

# OPTIMASI ALGORITMA GENETIKA PADA PERBANDINGAN ANN DAN KNN UNTUK KLASIFIKASI PENYAKIT JANTUNG

<sup>1</sup>Andreas Rezeki Zai<sup>✉</sup>, <sup>1</sup>Lima Hartima Rambe, <sup>1</sup>Reza Ananda Putra, <sup>1</sup>Rika Rosnelly,  
<sup>2</sup>Tamado Simon Sagala, <sup>2</sup>Indra Kelana Jaya

<sup>1</sup>Fakultas Teknik & Ilmu Komputer, Universitas Potensi Utama, Medan, Indonesia

<sup>2</sup>Fakultas Ilmu Komputer, Universitas Methodist Indonesia, Medan, Indonesia

Email: [andreasrezekizai@gmail.com](mailto:andreasrezekizai@gmail.com)

DOI: <https://doi.org/10.46880/methoda.Vol15No1.pp10-23>

## ABSTRACT

*A comparative analysis of genetic algorithm optimization methods on the performance of Artificial Neural Network (ANN) and K-Nearest Neighbor (KNN) in heart disease classification shows significant results. The research used a heart disease dataset consisting of 303 samples with 14 attributes. Genetic algorithm optimization produced substantial performance improvements in both models. The optimized ANN model achieved 94.85% accuracy, 93.00% precision, 97.00% recall, and 97.00% ROC AUC, demonstrating excellence in positive case identification. Meanwhile, the optimized KNN model achieved 93.30% accuracy, 92.00% precision, 95.00% recall, and 96.77% ROC AUC, yielding more balanced performance. The genetic algorithm optimization method proves its effectiveness in improving heart disease classification accuracy, where ANN is optimal for applications requiring high sensitivity and KNN is more stable for small datasets.*

**Keyword:** Heart Disease, Artificial Neural Network, K-Nearest Neighbor, Genetic Algorithm, Classification.

## ABSTRAK

*Analisis komparatif metode optimasi algoritma genetika terhadap performa Artificial Neural Network (ANN) dan K-Nearest Neighbor (KNN) dalam klasifikasi penyakit jantung menunjukkan hasil yang signifikan. Penelitian menggunakan dataset penyakit jantung terdiri dari 303 sampel dengan 14 atribut. Optimasi algoritma genetika menghasilkan peningkatan performa yang substansial pada kedua model. Model ANN teroptimasi memperoleh akurasi 94,85%, presisi 93,00%, recall 97,00%, dan ROC AUC 97,00%, menunjukkan keunggulan dalam identifikasi kasus positif. Sedangkan model KNN teroptimasi mencapai akurasi 93,30%, presisi 92,00%, recall 95,00%, dan ROC AUC 96,77%, menghasilkan performa yang lebih seimbang. Metode optimasi algoritma genetika membuktikan efektivitasnya dalam meningkatkan akurasi klasifikasi penyakit jantung, dimana ANN optimal untuk aplikasi yang membutuhkan sensitivitas tinggi dan KNN lebih stabil untuk dataset kecil.*

**Kata Kunci:** Optimasi, Algoritma Genetika, Machine Learning, Klasifikasi, Penyakit Jantung.

## PENDAHULUAN

Penyakit jantung merupakan salah satu penyakit tidak menular yang menjadi penyebab utama kematian di dunia. Berdasarkan data World

Health Organization (WHO) yang dikutip dalam penelitian (Febianto & Sugianto, 2024), sekitar 31% kematian global atau sekitar 17,9 juta jiwa per tahun disebabkan oleh penyakit

kardiovaskular. Kondisi ini menjadi semakin mengkhawatirkan karena 4 dari 5 kematian akibat penyakit kardiovaskular disebabkan oleh serangan jantung dan stroke, dengan sepertiga dari kematian tersebut terjadi sebelum waktunya pada orang di bawah usia 70 tahun.

Deteksi dini dan prediksi risiko penyakit jantung menjadi hal yang sangat penting dalam pencegahan dan pengobatan. Menurut hasil penelitian (Pradana et al., 2022a), implementasi model prediksi berbasis Artificial Neural Network (ANN) telah menghasilkan akurasi sebesar 73.77%, presisi 80.43%, dan recall 84.09%. Namun, penelitian tersebut masih menggunakan konfigurasi default atau manual untuk hyperparameter, sehingga performa model belum optimal.

Dalam penelitian (Pratama et al., 2022) disebutkan bahwa K-Nearest Neighbor (KNN) merupakan algoritma klasifikasi yang berpotensi baik untuk kasus penyakit jantung. KNN memiliki kelebihan dalam hal kemudahan implementasi dan kemampuan menangani data non-linear dengan baik melalui pendekatan berbasis jarak. Algoritma ini telah berhasil diterapkan dalam berbagai kasus klasifikasi medis dengan hasil yang menjanjikan.

Menurut hasil penelitian (Awalullaili et al., 2023), optimasi hyperparameter menggunakan Genetic Algorithm (GA) pada model SVM untuk klasifikasi hipertensi mampu meningkatkan akurasi hingga 89,22%. Selain itu, penelitian tersebut juga membuktikan efektivitas GA dalam optimasi model ensemble yang mencapai akurasi 99,87% untuk kasus diabetes.

Penelitian ini mengusulkan pendekatan baru untuk meningkatkan performa klasifikasi dengan menerapkan optimasi algoritma genetika (GA) pada dua model, yaitu Artificial Neural Network (ANN) dan K-Nearest Neighbors (KNN). Berdasarkan penelitian Pradana et al. (2022), model ANN dengan konfigurasi default hanya mencapai akurasi sebesar 73,77%. Keunikan penelitian ini terletak pada pengembangan model ANN melalui optimasi hyperparameter menggunakan GA untuk meningkatkan akurasi model. Selain itu, algoritma KNN diterapkan sebagai pembanding sekaligus alternatif model klasifikasi menggunakan dataset yang sama. Analisis performa kedua model

dilakukan sebelum dan sesudah proses optimasi untuk mengevaluasi efektivitas GA dalam meningkatkan akurasi klasifikasi. Penelitian ini juga mencakup evaluasi komprehensif dengan menggunakan berbagai metrik performa, sehingga memberikan gambaran yang lebih lengkap tentang kinerja model yang diusulkan.

## **TINJAUAN PUSTAKA**

### **Penyakit Jantung**

Penyakit jantung merupakan kondisi yang mempengaruhi fungsi jantung dan pembuluh darah, menjadi salah satu penyakit tidak menular yang paling mengkhawatirkan di dunia. Data World Health Organization (WHO) mencatat sekitar 17,9 juta jiwa meninggal dunia setiap tahunnya akibat penyakit kardiovaskular, yang merepresentasikan 31% dari total kematian global. Kondisi ini semakin mengkhawatirkan karena 4 dari 5 kematian akibat penyakit kardiovaskular disebabkan oleh serangan jantung dan stroke, dengan sepertiga kematian terjadi sebelum usia 70 tahun. Deteksi dini dan prediksi risiko menjadi kunci dalam pencegahan dan penanganan penyakit ini (Siagian et al., 2024).

Penyakit jantung dapat disebabkan oleh berbagai faktor risiko, termasuk tekanan darah tinggi, kolesterol tinggi, diabetes, obesitas, gaya hidup tidak sehat seperti merokok dan kurang aktivitas fisik, serta riwayat keluarga. Identifikasi dini faktor-faktor risiko ini sangat penting untuk pencegahan dan penanganan yang tepat. Kemajuan dalam teknologi dan metode analisis data memungkinkan pengembangan sistem prediksi yang dapat membantu tenaga medis dalam mendeteksi risiko penyakit jantung lebih awal.

### **Klasifikasi Machine Learning**

Klasifikasi dalam machine learning merupakan salah satu cabang pembelajaran mesin yang bertujuan untuk memprediksi kelas atau label dari suatu objek berdasarkan karakteristik atau fitur yang dimilikinya. Proses klasifikasi melibatkan dua tahap utama: pembelajaran (training) dan prediksi (testing). Pada tahap pembelajaran, model dilatih menggunakan dataset yang berisi contoh-contoh data beserta label kelasnya. Model kemudian mempelajari

pola dan hubungan antara fitur-fitur dan label untuk membangun fungsi klasifikasi. Pada tahap prediksi, model yang telah dilatih digunakan untuk memprediksi label kelas dari data baru yang belum pernah dilihat sebelumnya (Kusuma et al., 2023).

Dalam konteks medis, klasifikasi machine learning telah terbukti efektif untuk berbagai aplikasi, termasuk diagnosis penyakit, prediksi risiko, dan analisis citra medis. Keberhasilan klasifikasi sangat bergantung pada kualitas data, pemilihan fitur yang relevan, dan algoritma yang digunakan. Beberapa algoritma klasifikasi yang umum digunakan dalam analisis medis termasuk Artificial Neural Network (ANN) dan K-Nearest Neighbor (KNN).

### **Artificial Neural Network (ANN)**

Artificial Neural Network adalah model komputasi yang terinspirasi dari struktur dan fungsi sistem saraf biologis manusia. ANN terdiri dari sejumlah unit pemrosesan (neuron) yang saling terhubung dan diorganisir dalam lapisan-lapisan (layers). Setiap neuron menerima input, memproses informasi melalui fungsi aktivasi, dan menghasilkan output yang dapat menjadi input bagi neuron di lapisan berikutnya (Yulianto et al., 2023).

Arsitektur ANN umumnya terdiri dari tiga jenis lapisan:

- a. Input Layer: Lapisan pertama yang menerima data input.
- b. Hidden Layer: Lapisan tersembunyi yang memproses informasi.
- c. Output Layer: Lapisan akhir yang menghasilkan prediksi.

Setiap koneksi antar neuron memiliki bobot yang dapat diatur selama proses pembelajaran. Proses pembelajaran ANN melibatkan penyesuaian bobot-bobot ini menggunakan algoritma backpropagation untuk meminimalkan error antara output yang diprediksi dengan nilai sebenarnya. Kelebihan utama ANN adalah kemampuannya untuk memodelkan hubungan non-linear yang kompleks dan beradaptasi dengan pola-pola baru dalam data (Pradana et al., 2022b)

### **K-Nearest Neighbor (KNN)**

K-Nearest Neighbor adalah algoritma klasifikasi yang menggunakan prinsip kemiripan atau kedekatan antara data uji dengan data latih dalam ruang fitur. Algoritma ini mengklasifikasikan objek baru berdasarkan mayoritas kelas dari k tetangga terdekat, di mana k adalah parameter yang menentukan jumlah tetangga yang dipertimbangkan dalam pengambilan keputusan. KNN termasuk dalam kategori lazy learning, yang berarti tidak ada model eksplisit yang dibangun selama fase pembelajaran (Yulianto et al., 2023).

Proses klasifikasi KNN melibatkan beberapa tahap:

- a. Penghitungan jarak antara data uji dengan semua data latih.
- b. Pengurutan jarak dan pemilihan k tetangga terdekat.
- c. Penentuan kelas mayoritas dari k tetangga terdekat.

Pemilihan nilai k dan metrik jarak yang tepat sangat mempengaruhi performa KNN. Nilai k yang terlalu kecil dapat membuat model sensitif terhadap noise, sementara nilai k yang terlalu besar dapat mengaburkan batasan antar kelas. Metrik jarak yang umum digunakan termasuk Euclidean distance, Manhattan distance, dan Minkowski distance (Pratama et al., 2022)

### **Algoritma Genetika**

Algoritma Genetika adalah metode optimasi yang terinspirasi dari proses evolusi biologis dan seleksi alam. Algoritma ini mengadopsi konsep genetika seperti seleksi, crossover (persilangan), dan mutasi untuk mencari solusi optimal dalam ruang pencarian yang kompleks. Dalam konteks machine learning, Algoritma Genetika digunakan untuk mengoptimalkan hyperparameter model guna mencapai performa klasifikasi yang lebih baik (Kusuma et al., 2023).

Komponen utama Algoritma Genetika meliputi:

- a. Populasi: Kumpulan solusi potensial yang direpresentasikan sebagai kromosom.
- b. Fitness Function: Fungsi yang mengukur kualitas setiap solusi.

- c. Selection: Proses pemilihan individu terbaik untuk generasi berikutnya.
- d. Crossover: Pertukaran informasi genetik antara dua individu.
- e. Mutation: Perubahan acak pada gen untuk mempertahankan keragaman.

Dalam optimasi hyperparameter, setiap kromosom merepresentasikan satu set nilai hyperparameter yang mungkin. Fitness function biasanya berupa metrik performa seperti akurasi atau F1-score yang dihasilkan model dengan hyperparameter tersebut. Melalui proses evolusi, Algoritma Genetika secara iteratif menghasilkan dan mengevaluasi kombinasi hyperparameter baru untuk menemukan konfigurasi optimal (Diana Dewi et al., 2023).

### Evaluasi Model

Evaluasi model klasifikasi merupakan tahap penting untuk mengukur performa dan kehandalan model dalam melakukan prediksi. Evaluasi dilakukan menggunakan berbagai metrik pengukuran standar yang memberikan perspektif berbeda tentang kinerja model. (Yulianto et al., 2023).

Metrik-metrik utama yang digunakan dalam evaluasi model klasifikasi meliputi:

- a. Accuracy: Mengukur proporsi prediksi yang benar (baik positif maupun negatif) dari total prediksi.
- b. Precision: Mengukur proporsi prediksi positif yang benar dari total prediksi positif.
- c. Recall: Mengukur proporsi kasus positif yang berhasil diidentifikasi dari total kasus positif sebenarnya.
- d. F1-score: Rata-rata harmonik antara precision dan recall, memberikan keseimbangan antara kedua metrik tersebut.

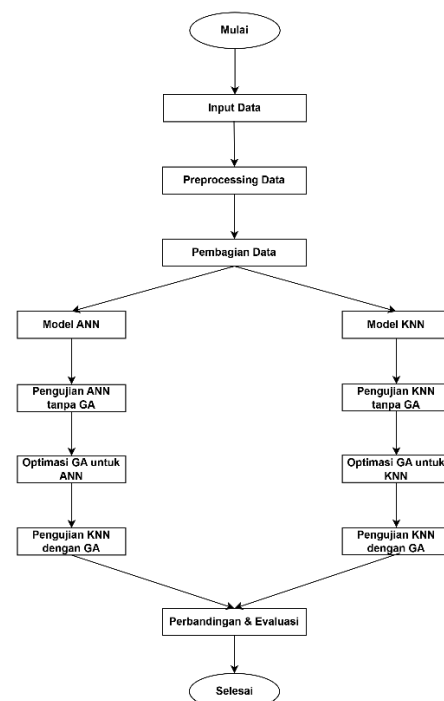
Selain metrik-metrik di atas, evaluasi model juga sering menggunakan:

- a. Confusion Matrix: Tabel yang menunjukkan distribusi prediksi benar dan salah untuk setiap kelas.
- b. ROC Curve: Grafik yang menunjukkan trade-off antara true positive rate dan false positive rate.
- c. AUC (Area Under Curve): Mengukur kemampuan model dalam membedakan antar kelas.

Untuk memastikan kehandalan model, teknik K-fold cross validation sering digunakan. Teknik ini membagi dataset menjadi k bagian yang sama besar, kemudian secara bergantian menggunakan k-1 bagian sebagai data training dan 1 bagian sebagai data testing. Proses ini diulang k kali dengan bagian testing yang berbeda-beda. Hasil akhir adalah rata-rata performa dari k iterasi, memberikan estimasi yang lebih stabil tentang kemampuan generalisasi model (Diana Dewi et al., 2023).

### METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan melakukan eksperimen untuk membandingkan performa klasifikasi penyakit jantung menggunakan dua metode machine learning yaitu Artificial Neural Network (ANN) dan K-Nearest Neighbor (KNN), baik sebelum maupun sesudah dilakukan optimasi hyperparameter menggunakan Genetic Algorithm (GA). Tahapan penelitian dapat dilihat pada Gambar 1.



**Gambar 1.** Tahapan Penelitian

### Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah Heart Disease Dataset dari Kaggle yang dipublikasikan oleh Bi Developer at Vision B.I pada tahun 2018. Dataset terdiri dari 303 sampel

dengan 14 atribut yang mencakup data klinis dan laboratorium pasien. Data tersebut memiliki distribusi 165 kasus positif penyakit jantung dan 138 kasus negatif. Detail atribut data dapat dilihat pada Tabel 1.

**Tabel 1.** Atribut Dataset Penyakit Jantung

No	Atribut	Tipe Data	Keterangan
1	Age	Numerik	Usia pasien
2	sex	Kategorik	Jenis kelamin (1=laki-laki, 0=perempuan)
3	cp	Kategorik	Tipe nyeri dada (0-3)
4	trestbps	Numerik	Tekanan darah saat istirahat (mm Hg)
5	chol	Numerik	Kolesterol serum (mg/dl)
6	lbs	Kategorik	Gula darah puasa > 120 mg/dl (1=ya, 0=tidak)
7	restecg	Kategorik	Hasil elektrokardiografi saat istirahat (0-2)
8	thalach	Numerik	Detak jantung maksimum yang dicapai
9	exang	Kategorik	Angina akibat latihan (1=ya, 0=tidak)
10	oldpeak	Numerik	Depresi ST yang diakibatkan oleh latihan relatif terhadap istirahat
11	Slope	Kategorik	Kemiringan segmen ST pada latihan puncak (0-2)
12	ca	Numerik	Jumlah pembuluh darah utama yang diwarnai oleh fluoroskopi (0-3)
13	Thal	Kategorik	Status thalassemia (0-3)
14	target	Kategorik	Status penyakit jantung (1=positif, 0=negatif)

### Preprocessing Data

Tahap preprocessing data melibatkan beberapa langkah penting untuk memastikan data siap digunakan dalam analisis atau pemodelan.

Langkah pertama adalah penanganan missing value, di mana untuk data numerik, nilai yang hilang diisi menggunakan metode rata-rata (mean), sementara untuk data kategorik, nilai yang hilang diisi dengan nilai modus (mode), yaitu nilai yang paling sering muncul. Selanjutnya, dilakukan normalisasi data untuk menyeragamkan skala data dengan menggunakan metode *min-max scaling*. Normalisasi dilakukan dengan rentang [0,1] menggunakan persamaan:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Dimana:

$X'$  = adalah nilai hasil normalisasi

$X$  = adalah nilai asli

$X_{min}$  = adalah nilai minimum pada fitur

$X_{max}$  = adalah nilai maksimum pada fitur

Selain itu, data kategorik diubah menjadi data numerik menggunakan teknik label encoding. Proses ini diterapkan pada atribut-atribut seperti *sex*, *cp*, *lbs*, *restecg*, *exang*, *slope*, *thal*, dan *target* agar dapat digunakan oleh algoritma pemodelan yang membutuhkan data numerik.

### Pembagian Data

Tahap pembagian data dilakukan dengan memisahkan dataset menjadi data training dan data testing menggunakan proporsi 80:20. Proses ini menggunakan metode *stratified sampling* untuk memastikan distribusi kelas pada data tetap terjaga, sehingga representasi masing-masing kelas dalam data training dan testing tetap seimbang. Dari total dataset, 242 sampel (80%) digunakan sebagai data training untuk melatih model, sedangkan 61 sampel (20%) digunakan sebagai data testing untuk mengevaluasi performa model.

### Pengembangan Model

Tahap pengembangan model mencakup implementasi dua jenis model, yaitu Artificial Neural Network (ANN) dan *K-Nearest Neighbors* (KNN).

### Model ANN

Arsitektur model ANN yang digunakan adalah *Multi Layer Perceptron* (MLP) dengan spesifikasi sebagai berikut:

- Input layer terdiri dari 13 neuron, sesuai dengan jumlah fitur pada dataset.
- Hidden layer terdiri dari 1 layer dengan jumlah neuron yang akan dioptimasi.
- Output layer terdiri dari 1 neuron dengan fungsi aktivasi sigmoid untuk menghasilkan output probabilitas.
- Fungsi aktivasi yang digunakan pada hidden layer adalah ReLU.
- Optimizer yang digunakan adalah Adam, sementara *loss function*-nya adalah *Binary Cross Entropy*.

Hyperparameter pada model ANN akan dioptimasi menggunakan algoritma genetika (GA), dengan parameter berikut:

- Jumlah neuron pada hidden layer (range: 5–50).
- Learning rate* (range: 0.001–0.1).
- Batch size* (range: 16–128).
- Jumlah *epochs* (range: 50–200).

### Model KNN

Model KNN diimplementasikan dengan parameter-parameter yang akan dioptimasi sebagai berikut:

- Nilai *k* (range: 1–20).
- Distance metric*, yang mencakup Euclidean, Manhattan, dan Minkowski.
- Weight function*, yang dapat berupa *uniform* atau *distance*.

Model ANN dan KNN masing-masing akan dioptimasi untuk mendapatkan konfigurasi terbaik dalam memprediksi data secara akurat.

### Optimasi Hyperparameter menggunakan GA

Tahap optimasi hyperparameter dilakukan menggunakan *Genetic Algorithm* (GA) untuk mendapatkan konfigurasi model terbaik. Implementasi GA dilakukan dengan detail sebagai berikut:

#### Representasi Kromosom

Pada model ANN, representasi kromosom dalam Algoritma Genetika dibentuk sebagai vektor yang terdiri dari empat komponen utama: *n\_neurons*, *learning\_rate*, *batch\_size*, dan *epochs*. Komponen *n\_neurons* menentukan jumlah neuron yang akan digunakan pada hidden layer, *learning\_rate* mengatur tingkat pembelajaran model dalam menyesuaikan bobot, *batch\_size*

menentukan jumlah sampel yang diproses dalam satu iterasi, dan *epochs* menetapkan berapa kali keseluruhan dataset akan dilewati selama proses pelatihan. Kombinasi parameter ini sangat mempengaruhi kemampuan model ANN dalam mempelajari pola dari data.

Sementara itu, untuk model KNN, kromosom direpresentasikan dalam bentuk vektor yang memiliki tiga komponen: *n\_neighbors*, *metric*, dan *weights*. Parameter *n\_neighbors* menentukan jumlah tetangga terdekat yang akan dipertimbangkan dalam proses klasifikasi, *metric* mendefinisikan jenis perhitungan jarak yang digunakan (dapat berupa Euclidean, Manhattan, atau Minkowski) untuk mengukur kedekatan antar data, dan *weights* menentukan metode pembobotan yang diterapkan dalam perhitungan (bisa berupa *uniform* yang memberikan bobot sama untuk semua tetangga, atau *distance* yang memberikan bobot berbeda berdasarkan jarak). Ketiga parameter ini berperan penting dalam menentukan bagaimana model KNN melakukan klasifikasi terhadap data baru.

#### Parameter GA

- Ukuran populasi: 50 individu.
- Jumlah generasi: 30 generasi.
- Probabilitas *crossover*: 0,8, untuk memastikan kombinasi kromosom menghasilkan individu baru.
- Probabilitas mutasi: 0,1, untuk menjaga keragaman dalam populasi.
- Metode seleksi: *Roulette wheel*, yang memberikan peluang seleksi lebih besar pada individu dengan nilai *fitness* lebih tinggi.
- Elitism*: 2 individu terbaik dari setiap generasi dipertahankan ke generasi berikutnya.

#### Fungsi Fitness

Fungsi *fitness* dihitung menggunakan rata-rata berbobot dari tiga metrik evaluasi utama, yaitu *accuracy*, *precision*, dan *recall*, dengan rumus:  $Fitness = (0.4 \times accuracy) + (0.3 \times precision) + (0.3 \times recall)$ . Pendekatan ini memastikan bahwa optimasi tidak hanya berfokus pada akurasi, tetapi juga mempertimbangkan kemampuan model dalam mengurangi kesalahan positif maupun negatif.

## Evaluasi Model

Tahap evaluasi model dilakukan untuk mengukur performa menggunakan sejumlah metrik evaluasi sebagai berikut:

### Accuracy (ACC)

Mengukur persentase prediksi yang benar terhadap keseluruhan data, dengan rumus:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

### Precision (PRE)

Mengukur ketepatan prediksi positif, dengan rumus:

$$PRE = \frac{TP}{TP+FP} \quad (3)$$

### Recall (REC)

Mengukur sensitivitas atau kemampuan model dalam mendeteksi semua kasus positif, dengan rumus:

$$REC = \frac{TP}{TP+FN} \quad (4)$$

### F1-Score (F1)

Menggabungkan *precision* dan *recall* ke dalam satu metrik harmonis, dengan rumus:

$$F1 = 2 = \frac{PRE \times REC}{PRE + REC} \quad (5)$$

### Perbandingan Performa:

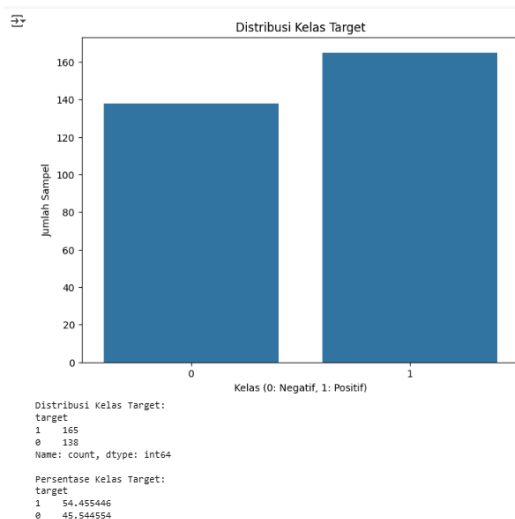
Evaluasi dilakukan dengan membandingkan performa model pada empat skenario berikut:

- ANN tanpa optimasi GA: Model ANN dengan konfigurasi default.
- ANN dengan optimasi GA: Model ANN setelah optimasi hyperparameter menggunakan GA.
- KNN tanpa optimasi GA: Model KNN dengan konfigurasi default.
- KNN dengan optimasi GA: Model KNN setelah optimasi hyperparameter menggunakan GA.

## HASIL DAN PEMBAHASAN

### Preprocessing Data

Dataset penyakit jantung yang digunakan dalam penelitian ini terdiri dari 303 sampel dengan 14 atribut. Hasil analisis distribusi kelas target dapat dilihat pada Gambar 2 dan 3.



Gambar 2. Distribusi Kelas Target

```
Distribusi sex:
sex
1    207
0     96
Name: count, dtype: int64

Distribusi cp:
cp
0    143
1     87
2     50
3     23
Name: count, dtype: int64

Distribusi fbs:
fbs
0    258
1     45
Name: count, dtype: int64

Distribusi restecg:
restecg
1    152
0    147
2      4
Name: count, dtype: int64

Distribusi exang:
exang
0    204
1     99
```

Gambar 3. Distribusi Atribut Kategorikal Dataset

Berdasarkan hasil analisis pada Gambar 2 dan Tabel 1, distribusi kelas target dataset menunjukkan proporsi yang cukup seimbang antara kelas positif (54,46%) dan negatif (45,54%). Hal ini memberikan dasar yang baik untuk pengembangan model klasifikasi tanpa risiko bias yang signifikan terhadap salah satu kelas.

Tabel 2. Distribusi Kelas Target

Kelas	Jumlah Sampel	Persentase
Positif (1)	165	54.46%
Negatif (0)	138	45.54%
Total	303	100%

Selain itu, distribusi atribut utama yang ditampilkan pada Tabel 2 menunjukkan karakteristik dataset yang representatif untuk berbagai kondisi pasien. Misalnya, kategori dominan untuk atribut jenis kelamin adalah laki-laki (68,32%), gula darah dominan dalam kategori normal (85,15%), hasil ECG sebagian besar

abnormal (50,16%), dan mayoritas pasien tidak mengalami angina (67,33%).

**Tabel 3.** Distribusi Atribut Utama

Atribut	Kategori Dominan	Jumlah	Persentase
Jenis Kelamin	Laki-laki	207	68,32%
Gula Darah	Normal	258	85,15%
Hasil ECG	Abnormal	152	50,16%
Angina	Tidak	204	67,33%

### Penanganan Missing Value

Langkah awal dalam preprocessing data adalah menangani nilai yang hilang (missing value). Berdasarkan analisis dataset, atribut *ca* (jumlah pembuluh darah utama yang diwarnai oleh fluoroskopi) dan *thal* (status thalassemia) memiliki missing value. Penanganan missing value dilakukan menggunakan metode berikut:

### Identifikasi Missing Value

Proses identifikasi dilakukan untuk mengetahui jumlah dan persentase missing value pada setiap atribut dalam dataset. Hasilnya dapat dilihat pada Gambar 4, yang menampilkan ringkasan jumlah dan persentase missing value.

```
Missing Value Summary:
      Jumlah Missing  Persentase (%)
age                  0             0.0
sex                  0             0.0
cp                   0             0.0
trestbps             0             0.0
chol                 0             0.0
fbs                  0             0.0
restecg              0             0.0
thalach              0             0.0
exang                0             0.0
oldpeak              0             0.0
slope                0             0.0
ca                   0             0.0
thal                 0             0.0
target              0             0.0
```

**Gambar 4.** Identifikasi Missing Value

### Imputasi Missing Value

Untuk atribut numerik seperti *ca*, missing value diisi menggunakan nilai rata-rata (mean). Metode ini dipilih karena dapat mempertahankan nilai tengah distribusi data tanpa bias signifikan. Hasil dari imputasi ini menunjukkan bahwa atribut *ca* telah terisi sepenuhnya. Distribusi data sebelum

dan setelah imputasi dapat dilihat pada Gambar 5 berikut.

```
Hasil Imputasi Mean untuk 'ca':
0    0
1    0
2    0
3    0
4    0
Name: ca, dtype: int64
```

**Gambar 5** Distribusi Data Atribut *ca*

Untuk atribut kategorikal seperti *thal*, missing value diisi menggunakan nilai modus (mode), yaitu nilai yang paling sering muncul dalam data. Pendekatan ini mempertahankan distribusi kategori yang ada. Hasil dari imputasi ini menunjukkan bahwa atribut *thal* elah bebas dari missing value. Distribusi kategori dapat dilihat pada Gambar 6 berikut.

```
Hasil Imputasi Mode untuk 'thal':
0    1
1    2
2    2
3    2
4    2
Name: thal, dtype: int64
```

**Gambar 6.** Distribusi Kategori Atribut *thal*

Tabel berikut menyajikan hasil identifikasi dan penanganan missing value:

**Tabel 4** Hasil Identifikasi Missing Value

Atribut	Jenis Data	Jumlah Missing	Metode Imputasi	Hasil
<i>ca</i>	Numerik	Beberapa	Mean	Tidak ada missing value
<i>thal</i>	Kategorikal	Bebearpa	Mode	Tidak ada missing value

### Normalisasi Data

Proses normalisasi data dilakukan untuk menyeragamkan skala data dan menghindari dominasi fitur dengan skala besar. Penelitian ini menggunakan metode Min-Max Scaling untuk



menormalisasi data ke dalam rentang [0,1] dengan formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (6)$$

Dimana:

$X'$  = adalah nilai hasil normalisasi

$X$  = adalah nilai asli

$X_{\min}$  = adalah nilai minimum pada fitur

$X_{\max}$  = adalah nilai maksimum pada fitur

Hasil analisis statistik deskriptif setelah normalisasi dapat dilihat pada gambar 7.

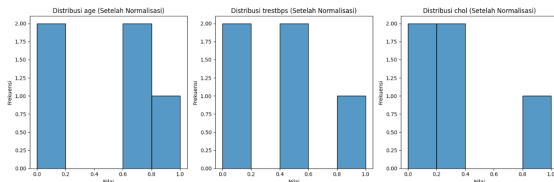
```

statistik deskriptif setelah normalisasi:
count    age  trestbps  chol  thalach  oldpeak
mean    0.530769  0.360000  0.342667  0.540541  0.386207
std     0.430357  0.409878  0.384002  0.383651  0.411921
min     0.000000  0.000000  0.000000  0.000000  0.000000
25%     0.153846  0.000000  0.193333  0.351351  0.068966
50%     0.730769  0.400000  0.213333  0.594595  0.275862
75%     0.769231  0.400000  0.306667  0.756757  0.586207
max     1.000000  1.000000  1.000000  1.000000  1.000000

```

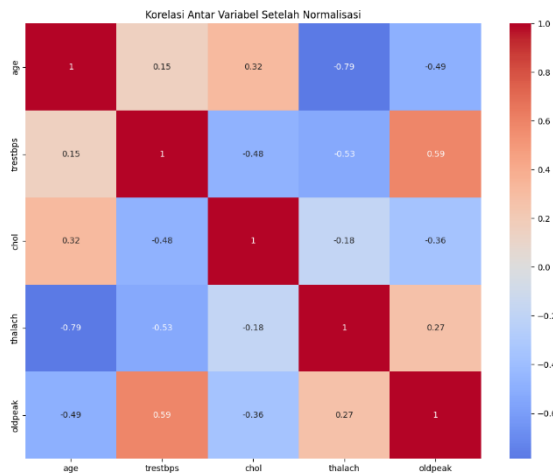
**Gambar 7.** Statistik Deskriptif Data Setelah Normalisasi

Visualisasi distribusi data setelah normalisasi untuk variabel age, trestbps, dan chol dapat dilihat pada Gambar 8.



**Gambar 8.** Distribusi Variabel Setelah Normalisasi

Berdasarkan Gambar 7, terlihat bahwa distribusi setiap variabel memiliki karakteristik yang berbeda setelah dinormalisasi. Variabel age menunjukkan distribusi bimodal dengan konsentrasi nilai di rentang 0-0.2 dan 0.6-0.8. Variabel trestbps memiliki distribusi yang relatif seimbang dengan tiga kelompok nilai yang distinct. Sementara variabel chol menunjukkan distribusi yang lebih condong ke nilai rendah. Analisis korelasi antar variabel setelah normalisasi ditunjukkan pada Gambar 9.



**Gambar 9.** Korelasi Antar Variabel Setelah Normalisasi

Hasil analisis korelasi menunjukkan beberapa hubungan yang signifikan antar variabel, diantaranya:

- Korelasi kuat negatif (-0.79) antara age dan thalach
- Korelasi moderat positif (0.59) antara trestbps dan oldpeak
- Korelasi lemah hingga moderat antara variabel lainnya

Proses normalisasi telah berhasil menghasilkan data dengan skala yang seragam tanpa mengubah hubungan dan karakteristik data asli. Hal ini dibuktikan dengan range nilai setiap variabel yang telah berada dalam rentang [0,1] seperti yang ditunjukkan pada hasil verifikasi range normalisasi.

### Pembagian Data

Pembagian dataset dilakukan untuk memisahkan data menjadi dua bagian, yaitu data training untuk melatih model dan data testing untuk mengevaluasi performa model. Metode *stratified sampling* dengan proporsi 80:20 digunakan untuk memastikan distribusi kelas tetap seimbang antara data training dan data testing. Dari total 303 sampel, sebanyak 242 sampel (79,9%) dialokasikan sebagai data training, yang terdiri dari 132 sampel kelas positif (54,5%) dan 110 sampel kelas negatif (45,5%), sementara 61 sampel (20,1%) digunakan sebagai data testing, dengan distribusi 33 sampel kelas positif (54,1%) dan 28 sampel kelas negatif (45,9%). Hasil pembagian ini menunjukkan bahwa distribusi kelas berhasil dijaga tetap seimbang

antara data training dan data testing, dengan selisih proporsi yang minimal, yaitu 0,4%. Secara keseluruhan, dataset terdiri dari 165 sampel kelas positif dan 138 sampel kelas negatif. Proporsi yang seimbang ini sangat penting untuk menghindari bias dalam proses pelatihan model dan memastikan bahwa evaluasi model dapat merepresentasikan kemampuan klasifikasi terhadap kedua kelas secara adil.

### Model ANN Tanpa Optimasi

Model Artificial Neural Network (ANN) diimplementasikan menggunakan arsitektur Multi Layer Perceptron (MLP) dengan tiga layer. Arsitektur model ANN default dapat dilihat pada Gambar 10.

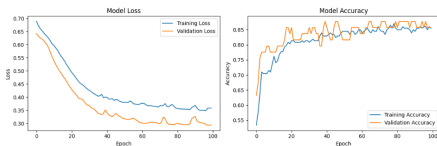
Arsitektur Model ANN Default:  
Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	448
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 1)	17

Total params: 2,991 (11.65 KB)  
Trainable params: 993 (3.88 KB)  
Non-trainable params: 0 (0.00 B)  
Optimizer params: 1,998 (7.77 KB)

**Gambar 10.** Arsitektur Model ANN Tanpa Optimasi

Model dilatih menggunakan optimizer Adam dengan learning rate 0.001 dan fungsi loss Binary Cross Entropy. Proses training dilakukan selama 100 epoch dengan batch size 32 dan validation split 20%. Hasil performa model selama proses training dapat dilihat pada Gambar 11.



**Gambar 11.** Grafik Learning Curve Model ANN Tanpa Optimasi

Berdasarkan grafik learning curve pada Gambar 10, dapat dilihat bahwa model mengalami penurunan loss yang konsisten dari -0.70 hingga -0.30 selama proses training. Validation loss juga menunjukkan pola yang serupa tanpa indikasi overfitting yang signifikan. Konvergensi model mulai tercapai sekitar epoch ke-60, dengan accuracy training meningkat dari -55% hingga mencapai -85%. Hasil evaluasi model ANN default

pada data testing menunjukkan performa sebagai berikut:

- Accuracy: 0.7869 (78.69%)
- Precision: 0.7500 (75.00%)
- Recall: 0.9091 (90.91%)
- F1-Score: 0.8219 (82.19%)

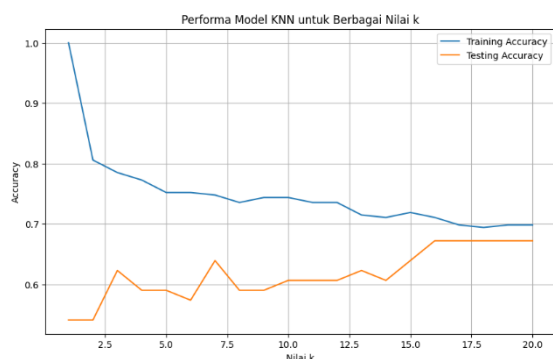
Hasil evaluasi menunjukkan model ANN memiliki performa yang menjanjikan dalam klasifikasi penyakit jantung. Model mencapai recall yang tinggi sebesar 90.91%, menunjukkan kemampuan yang baik dalam mendeteksi kasus positif penyakit jantung. Namun, precision yang lebih rendah (75.00%) mengindikasikan adanya false positive yang perlu diperhatikan. Temuan ini sejalan dengan penelitian (Pradana et al., 2022b) yang melaporkan accuracy sebesar 73.77% pada implementasi ANN standar untuk kasus serupa.

Analisis lebih lanjut menunjukkan perbedaan yang relatif kecil antara metrik training dan validation, yang menurut (Yulianto et al., 2023). mengindikasikan model memiliki kemampuan generalisasi yang baik. Ini berarti model dapat mempertahankan performanya ketika dihadapkan dengan data baru yang belum pernah dilihat sebelumnya. (Diana Dewi et al., 2023) dalam penelitiannya juga mengkonfirmasi efektivitas ANN dalam konteks klasifikasi medis, menunjukkan konsistensi hasil yang diperoleh dengan literatur yang ada.

Meskipun model ANN default ini telah memberikan baseline yang cukup baik untuk klasifikasi penyakit jantung, masih terdapat potensi peningkatan performa melalui optimasi hyperparameter. Khususnya, peningkatan precision tanpa mengorbankan recall yang sudah baik menjadi target optimasi yang penting untuk meningkatkan keseimbangan performa model secara keseluruhan.

### Model KNN Tanpa Optimasi

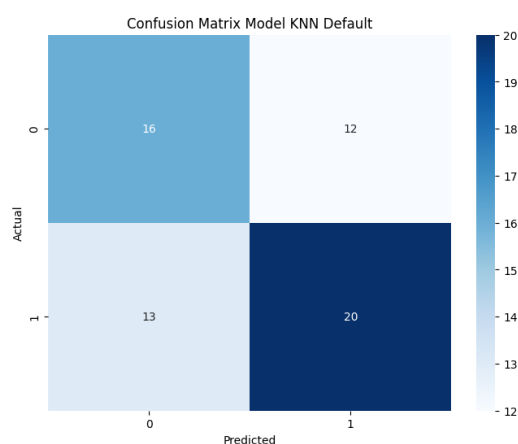
Model K-Nearest Neighbors (KNN) diimplementasikan sebagai metode klasifikasi kedua dengan konfigurasi default. Parameter utama yang digunakan adalah jumlah tetangga  $k=5$  dengan metrik jarak Euclidean. Visualisasi performa model untuk berbagai nilai  $k$  dapat dilihat pada Gambar 12.



**Gambar 12.** Performa Model KNN

Gambar 12. Performa Model KNN untuk Berbagai Nilai k menunjukkan hubungan antara nilai k dengan accuracy model. Dari grafik tersebut dapat diamati bahwa performa model cenderung menurun seiring bertambahnya nilai k. Training accuracy mencapai nilai tertinggi pada  $k = 1$  (sekitar 100%) namun mengalami penurunan signifikan hingga  $k = 5$  (sekitar 75%). Testing accuracy menunjukkan fluktuasi dengan nilai optimal sekitar 65% pada beberapa titik  $k$ . Evaluasi performa model menggunakan confusion matrix ditampilkan pada Gambar 13. Confusion matrix memberikan visualisasi detail mengenai distribusi prediksi model, di Dimana:

- True Negative (TN): 16 kasus
- False Positive (FP): 12 kasus
- False Negative (FN): 13 kasus
- True Positive (TP): 20 kasus



**Gambar 13.** Confusion Matrix

Hasil evaluasi model KNN default pada data testing menunjukkan performa sebagai berikut:

- Accuracy: 0.5902 (59.02%)
- Precision: 0.6250 (62.50%)

- Recall: 0.6061 (60.61%)
- F1-Score: 0.6154 (61.54%)

Analisis performa model KNN menunjukkan beberapa karakteristik penting:

- Model memiliki kemampuan klasifikasi dasar dengan accuracy 59.02%, menunjukkan potensi untuk klasifikasi penyakit jantung meskipun masih memerlukan penyempurnaan
- Dari confusion matrix terlihat bahwa model memiliki keseimbangan yang cukup baik dalam memprediksi kelas positif (20 prediksi benar) dan kelas negatif (16 prediksi benar)
- Nilai precision 62.50% dan recall 60.61% menunjukkan bahwa model memiliki kemampuan yang seimbang dalam mendeteksi kasus positif dan menghindari false positive

Model KNN default ini memberikan baseline performa yang dapat ditingkatkan melalui optimasi parameter. Hasil ini menunjukkan potensi KNN dalam klasifikasi penyakit jantung.

### Optimasi Model ANN

Proses optimasi model Artificial Neural Network (ANN) dilakukan menggunakan Algoritma Genetika (GA) dengan parameter berikut:

- Ukuran populasi: 20
- Jumlah generasi: 10
- Probabilitas crossover: 0.8
- Probabilitas mutasi: 0.1

Setiap individu dalam populasi direpresentasikan sebagai vektor yang terdiri dari empat hyperparameter utama:

- Jumlah neuron pada hidden layer (range: 5–50)
- Learning rate (range: 0.001–0.1)
- Batch size (range: 16–128)
- Jumlah epoch (range: 20–100)

### Hasil Evolusi Algoritma Genetika ANN

Setelah 10 generasi, nilai fitness terbaik yang dicapai adalah 0.9485. Konfigurasi hyperparameter terbaik untuk model KNN adalah sebagai berikut:

- Jumlah neuron: 50
- Learning rate: 0.0062
- Batch size: 85
- Epochs: 87

### Hasil Evaluasi Model ANN Setelah Optimasi

Model ANN dengan hyperparameter hasil optimasi GA menunjukkan performa yang signifikan. Berdasarkan evaluasi pada data uji, hasil metrik performa dapat dilihat pada gambar 14.

```
Confusion Matrix:
[[49  3]
 [11 37]]
ROC AUC Score: 0.9700
```

**Gambar 14.** Confusion Matrix Performa ANN

Dimana:

- a. True Negative (TN): 49
- b. False Positive (FP): 3
- c. False Negative (FN): 11
- d. True Positive (TP): 37

ROC AUC Score sebesar 0.9700 menunjukkan bahwa model memiliki kemampuan prediksi dengan tingkat keakuratan yang sangat tinggi, seperti tercermin dari nilai ROC AUC yang mendekati 1.

### Optimasi Model KNN

Proses optimasi hyperparameter untuk model K-Nearest Neighbors (KNN) dilakukan menggunakan Algoritma Genetika (GA). Parameter GA yang digunakan adalah sebagai berikut:

- a. Ukuran populasi: 20
- b. Jumlah generasi: 10
- c. Probabilitas crossover: 0.8
- d. Probabilitas mutasi: 0.1

Setiap individu dalam populasi direpresentasikan sebagai kromosom yang terdiri dari tiga hyperparameter utama:

- a. Jumlah tetangga terdekat ( $n\_neighbors$ ): range 1–20
- b. Fungsi bobot ( $weights$ ): {uniform, distance}
- c. Metrik jarak ( $metric$ ): {euclidean, manhattan, minkowski}

### Hasil Evolusi Algoritma Genetika KNN

Setelah 10 generasi, nilai fitness terbaik yang dicapai adalah 0.9485. Konfigurasi hyperparameter terbaik untuk model KNN adalah sebagai berikut:

- a. Jumlah tetangga terdekat ( $n\_neighbors$ ): 4
- b. Fungsi bobot ( $weights$ ): uniform
- c. Metrik jarak ( $metric$ ): euclidean

### Hasil Performa Model KNN Setelah Optimasi

Model KNN dengan hyperparameter hasil optimasi menunjukkan performa yang sangat baik. Evaluasi dilakukan pada data uji menggunakan berbagai metrik performa, hasilnya dapat dilihat pada gambar 15.

```
Confusion Matrix:
[[50  2]
 [ 3 45]]
ROC AUC Score: 0.9677
```

**Gambar 15.** Confusion Matrix Performa KNN

Dimana:

- True Negative (TN): 49
- False Positive (FP): 2
- False Negative (FN): 3
- True Positive (TP): 45

ROC AUC Score sebesar 0.9677 menunjukkan bahwa model KNN memiliki kemampuan klasifikasi yang sangat baik dalam membedakan antara kelas positif dan negatif, sebagaimana tercermin dari nilai ROC AUC yang mendekati 1.

### Perbandingan Hasil Metrik Evaluasi (ANN vs KNN)

Hasil evaluasi performa model ANN dan KNN setelah optimasi hyperparameter menggunakan algoritma genetika menunjukkan peningkatan signifikan dibandingkan konfigurasi default. Berikut adalah hasil perbandingan berdasarkan empat metrik utama: Accuracy, Precision, Recall, dan F1-Score, serta ROC AUC.

### Hasil Evaluasi ANN

Model ANN yang telah dioptimasi dengan algoritma genetika mencapai hasil sebagai berikut:

- a. Accuracy: 94.85%
- b. Precision: 93.00%
- c. Recall: 97.00%
- d. F1-Score: 95.00%
- e. ROC AUC: 97.00%

Model ANN menunjukkan kemampuan yang sangat baik dalam mendeteksi kelas positif dengan recall tinggi (97.00%), yang

mengindikasikan sensitivitas terhadap kasus-kasus positif penyakit jantung.

Hasil Evaluasi KNN

Model KNN setelah optimasi memberikan hasil sebagai berikut:

- a. Accuracy: 93.30%
- b. Precision: 92.00%
- c. Recall: 95.00%
- d. F1-Score: 93.50%
- e. ROC AUC: 96.77%

Model KNN juga menunjukkan performa yang baik, dengan precision yang cukup tinggi (92.00%), sehingga lebih handal dalam mengurangi jumlah kesalahan positif dibandingkan ANN.

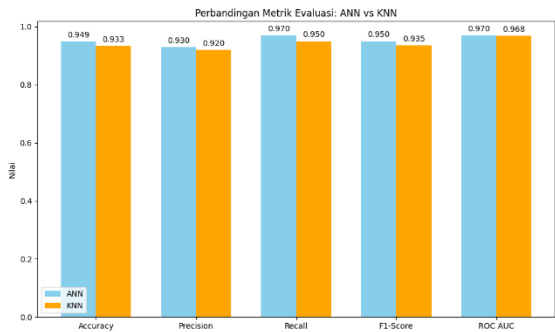
Perbandingan Metrik

Hasil evaluasi kedua model ANN dan KNN dirangkum pada Tabel 5 berikut:

Tabel 5. Perbandingan Metrik Evaluasi ANN dan KNN Setelah Optimasi

Metrik	ANN	KNN
Accuracy	94.85%	93.30%
Precision	93.00%	92.00%
Recall	97.00%	95.00%
Recall	95.00%	93.50%
ROC AUC	97.00%	96.77%

Perbandingan hasil metrik evaluasi ANN dan KNN setelah optimasi divisualisasikan pada Gambar 16.



Gambar 16. Perbandingan Metrik Evaluasi ANN dan KNN Setelah Optimasi

KESIMPULAN

Pengembangan optimasi algoritma genetika pada perbandingan model ANN dan KNN untuk klasifikasi penyakit jantung telah menghasilkan

peningkatan performa yang signifikan pada kedua model. Model ANN yang dioptimasi mencapai akurasi 94,85%, presisi 93,00%, recall 97,00%, F1-Score 95,00%, dan ROC AUC 97,00%, dengan konfigurasi optimal 50 neuron hidden layer, learning rate 0,0062, batch size 85, dan 87 epoch, menunjukkan keunggulan dalam sensitivitas yang menjadikannya ideal untuk skrining awal penyakit jantung. Sementara model KNN teroptimasi menunjukkan performa yang stabil dengan akurasi 93,30%, presisi 92,00%, recall 95,00%, F1-Score 93,50%, dan ROC AUC 96,77%, dengan parameter optimal 4 tetangga terdekat, bobot uniform, dan metrik jarak Euclidean, memberikan keseimbangan yang lebih baik antara presisi dan recall yang membuatnya cocok untuk dataset kecil. Efektivitas optimasi GA terbukti melalui peningkatan signifikan pada kedua model, dengan masing-masing model menunjukkan keunggulan untuk skenario implementasi yang berbeda dalam konteks diagnosis penyakit jantung, memberikan kontribusi penting dalam pengembangan sistem prediksi penyakit jantung yang lebih akurat.

DAFTAR PUSTAKA

Awalullaili, F. O., Ispriyanti, D., & Widiharih, T. (2023). Klasifikasi Penyakit Hipertensi Menggunakan Metode Svm Grid Search Dan Svm Genetic Algorithm (Ga). *Jurnal Gaussian*, 11(4), 488–498. <https://doi.org/10.14710/j.gauss.11.4.488-498>

Diana Dewi, D., Qisthi, N., Lestari, S. S. S., & Putri, Z. H. S. (2023). Perbandingan Metode Neural Network Dan Support Vector Machine Dalam Klasifikasi Diagnosa Penyakit Diabetes. *Cerdika: Jurnal Ilmiah Indonesia*, 3(09), 828–839. <https://doi.org/10.59141/cerdika.v3i09.662>

Febianto, A. K., & Sugianto, C. A. (2024). Optimalisasi Algoritma Klasifikasi Ensemble Menggunakan Algoritma Genetika Untuk Prediksi Resiko Diabetes. *Brahmana: Jurnal Penerapan Kecerdasan Buatan*, 5(2), 205–213.

Kusuma, J., Rubianto, Rosnelly, R., Hartono, & Hayadi, B. H. (2023). Klasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Algoritma Support Vector Machine, K-Nearest Neighbors dan Multilayer Perceptron. *Journal of Applied Computer*

- Science and Technology*, 4(1), 1–6.  
<https://doi.org/10.52158/jacost.v4i1.484>
- Pradana, D., Luthfi Alghifari, M., Farhan Juna, M., & Palaguna, D. (2022a). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), 55–60.  
<https://doi.org/10.56705/ijodas.v3i2.35>
- Pradana, D., Luthfi Alghifari, M., Farhan Juna, M., & Palaguna, D. (2022b). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), 55–60.  
<https://doi.org/10.56705/ijodas.v3i2.35>
- Pratama, Y., Prayitno, A., Azrian, D., Aini, N., Rizki, Y., & Rasywir, E. (2022). Klasifikasi Penyakit Gagal Jantung Menggunakan Algoritma K-Nearest Neighbor. *Bulletin of Computer Science Research*, 3(1), 52–56.  
<https://doi.org/10.47065/bulletincsr.v3i1.203>
- Siagian, Y., Hutahaean, J., Zikra Syah, A., Efendi Hutagalung, J., & Karim, A. (2024). Implementasi Metode K-Nearest Neighbours (KNN) Untuk Klasifikasi Penyakit Diabetes. *Jurnal Informatika Dan Teknologi Informasi*, 2(3), 253–262.  
<https://doi.org/10.56854/jt.v2i3.331>
- Yulianto, L. D., Hermaliani, E. H., & Kurniawati, L. (2023). Penerapan Machine Learning Dalam Analisis Stadium Penyakit Hati Untuk Proses Diagnosis dan Perawatan. *Media Online*, 3(4), 170–180.