanan Fintech Dalam Meningkatkan Keuangan Inklusif Pada Umkm Di Kota Palopo,” *Ecobisma (Jurnal Ekonomi, Bisnis Dan Manajemen)*, vol. 9, no. 2, pp. 114–125, 2022.
- [11] S. A. Putri, N. Selayanti, M. Kristanaya, M. P. Azzahra, M. G. Navsih, and K. M. Hindrayani, “Penerapan Machine Learning Algoritma Random Forest Untuk Prediksi Penyakit Jantung,” *Prosiding Seminar Nasional Sains Data*, vol. 4, no. 1, pp. 895–906, 2024.
- [12] N. Nuraeni, “Klasifikasi Data Mining Untuk Prediksi Penyakit Kardiovaskular,” *Jurnal TEKINKOM*, vol. 7, no. 1, pp. 161–170, 2024.
- [13] R. Hidayat, Y. Saputra Sy, T. Sujana, M. Husnah, H. T. Saputra, and F. Okmayura, “Implementasi Machine Learning Untuk Prediksi Penyakit Jantung Menggunakan Algoritma Support Vector Machine,” *BIOS : Jurnal Teknologi Informasi dan Rekayasa Komputer*, vol. 5, no. 2, pp. 161–168, 2024.
- [14] M. Muttaqin *et al.*, *Pengenalan Data Mining*, 1st ed. Yayasan Kita Menulis, 2023.