

Tersedia online di [www.journal.unipdu.ac.id](http://www.journal.unipdu.ac.id)  
**Unipdu**Halaman jurnal di [www.journal.unipdu.ac.id/index.php/teknologi](http://www.journal.unipdu.ac.id/index.php/teknologi)**Research article**

# Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM

Zakha Maisat Eka Darmawan <sup>a</sup>, Ashafidz Fauzan Dianta <sup>b\*</sup><sup>a,b</sup> Departemen Teknologi Multimedia Kreatif, Politeknik Elektronika Negeri Surabaya, Indonesiaemail: <sup>a</sup> [zakha@pens.ac.id](mailto:zakha@pens.ac.id) <sup>b\*</sup> [ashafidz@pens.ac.id](mailto:ashafidz@pens.ac.id)

\*Korespondensi

Dikirim 25 Desember 2022; Direvisi 01 Januari 2023; Diterima 15 Januari 2023; Diterbitkan 20 Januari 2023

**Abstrak**

Serangan jantung disebabkan oleh penyumbatan atau penyempitan pada pembuluh darah secara tiba-tiba sehingga aliran darah yang mengalir ke jantung terjadi gangguan. Jika aliran darah berkurang, penderita akan merasakan nyeri dada sehingga itu pertanda atau peringatan seseorang dapat terkena serangan jantung. Salah satu cara efektif dalam mengidentifikasi dan memprediksi penyakit jantung adalah dengan memanfaatkan algoritma *machine learning*. Pada penelitian terdahulu banyak sistem prediksi serangan jantung menggunakan algoritma *naive bayes*. Tujuan dari penelitian ini adalah mengimplementasikan algoritma *Support Vector Machine* (SVM) untuk memprediksi kemungkinan terkena serangan jantung dengan menggunakan model klasifikasi dari dua kelas yaitu kelas kemungkinan kecil terkena serangan jantung dan kelas kemungkinan besar terkena serangan jantung. Teknik *data mining* digunakan untuk pengolahan data untuk mencari pengetahuan baru dari basis data sehingga ditemukan pola-pola yang valid, bermanfaat dan dapat dipelajari dengan mudah. Penelitian ini menggunakan teknik optimasi *hyperparameter GridSearchCV* pada model klasifikasi SVM dan pengujian akurasi yang dilakukan menggunakan *Confusion Matrix*. Dari 302 total jumlah data, 211 data digunakan sebagai *data training*, 45 data digunakan sebagai data uji dan 46 data digunakan data validasi, sistem prediksi serangan jantung bekerja dengan performa yang baik karena memiliki nilai *accuracy* sebesar 86,0%, *precision* sebesar 84,0%, *recall* sebesar 91,0%, dan *f1-score* sebesar 87,0%.

**Kata Kunci:** Prediksi Serangan Jantung, *Data Mining*, *Support Vector Machine* (SVM), *GridSearch CV*

## Implementation of GridSearchCV Hyperparameter Optimization in Heart Attack Prediction System Using SVM

**Abstract**

A heart attack is caused by a sudden blockage or narrowing of a blood vessel so that blood flow to the heart is disrupted. If blood flow is reduced, the sufferer will feel chest pain so that is a sign or warning that someone could have a heart attack. One of the effective ways to identify and predict heart disease is to utilize machine learning algorithms. In previous studies, many heart attack prediction systems used the *naive Bayes* algorithm. The purpose of this study is to implement the *Support Vector Machine* (SVM) algorithm to predict the probability of having a heart attack by using a classification model of two classes, namely the class with a low probability of having a heart attack and the class with a high probability of having a heart attack. Data processing is processed using data mining techniques to search for new knowledge from the database so that patterns are found that are valid, useful and can be learned easily. This study uses the *GridSearchCV* hyperparameter optimization technique in the SVM classification model and accuracy testing is carried out using the *Confusion Matrix*. Of the 302 total data, 211 data were used as training data, 45 data were used as test data and 46 data were used as validation data. The heart attack prediction system worked well because it had an accuracy value of 86.0%, a precision of 84.0 %, recall of 91.0%, and f1-score of 87.0%.

**Keywords:** Heart Attack Prediction, *Data Mining*, *Support Vector Machine* (SVM), *GridSearch CV*.

Untuk mengutip artikel ini dengan APA Style:

Darmawan Z. M. E., Dianta A. F. (2022). Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 13 (1), 8-15. <https://doi.org/10.26594/teknologi.v13i1.3098>



## 1. Pendahuluan

Menurut data *Global Burden of Disease (GBD)* dan *Institute for Health Metrics and Evaluation (IHME)* 2014-2019 penyebab kematian tertinggi di Indonesia adalah penyakit jantung. Sedangkan data *World Health Organization (WHO)* menyebutkan, sepertiga dari kematian secara global disebabkan oleh penyakit jantung (*cardiovascular*). Penyakit *Cardiovascular* menyebabkan kematian sekitar 17,9 juta orang setiap tahun di seluruh dunia, Sebagian besar disebabkan penggunaan tembakau, diet tidak sehat dan obesitas, kurangnya aktivitas fisik atau tidak rutin berolahraga, dan konsumsi alkohol berlebihan.

Klasifikasi adalah proses untuk menemukan model yang dapat membedakan kelas-kelas dari setiap data yang ada, dengan tujuan agar model tersebut dapat digunakan untuk memprediksi data-data yang belum diketahui kelasnya (Han et al., 2012). Klasifikasi merupakan salah satu tugas yang penting untuk beberapa aplikasi seperti *text categorization*, *tone recognition*, dan *data classification* (Srivastava & Bhambhu, 2010). Berkembangnya teknologi kecerdasan buatan maka berkembanglah *machine learning* yang banyak digunakan untuk memecahkan berbagai permasalahan dalam membantu pekerjaan manusia. Metode klasifikasi didalam *machine learning* diantaranya *Random Forest*, *Naive Bayes*, *Classification and Regression Trees (CART)*, *Support Vector Machines (SVM)*, dan lain-lain.

Metode *Naive Bayes* merupakan salah satu metode yang digunakan untuk mengidentifikasi dan memprediksi penyakit. Misalnya, penerapan metode *Naive Bayes* untuk klasifikasi penyakit jantung, seperti penelitian dengan judul "Deteksi Penyakit Jantung Menggunakan Algoritma Klasifikasi *Data Mining Naive Bayes*", peneliti menggunakan 24 atribut dimana tingkat akurasi yang dihasilkan sebesar 81,19% (Zuliyanti, 2017). Adapun penelitian lain terkait prediksi penyakit jantung yang menggunakan 14 atribut memiliki tingkat akurasi sebesar 85,33% (Sari & Alfionita, 2022) dan 83,1% (Larassati et al., 2022). Alogaritma yang lain juga digunakan untuk memprediksi penyakit jantung, misalnya alogaritma XGBOOST memiliki tingkat akurasi sebesar 80% (Nugraha, 2021).

Alogaritma *Support Vector Machines (SVM)* adalah metode klasifikasi untuk *data linear dan non-linear*. *SVM* merupakan salah satu teknik *machine learning* yang populer digunakan untuk klasifikasi serta regresi. *SVM* memanfaatkan fungsi kernel yang secara sistematis membuat *support vector classifier* dalam bentuk dimensi yang lebih tinggi (Chowdhury & Schoen, 2020). Cara kerja dari *SVM* pada umumnya adalah menemukan jarak terjauh dari *hyperplane* dengan kedua kelas. Menentukan jarak terjauh dilakukan proses berulang kali hingga menemukan *hyperplane* terbaik (Jumeilah, 2017). Maka dari itu diperlukan optimasi pada *SVM* untuk menemukan jarak maksimum *hyperplane* dengan kedua kelas tersebut. *SVM* banyak digunakan para peneliti bioinformatika untuk memprediksi penyakit, misalnya penelitian memprediksi penyakit *diabetes millitus* (Hovi et al., 2022), dalam penelitian tersebut dengan menggunakan metode alogaritma *SVM Radial Basis Function (RBF)* telah berhasil dicapai tingkat akurasi yang tinggi yakni *accuracy* sebesar 91.2%, *precision* sebesar 93.0%, *recall* sebesar 94.3%, dan *f1-scorer* sebesar 93.7%. Penelitian lain digunakan untuk memprediksi penyakit liver, dimana peneliti berhasil mendapatkan akurasi 77,36% (Musyafaa & Rifai, 2018).

Penelitian lain menggunakan *SVM* digunakan untuk memprediksi bencana banjir (Dwiasnati & Devianto, 2021), tingkat akurasi yang dihasilkan dengan menggunakan alogaritma *SVM* dengan optimasi *Particle Swarm Optimization (PSO)* sebesar *SVM* 97,62%. Selain itu alogaritma *SVM* juga digunakan untuk memprediksi prestasi siswa sekolah menengah kejuruan dengan subjek penelitian sebanyak 101 siswa, mendapatkan akurasi sebesar 83,3% (Prasojo et al., 2021).

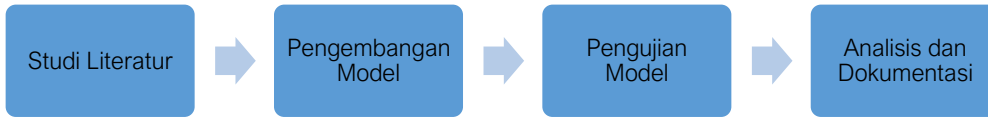
Untuk meningkatkan akurasi model klasifikasi dengan cara optimasi *hyperparameter* (Siji George C. G. & B. Sumathi, 2020). Penelitian terkait optimasi menggunakan *GridSearchCV* telah dilakukan, dimana peneliti berhasil memprediksi kualitas udara di Jakarta dengan mendapatkan peningkatan akurasi sebesar 94,8% (Toha et al., 2022).

Tujuan dari penelitian ini adalah membuat sistem prediksi serangan jantung menggunakan metode *Support Vector Machine* dengan optimasi *Hyperparameter GridSearchCV* menggunakan *dataset* yang berasal dari *Kaggle* dimana target nilai 1 digunakan untuk memprediksi dengan kemungkinan besar terkena serangan jantung sedangkan target 0 untuk memprediksi dengan kemungkinan kecil terkena serangan jantung.

2. Metode Penelitian

2.1. Alur Penelitian

Terdapat 4 tahapan yang dilakukan pada penelitian ini, Tahapan-tahapan seperti terlihat pada gambar 1.

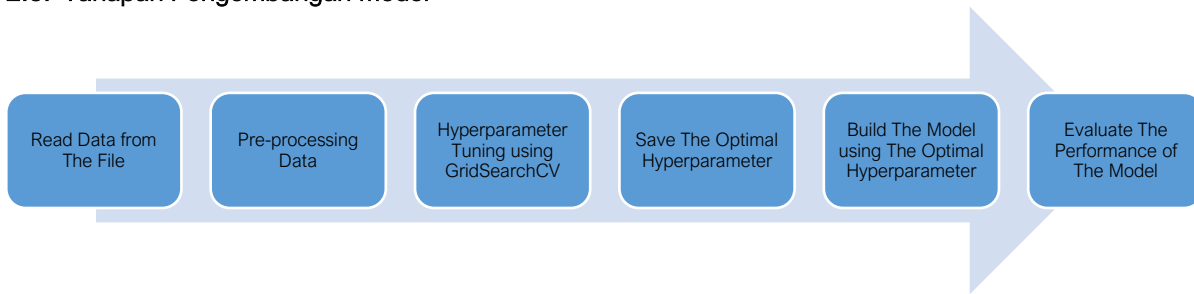


Gambar 1. Tahapan Penelitian

2.2. Dataset

Data yang digunakan pada penelitian ini diambil dari situs terkenal bernama *Kaggle* dengan judul “*Heart Attack Analysis & Prediction Dataset*”. *Dataset* dapat diakses pada laman <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>.

2.3. Tahapan Pengembangan Model



Gambar 2. Tahapan Pengembangan Model

Terdapat enam tahapan pada pengembangan sistem prediksi serangan jantung menggunakan *SVM* ini. Berikut merupakan penjelasan dari masing-masing tahap

a). *Read Data from The File*

Pada tahap ini dilakukan untuk membaca data yang masih dalam bentuk file yang berekstensi *csv* selanjutnya diubah menjadi sebuah variabel bertipe *dataframe* yang akan mempermudah pengolahan data di tahap berikutnya. Atribut pada data tersebut antara lain: *age, sex, cp, trtbps, chol, fbr, restecg, thalachh, exng, oldpeak, slp, caa, thall, output*.

b). *Pre-processing Data*

Pada tahap ini dilakukan pemeriksaan baris data yang duplikat atau memiliki nilai nil dalam kolomnya, lalu data dinormalisasi dengan tujuan mengurangi redudansi data dan memastikan dependasi data. Selain itu, supaya mendapatkan hasil evaluasi yang valid, dilakukan pemisahan data menjadi tiga jenis, yaitu *train, test, dan validation* dengan persentase yang ditentukan.

Tabel 1. Jumlah Data pada Setiap Jenis Data

No	Jenis Data	Persentase	Jumlah Data
1	Train	70%	211
2	Test	15%	45
3	Validation	15%	46
Jumlah Keseluruhan		100%	302

c). *Hyperparameter Tuning using GridSearchCV*

Pada tahap ini dilakukan *hyperparameter tuning* dengan tujuan mengetahui parameter-parameter yang dimasukkan ke dalam model yang dapat menghasilkan performa yang paling optimal. Sementara, *cross-validation grid serach (GridSearchCV)* digunakan untuk mengoptimalkan nilai akurasi (Siji George C. G. & B. Sumathi, 2020). *GridSearchCV* merupakan bagian dari modul *scikit-learn* yang melakukan validasi untuk lebih dari satu model serta menyediakan *hyperparameter* masing-masing secara otomatis dan sistematis.

Tabel 2. Prediksi Parameter yang Akan Dimasukkan ke Model

No	Parameter	Nilai
1	C	0.1, 1, 10, 100, 1000
2	Gamma	1, 0.1, 0.01, 0.001, 0.0001
3	Kernel	'rbf'

**d). Save The Optimal Hyperparameter**

Setelah dilakukan *hyperparameter tuning*, maka akan diketahui parameter-parameter yang diprediksi dapat memberikan performa yang optimal pada model yang dibangun. Oleh karena itu, perlu disimpan supaya mempermudah proses evaluasi model.

**e). Build The Model using The Optimal Hyperparameter**

Pada tahap ini dilakukan pembangunan model menggunakan parameter-parameter yang telah disimpan sebelumnya, lalu model tersebut akan menyesuaikan antara data yang menjadi *features* dan *class-label*, biasanya ini diketikkan pada baris kode sebagai model.fit().

**f). Evaluate The Performance of The Model**

Pada tahap ini dilakukan evaluasi terhadap model yang telah dibangun dengan cara menjalankan skenario pengujian yang telah dirancang, serta menghitung dan visualisasi *performance metrics* untuk sistem klasifikasi

**2.4. Skenario Pengujian**

Terdapat 15% dari keseluruhan data yang digunakan untuk pengujian pada penelitian ini dengan tujuan mengevaluasi model yang telah dikembangkan. Untuk menilai performa dari sebuah model klasifikasi, ada empat matriks yang populer dipakai, diantaranya adalah *accuracy*, *precision*, *recall*, dan *f1-score*. Dimana keempat matriks tersebut termasuk ke dalam *Confusion Matrix*.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) <i>Type I Error</i>
	0 (Negative)	<b>FN</b> (False Negative) <i>Type II Error</i>	<b>TN</b> (True Negative)

Gambar 3. *Confusion Matrix*

*Confusion matrix* adalah sebuah *tool* yang berguna dalam menganalisis seberapa baik metode klasifikasi yang digunakan dapat mengenali objek pengamatan dari kelas yang berbeda (Maalouf & Siddiqi, 2014). Terdapat empat istilah yang digunakan untuk pengukuran kinerja pada *Confusion Matrix*, sebagai representasi hasil proses klasifikasi yang terlihat pada Gambar 3. Keempat istilah tersebut dikenal dengan *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)* dan *False Negative (FN)*. Evaluasi dengan *confusion matrix* menghasilkan nilai *accuracy*, *precision*, dan *recall*.

**a). Accuracy**

*Accuracy* merupakan proporsi jumlah hasil yang benar dari keseluruhan data yang ada. *Accuracy* digunakan untuk menggambarkan seberapa akurat model yang digunakan dalam klasifikasi. Persamaan untuk menghitung ketelitian dapat dilihat pada Persamaan 1.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Predictions} \quad (1)$$

**b). Precision**

*Precision* merupakan rasio dari hasil yang diidentifikasi secara positif yang benar. *Precision* digunakan untuk menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Persamaan untuk menghitung presisi dapat dilihat pada Persamaan 2.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

**c). Recall**

*Recall* merupakan proporsi data yang benar-benar positif yang diprediksi positif. Persamaan untuk menghitung *recall* dapat dilihat pada Persamaan 3.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

#### d). F1 Score

F1 Score merupakan nilai gabungan dari presisi dan recall dan menghitungnya sebagai rata-rata harmonik. Persamaan untuk menghitung *F1-score* dapat dilihat pada Persamaan 4.

$$F1\ Score = \frac{2 \times Precision \times recall}{Precision + Recall} \quad (4)$$

### 2.5. Library yang Digunakan

Terdapat dua *library* yang berbasis bahasa pemrograman *Python* yang dimanfaatkan untuk pengembangan model klasifikasi menggunakan *SVM* dalam penelitian ini, diantaranya adalah sebagai berikut.

#### a). Scikit-learn

*Scikit-learn* atau *sklearn* adalah modul untuk bahasa pemrograman *python* yang dibangun diatas *NumPy*, *SciPy*, dan *matplotlib*. *Scikit-learn* berfungsi untuk membantu melakukan *processing* data ataupun melakukan *training data* untuk kebutuhan *machine-learning*. Fitur yang dapat digunakan didalam *Scikit-learn* antara lain, seperti *Clustering*, *Classification*, *Regression*, *Dimensionality*, *Model selection*, *Preprocessing Data*, dan *Reduction*.

#### b). Pandas

*Pandas* menyediakan struktur data yang digunakan dalam bahasa pemrograman *python*, memiliki pustaka *open source* berlisensi *BSD* dan analisis data berkinerja tinggi. Tahun 2008 *Wes McKinney* menginisiasi *Pandas* saat berada di lokasi *AQR Management Capital* (Nelli, 2018). *Pandas* sering digunakan untuk membaca data atau memanggil data dari berbagai macam jenis data khususnya tekstual. Untuk memanggil fungsi *pandas* menggunakan *script* "*Import pandas as pd*".

## 3. Hasil dan Pembahasan

### 3.1. Transformasi Data

*Dataset* yang digunakan didalam penelitian ini merupakan data rekam medik pasien dengan jumlah data sebanyak 303 dan jumlah atribut sebanyak 14 dengan atribut dapat dilihat pada Tabel 3.

Tabel 3. Ringkasa Atribut *Dataset*

No	Atribut	Nilai
1	Age (Usia)	29-77
2	Sex (jenis kelamin)	0;1
3	CP/Chest Pain Type (jenis nyeri dada)	0;1;2;3
4	trtbps/Trestbps (tekanan darah)	94-200
5	Chol (kolesterol)	126-564
6	Fbs/Fasting Blood Sugar (gula darah)	0;1
7	Restecg (hasil elektrokardiografi)	0;1;2
8	Thalachh (detak jantung maksimum)	71-202
9	Exng/Exang (nyeri dada apabila berolahraga)	0;1
10	Oldpeak (segmen ST yang didapatkan berdasarkan latihan relatif pada istirahat)	0,0-6,2
11	Slp/Slope (kemiringan segmen ST dalam latihan maksimal)	0;1;2
12	Caa (total vessel utama yang dilakukan pewarnaan oleh fluoroskopi)	0;1;2;3;4
13	Thall (status jantung)	0;1;2;3
14	Output (variable kelas)	0;1

Terdapat dua *class-label* pada atribut *output* yaitu 0 yang berarti kemungkinan kecil terkena serangan jantung dan 1 yang berarti kemungkinan besar terkena serangan jantung.

### 3.2. Pengujian

Pengujian ini bertujuan untuk mencari nilai akurasi sebelum dan setelah di optimasi.

#### a). Pengujian Sebelum Optimasi

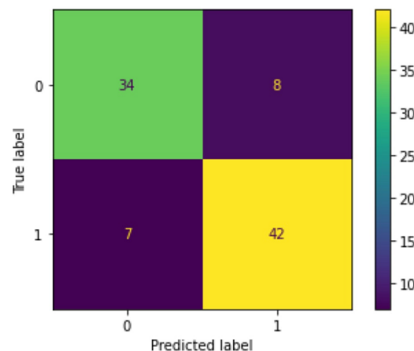
Hasil pengujian dapat diketahui bahwa performa model sebelum dilakukan optimasi menggunakan *GridSearchCV* sebagai *hyperparameter tuning* memiliki nilai yang kurang dari 90% yang berarti masih

terdapat banyak missing dalam memprediksi berdasarkan data yang ada. Hasil ujicoba yang sudah dilakukan, nilai *confusion matrix* sebelum optimasi dapat dilihat pada Tabel 4. Sedangkan hasil perhitungannya sebagai berikut:

- $Accuracy = \frac{34+42}{91} = 0.83$
- $Precision = \frac{34}{34+8} = 0.84$
- $Recall = \frac{34}{34+7} = 0.85$
- $F1\ Score = \frac{2 \times 0.81 \times 0.83}{0.81 + 0.83} = \frac{1.34}{1.64} = 0.84$

Tabel 4. Hasil Confusion Matrix Testing Sebelum Optimasi

No	Metric	Nilai
1	Accuracy	0.8351
2	Precision	0.84
3	Recall	0.8571
4	F1 Score	0.8484

Gambar 4. Visualisasi *Confusion Matrix* sebelum optimasi

#### b). Pengujian Setelah Optimasi

Untuk membantu meningkatkan performa dari model yang akan bekerja untuk prediksi serta meningkatkan kualitas dari sistem itu sendiri dilakukan optimasi menggunakan *GridSearchCV* sebagai *hyperparameter tuning*. Pada proses *hyperparameter tuning* menggunakan *GridSearchCV* pada tahapan pengembangan model klasifikasi diperoleh kombinasi parameter yang diprediksi menghasilkan performa yang paling optimal. Pada Tabel 5 di bawah ini menjabarkan parameter-parameter apa saja yang dimasukkan ke dalam model.

Tabel 5. Prediksi Parameter yang Akan Dimasukkan ke Model

No	Parameter	Nilai
1	C	1000
2	Gamma	0.0001
3	Kernel	'rbf'

Setelah membangun model menggunakan parameter-parameter tersebut, lalu dilakukan evaluasi sesuai skenario pengujian yang telah ditentukan. Evaluasi model didokumentasikan dalam bentuk perhitungan nilai dan visualisasi diagram dari *Confusion Matrix*. Perhitungan performa model setelah dilakukan optimasi menggunakan rumus 1, 2, 3 dan 4 adalah sebagai berikut:

- $Accuracy = \frac{18+21}{45} = 0.86$
- $Precision = \frac{18}{18+4} = 0.84$
- $Recall = \frac{18}{18+2} = 0.91$
- $F1\ Score = \frac{2 \times 0.82 \times 0.9}{0.82 + 0.9} = \frac{1.48}{1.72} = 0.87$

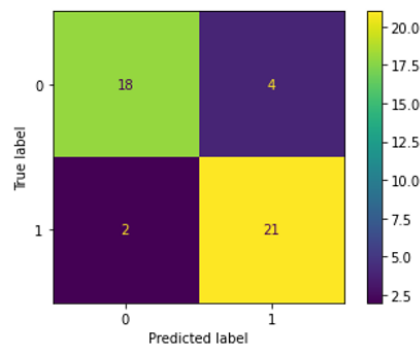
Berdasarkan Tabel 6 dapat diketahui bahwa model klasifikasi dikembangkan menggunakan *SVM* untuk sistem prediksi serangan jantung memiliki performa yang cukup tinggi meskipun tidak sampai

menyentuh angka 0.9 dimana range nilai dari performance metrics adalah antara 0 sampai 1.

Selain itu, model juga menghasilkan nilai dari setiap *performance metrics* yang sama. Hal ini menunjukkan bahwa data yang digunakan untuk proses pembelajaran dari model yang dikembangkan telah ternormalisasi dan penyebaran antar kelas pada setiap jenis data tersusun dengan baik. Sedangkan pada Gambar 5 dapat disimpulkan bahwa prediksi yang paling akurat yaitu pada label 1 dan prediksi yang paling buruk yaitu pada label 0, karena prediksi yang benar lebih sedikit dari prediksi yang salah. Analisis pada *Confusion Matrix* ini ditunjukkan dengan perbedaan warna, dimana warna kuning menunjukkan objek pengamatan yang paling baik.

Tabel 6. Hasil *Confusion Matrix Testing* Setelah Optimasi

No	Metric	Nilai
1	<i>Accuracy</i>	0.86
2	<i>Precision</i>	0.84
3	<i>Recall</i>	0.91
4	<i>F1 Score</i>	0.87



Gambar 5. Visualisasi *Confusion Matrix* dari Performa Model

#### 4. Kesimpulan

Setelah melakukan beberapa eksperimen dan mendapatkan hasil pengujian dari pengembangan model, maka dapat disimpulkan bahwa optimasi *hyperparameter gridsearchCV* pada sistem prediksi serangan jantung menggunakan SVM bekerja dengan performa yang baik karena memiliki nilai *accuracy* sebesar 86,0%, *precision* sebesar 84,0%, *recall* sebesar 91,0%, dan *f1-score* sebesar 87,0%.

Saran yang dapat diberikan untuk kemajuan pada penelitian selanjutnya berdasarkan hasil eksperimen yang dilakukan pada penelitian ini adalah melakukan metode pengumpulan data yang lebih meluas sehingga data yang digunakan untuk membangun model lebih beragam.

#### 5. Kontribusi Penulis

**Z. M. E. Darmawan:** *Conceptualization, Supervision, Formal Analysis, Investigation, Methodology, Software, Visualization, dan Writing – original draft.* **A. F. Dianta:** *Investigation, Methodology, Visualization, dan Writing – review & editing.*

#### 6. Declaration of Competing Interest

Penulis menyatakan tidak ada konflik kepentingan.

#### 7. Referensi

- Chowdhury, S., & Schoen, M. P. (2020). Research Paper Classification using Supervised Machine Learning Techniques. *2020 Intermountain Engineering, Technology and Computing (IETC)*, 1–6. <https://doi.org/10.1109/IETC47856.2020.9249211>
- Dwiasnati, S., & Devianto, Y. (2021). Optimasi Prediksi Bencana Banjir menggunakan Algoritma SVM untuk penentuan Daerah Rawan Bencana Banjir. *Seminar Nasional Sistem Informasi Dan Teknologi (SISFOTEK)*, 202–207.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
- Hovi, S. W., Hadiana, A. I., & Umbara, F. R. (2022). Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM). *Informatics Digital Expert*, 4(1), 40–45.
- Jumeilah, F. S. (2017). Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 1(1), 19–25. <https://doi.org/10.29207/resti.v1i1.11>

- Larassati, D., Zaidiah, A., & Afrizal, S. (2022). Sistem Prediksi Penyakit Jantung Koroner Menggunakan Metode Naïve Bayes. *Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika*, 7(2), 533–546.
- Maalouf, M., & Siddiqi, M. (2014). Weighted logistic regression for large-scale imbalanced and rare events data. *Knowledge-Based Systems*, 59, 142–148. <https://doi.org/10.1016/j.knosys.2014.01.012>
- Musyafaa, N., & Rifai, B. (2018). Model Support Vector Machine Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Liver. *Jurnal Ilmu Pengetahuan Dan Teknologi Komputer*, 3(2), 189–194.
- Nelli, F. (2018). *Python Data Analytics: With Pandas, NumPy, and Matplotlib* (2nd ed.). Apress.
- Nugraha, W. (2021). Prediksi Penyakit Jantung Cardiovascular Menggunakan Model Algoritma Klasifikasi. *Jurnal Sigmata*, 9(2), 78–84.
- Prasojo, M., Sriyanto, & Triwidianti, J. (2021). Prediksi Prestasi Siswa SMK Masuk Pasar Kerja Menggunakan Teknik Data Mining (Studi Kasus SMKN 1 Kota Agung Timur Tanggamus, Lampung) Margi Prasoj. *Seminar Nasional Darmajaya*.
- ProgrammerSought. (2018). *Lenet-5, Alexnet detailed explanation and tensorflow code implementation*. Programmer Sought. <https://www.programmersought.com/article/54943696781/>
- Sari, A. N., & Alfionita, S. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Naïve Bayes. *Jurnal Analisa Metode Rekayasa Informatika*, 1(1), 22–26.
- Siji George C. G., & B. Sumathi. (2020). Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction. *International Journal of Advanced Computer Science and Applications*, 11(9). <https://doi.org/10.14569/IJACSA.2020.0110920>
- Srivastava, D. K., & Bhambhu, L. (2010). Data Classification Using Support Vector Machine. *Journal of Theoretical and Applied Information Technology*, 12(1), 1–7.
- Toha, A., Purwono, & Gata, W. (2022). Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV. *Buletin Ilmiah Sarjana Teknik Elektro*, 4(1), 12–21.
- Zuliyanti, A. (2017). *Deteksi Penyakit Jantung Menggunakan Algoritma Klasifikasi Data Mining Naive Bayes*. Universitas Islam Nahdlatul Ulama.