

## **PENINGKATAN AKURASI KLASIFIKASI ALGORITMA NAÏVE BAYES PADA PASIEN PENYAKIT JANTUNG MENGGUNAKAN METODE PARTICLE SWARM OPTIMIZATION (PSO) DI IHC RUMAH SAKIT PERTAMINA**

**Achmad Febrian Dwi Cahyono<sup>1)</sup>, Alexius Endy Budianto<sup>2)</sup>, Moh. Ahsan<sup>3)</sup>**

*Universitas PGRI Kanjuruhan Malang<sup>1,2,3)</sup>*

*[achmad.febrian1994@gmail.com](mailto:achmad.febrian1994@gmail.com)*

### ***Abstrak***

*Metode Particle Swarm Optimization (PSO) digunakan dalam penelitian ini untuk meningkatkan akurasi penggunaan algoritma Naïve Bayes dalam mengklasifikasikan angka harapan hidup pasien penyakit jantung. Data penyakit jantung untuk 300 sampel dari Kaggle digunakan dalam penelitian ini. Studi tersebut membandingkan akurasi klasifikasi sebelum dan sesudah penerapan PSO dengan algoritma Naïve Bayes. Dalam tiga kasus yang diperiksa, penerapan PSO terbukti berhasil meningkatkan akurasi klasifikasi. Dari 0,78 menjadi 0,83, akurasi meningkat sebesar 5% pada skenario pertama. Dari 0,81 menjadi 0,86, akurasinya juga meningkat sebesar 5% pada skenario kedua. Sementara itu, dari 0,8 menjadi 0,83 pada contoh ketiga, akurasi meningkat sebesar 3%. Secara keseluruhan, PSO mampu meningkatkan kinerja algoritma Naïve Bayes, menghasilkan peningkatan akurasi sebesar 5%, 5%, dan 3% pada setiap kasus. Dalam konteks data penyakit jantung, temuan ini menunjukkan bahwa PSO dapat digunakan sebagai metode yang efektif untuk meningkatkan akurasi model prediksi.*

**Kata Kunci :** Naïve Bayes; PSO; Meningkatkan Akurasi

### ***Abstract***

*The Particle Swarm Optimization (PSO) method is used in this study to improve the accuracy of using the Naïve Bayes algorithm in classifying the life expectancy of heart disease patients. Heart disease data for 300 samples from Kaggle are used in this study. The study compares the classification accuracy before and after applying PSO with the Naïve Bayes algorithm. In the three cases examined, the application of PSO proved successful in improving the classification accuracy. From 0.78 to 0.83, the accuracy increased by 5% in the first scenario. From 0.81 to 0.86, the accuracy also increased by 5% in the second scenario. Meanwhile, from 0.8 to 0.83 in the third example, the accuracy increased by 3%. Overall, PSO was able to improve the performance of the Naïve Bayes algorithm, resulting in an increase in accuracy of 5%, 5%, and 3% in each case. In the context of heart disease data, these findings suggest that PSO can be used as an effective method to improve the accuracy of predictive models.*

**Keywords :** Naïve Bayes; PSO; Improving Accuracy

## **1. PENDAHULUAN**

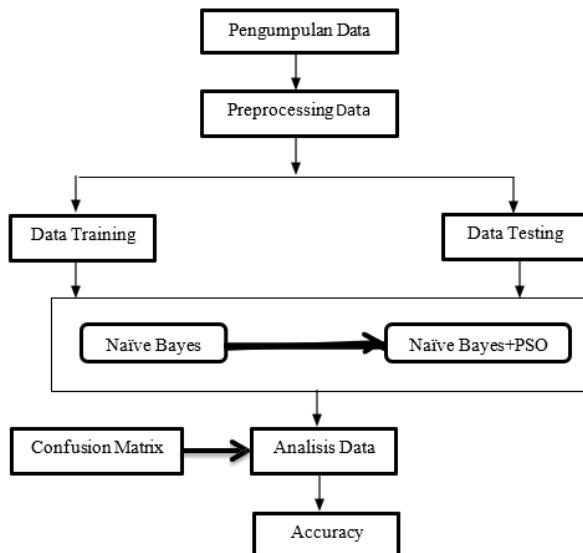
Teknologi Informasi semakin berkembang dari masa ke masa dan mengakibatkan tingkat kebutuhan akan informasi semakin meningkat. Informasi sangat dibutuhkan dalam berbagai bidang salah satunya bidang kesehatan. Keakuratan informasi dalam sebuah keputusan sangat dibutuhkan untuk mendapatkan hasil yang terbaik. Banyaknya informasi dalam dunia kesehatan dapat dijadikan sebuah data untuk mendapatkan hasil yang diinginkan dengan data mining. Data tersebut dapat dijadikan untuk mengambil sebuah keputusan. Dalam bidang kesehatan data mining dapat digunakan untuk mengklasifikasi suatu penyakit, salah satunya adalah penyakit jantung (Wulandari, 2013).

Penyakit jantung atau kardiovaskuler biasanya terjadi karena kerusakan sel otot-otot jantung

dalam memompa aliran darah keseluruh tubuh, yang disebabkan kekurangan oksigen yang dibawa darah ke pembuluh darah di jantung atau juga karena terjadi kejang pada otot jantung yang menyebabkan kegagalan organ jantung dalam memompa darah, sehingga menyebabkan kondisi jantung tidak dapat melaksanakan fungsinya dengan baik (Wahyudi dan Hartati, 2017). Naïve Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Tujuannya untuk memperoleh akurasi yang baik dalam mengklasifikasi harapan hidup pasien penderita penyakit jantung (Herdiana et al., n.d.). Particle Swarm Optimization (PSO) merupakan metode optimasi yang dapat digunakan untuk menentukan parameter-parameter proses yang menghasilkan nilai respon yang optimum. Pemanfaatan metode optimasi Particle swarm optimization pada Naïve Bayes merupakan salah satu teknik yang digunakan untuk menyelesaikan permasalahan klasifikasi dengan mengevaluasi akurasi yang dihasilkan pada Naïve Bayes (Angga S, Indra D. S dan Yuli Dharta 2018). Naïve Bayes sudah pernah digunakan pada penelitian sebelumnya oleh Taghfirul Azhima Yoga Siswa dan Prihandoko(2018) untuk mencari kinerja terbaik dalam mendeteksi kanker payudara. Penelitian ini dilakukan dengan dua kali percobaan. Percobaan pertama menggunakan algoritma Naïve Bayes(PSO) dengan peningkatan akurasi dari 97.37% menjadi 97.81%. Kemudian percobaan kedua untuk nilai akurasi algoritma K-Nearest Neighbor dari 95.45% mengalami peningkatan menjadi 96.63%. Selanjutnya dilakukan oleh Rusda Wajhillah(2014) mengklasifikasi penyakit jantung dengan menggunakan algoritma C4.5 Berbasis Particle Swarm Optimization yang dilakukan dengan dua kali percobaan. Percobaan pertama menggunakan algoritma C4.5 memperoleh akurasi sebesar 81,25%. Sedangkan percobaan kedua untuk nilai akurasi algoritma klasifikasi C4.5 berbasis PSO sebesar 93,75%.

## 2. METODE

Dalam penelitian ini, menggunakan data penyakit jantung yang didapatkan dari kaggle dan diolah menggunakan algoritma Naïve Bayes dan Particle Swarm Optimization, untuk meningkatkan nilai akurasi sehingga menghasilkan nilai akurasi yang lebih baik pada klasifikasi harapan hidup pasien penyakit jantung. Adapun tahapan dari penelitian yang dilakukan adalah sebagai berikut.



**Gambar .1 Rancangan Penelitian**

### 2.1.Populasi dan Sample

Pengumpulan data yang digunakan dalam penelitian ini menggunakan data penyakit jantung

dengan jumlah 300 data. Atribut yang digunakan dalam penelitian sebanyak 13 atribut. Berikut atribut yang digunakan dalam penelitian ini dapat dilihat pada tabel 3.1 adalah sebagai berikut:

**Tabel 1 Daftar Atribut dan Keterangannya**

No.	Atribut	Keterangan
1.	Age	Umur pasien
2.	Anemia	Penurunan sel darah merah atau hemoglobin
3.	Creatinine Phosphokinase	Tingkat enzim CPK dalam darah
4.	Diabetes	Memiliki Diabetes
5.	Ejection Fraction	Persentase darah yang meninggalkan jantung pada setiap kontraksi
6.	High Blood Pressure	Memiliki hipertensi
7.	Platelets	Trombosit dalam darah
8.	Serum Creatinine	Tingkat kreatinin dalam darah
9.	Serum Sodium	Tingkat sodium dalam darah
10.	Sex	Jenis kelamin
11.	Smoking	Merokok
12.	Time	Periode <i>follow-up</i> pasien
13.	Death Event	Jika pasien meninggal selama masa <i>follow-up</i>

## 2.2. Definisi Operasional / *Preprocessing* Data

Dalam proses ini dilakukan dengan pembersihan data dengan cara menghapus data yang kosong atau tidak terisi pada atribut yang sudah ditentukan. Hal ini dilakukan dengan tujuan untuk menyesuaikan data dengan kebutuhan sehingga terhindari dari data yang tidak konsisten. Ada beberapa tahapan dalam prosessing data

- Mengubah data kategorik menjadi data numerik

Mengonversi data kategorikal menjadi bentuk numerik dalam penerapan Particle Swarm Optimization (PSO) untuk meningkatkan Keakuratan klasifikasi harapan hidup pasien penyakit jantung menggunakan algoritma Naïve Bayes bertujuan pada beberapa hal penting. Hal ini termasuk membuat data kompatibel dengan algoritma, menghindari kesalahan dalam perhitungan, memungkinkan perbandingan dan analisis yang lebih baik, memfasilitasi operasi PSO yang efisien, dan pada akhirnya, meningkatkan akurasi klasifikasi penyakit jantung. Salah satu cara untuk melakukannya adalah dengan teknik pengkodean seperti One-Hot Encoding atau Label Encoding, bergantung pada jenis data kategorikal yang ada dan tujuan analisis yang ingin dicapai. Setelah data dikonversi, langkah selanjutnya adalah melatih model Naïve Bayes dan menerapkan PSO untuk optimasi parameter sehingga model dapat memberikan klasifikasi yang lebih akurat

- Normalisasi data

Metode yang digunakan dalam normalisasi data yaitu Min-Max. Min-Max merupakan salah satu teknik normalisasi yang dipakai dalam analisis data dan pemrosesan data untuk mengubah nilai-nilai dalam dataset menjadi rentang yang lebih terbatas, biasanya antara 0 dan 1.

Normalisasi data pada penerapan Particle Swarm Optimization (PSO) untuk meningkatkan akurasi klasifikasi harapan hidup pasien penyakit jantung menggunakan algoritma Naïve Bayes bertujuan pada beberapa hal penting:

- Perbedaan Skala Penanganan: Data yang digunakan dalam prediksi penyakit jantung seringkali memiliki variasi skala yang berbeda, seperti usia pasien, tekanan darah, dan kadar kolesterol. Normalisasi membantu menyamakan skala atribut-atribut ini, menghindari pengaruh yang tidak proporsional akibat variasi skala.
- Meningkatkan Efisiensi PSO: PSO adalah algoritma untuk menemukan solusi terbaik dalam ruang pencarian. Normalisasi membantu mencegah atribut tertentu mendominasi proses optimasi, memungkinkan PSO mencapai konvergensi lebih cepat

dan solusi keseluruhan yang lebih baik.

3. Meningkatkan Akurasi Algoritma Naïve Bayes: Algoritma Naïve Bayes mengasumsikan independensi bersyarat dari atribut. Normalisasi membantu mempertahankan asumsi-asumsi ini, meningkatkan akurasi klasifikasi dengan menghindari kemungkinan korelasi atau ketergantungan antar atribut.
  4. Mencegah Bias: Normalisasi menghindari bias yang dapat timbul jika satu atribut memiliki variasi yang jauh lebih besar dibandingkan atribut lainnya. Hal ini memastikan bahwa model Naïve Bayes tidak memberikan bobot yang tidak proporsional pada atribut berskala lebih besar.
  5. Memfasilitasi Interpretasi Data: Data yang telah dinormalisasi lebih mudah dipahami dan diinterpretasikan. Hal ini membantu para peneliti dan profesional medis untuk lebih memahami faktor-faktor yang mempengaruhi klasifikasi penyakit jantung dan membuat rekomendasi yang lebih akurat
- c) Bining

Metode yang digunakan yaitu Quantile. Quantile adalah nilai yang membagi data ke dalam fraksi yang sama dengan menggunakan pada quantile.

Tujuan penggunaan teknik binning dalam penerapan Particle Swarm Optimization (PSO) untuk meningkatkan akurasi klasifikasi harapan hidup pasien penyakit jantung dengan algoritma Naïve Bayes adalah sebagai berikut:

1. Mengurangi Variabilitas: Binning bertujuan untuk mengurangi variasi data dengan mengganti nilai kontinu dengan interval atau kategori tertentu, sehingga mengurangi fluktuasi kecil yang mungkin tidak relevan untuk klasifikasi penyakit jantung.
2. Meningkatkan Kekokohan Model: Dengan mengelompokkan data ke dalam interval atau kategori, kita dapat membuat model klasifikasi lebih kuat terhadap data yang tidak biasa atau outlier. Hal ini akan membuat model lebih stabil dan dapat diandalkan.
3. Optimasi Proses: Data yang diikat dapat mengurangi jumlah informasi yang harus diproses oleh algoritma Naïve Bayes, yang pada gilirannya dapat meningkatkan efisiensi dalam pelatihan dan klasifikasi model.
4. Kemudahan Interpretasi: Binning dapat membuat hasil model lebih mudah dipahami oleh mereka yang tidak memiliki latar belakang statistik atau pemahaman mendalam tentang komputasi. Interval atau kategori yang digunakan dalam binning lebih mudah diinterpretasikan dibandingkan dengan nilai kontinu.
5. Peningkatan Akurasi Model: Dengan membersihkan data dari noise dan meningkatkan kualitasnya, penggunaan binning berpotensi meningkatkan akurasi model Naïve Bayes dalam mengklasifikasi harapan hidup pasien penyakit jantung. Dari 13 atribut dalam penerapan Particle Swarm Optimization (PSO) untuk meningkatkan akurasi klasifikasi harapan hidup pasien penyakit jantung dengan algoritma Naïve Bayes hanya tujuh atribut yang akan di binning di antaranya Age, Creatinine Phosphokinase, Ejection Fraction, Platelets, Serum Creatinine, Serum Sodium, Time.

### **2.3.Instrumen Penelitian (Data Training dan Data Testing)**

Data training digunakan untuk pembentukan model data mining sedangkan data testing digunakan untuk mengetahui performa model pada proses testing. Sehingga sebelum data dianalisis maka terlebih dahulu menentukan data training dan data testing. Penentuan data training dan data testing dalam penerapan Particle Swarm Optimization (PSO) untuk meningkatkan akurasi klasifikasi harapan hidup pasien penyakit jantung dengan algoritma Naïve Bayes dilakukan secara random sebanyak tiga kali percobaan dengan persentase yang berbedah yaitu 70% dan 30%, 80% dan 20%, 90% dan 10%.

### **2.4.Pengumpulan Data (Analisis Data)**

Pada tahap ini adalah menganalisis data. Penerapan Particle Swarm Optimization(PSO). Dalam

Meningkatkan Akurasi Klasifikasi harapan hidup pasien Penyakit Jantung Dengan Algoritma Naïve Bayes, dilakukan proses perhitungan, untuk mendapatkan hasil akurasi yang lebih baik dengan melakukan pengujian menggunakan Confusion Matrix. Confusion Matrix merupakan salah satu jenis pengujian untuk mengevaluasi model yang dilakukan terbilang efektif, dengan tujuan untuk memperoleh hasil akurasi yang lebih baik.

### 3. HASIL DAN PEMBAHASAN

Dalam penelitian ini data yang digunakan adalah data pasien penyakit jantung yang diperoleh dari kaggle. Data yang diperoleh untuk penelitian sebanyak 300 data. Atribut yang digunakan dalam penelitian sebanyak 13 atribut diantaranya yaitu age, Anemia, Creatinine Phosphokinase, Diabetes , Ejection Fraction, High Blood Pressure, Platelets, Serum Creatinine, Serum Sodium, Sex, Smoking, Time, Death Event . Langkah-langkah pengumpulan data yang pertama yaitu mengumpulkan data mentah yang siap diolah menggunakan Exel dan diubah kedalam bentuk data CSV, pada proses ini merupakan langkah awal dalam memulai proses pada python. Data pada tabel 4.1 dibawah merupakan potongan data dari kaggle, untuk data yang lengkapnya akan ditampilkan pada lampiran.

**Tabel 2 Sebagian data yang diolah dari kaggle**

No	Age	Anemia	Creatinine Phosphokinase	Diabetes	Ejection Fraction	High Blood Pressure	Platelets	Serum Creatinine	Serum Sodium	Sex	Smoking	Time	Death Event
1	75	0	582	0	20	1	265000	1.9	130	1	0	4	1
2	55	0	7861	0	38	0	263385.03	1.1	136	1	0	6	1
3	65	0	146	0	20	0	162000	1.3	129	1	1	7	1
4	50	1	111	0	20	0	210000	1.9	137	1	0	7	1
5	65	1	160	1	20	0	327000	2.7	116	0	0	8	1
6	90	1	47	0	40	1	204000	2.1	132	1	1	8	1
7	75	1	246	0	15	0	127000	1.2	137	1	0	10	1
8	60	1	315	1	60	0	454000	1.1	131	1	1	10	1
9	65	0	157	0	65	0	263358.03	1.5	138	0	0	10	1
10	80	1	123	0	35	1	388000	9.4	133	1	1	10	1
11	75	1	81	0	38	1	368000	4	131	1	1	10	1
12	62	0	231	0	25	1	253000	0.9	140	1	1	10	1
13	45	1	981	0	30	0	136000	1.1	137	1	0	11	1
14	50	1	168	0	38	1	276000	1.1	137	1	0	11	1
15	49	1	80	0	30	1	427000	1	138	0	0	12	0
16	82	1	379	0	50	0	47000	1.3	136	1	0	13	1
17	87	1	149	0	38	0	262000	0.9	140	1	0	14	1
18	45	0	582	0	14	0	166000	0.8	127	1	0	14	1
19	70	1	125	0	25	1	237000	1	140	0	0	15	1
20	48	1	582	1	55	0	87000	1.9	121	0	0	15	1
.....													
300													

#### 3.1. Preprocessing Data

Preprocessing merupakan suatu tahapan yang sangat penting dalam proses data mining untuk menghindari permasalahan yang dapat mengganggu hasil dari proses data mining. Secara umum preprocessing data terdiri dari beberapa proses yaitu *cleaning* digunakan untuk mendekripsi, memperbaiki atau menghapus serta memeriksa data yang tidak konsisten atau tidak akurat, *integration* digunakan untuk menggabungkan data dari beberapa sumber ke dalam penyimpanan data yang sesuai, *transformation* digunakan untuk menormalisasi serta pengumpulan data menjadi sama, *reduction* digunakan menguraikan data ke dalam bentuk ukuran yang lebih kecil tetapi menghasilkan hasil analitis yang sama. Tahap pengumpulan data yang dilakukan pada penelitian ini adalah sebagai berikut :

- 1) Mengimport data yang sudah dikumpulkan melalui excel seperti pada tabel 4.1 di atas

ke *google colab python* dengan model csv. Tujuan mengubah data dari excel ke CSV agar data bisa terbaca oleh pemrograman python, data lengkapnya akan di tampilkan pada lampiran.

**Tabel 3. Data yang sudah di import ke python melalui CSV**

Index	No	Age	Anemia	Creatinine Phosphokinase	Diabetes	Ejection Fraction	High Blood Pressure	Platelets	Serum Creatinine	Serum Sodium	Sex	Smoking	Time	Death Event
0	1	75	0	582	0	20	1	265000	1.9	130	1	0	4	1
1	2	55	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	3	65	0	146	0	20	0	162000	1.3	129	1	1	7	1
3	4	50	1	111	0	20	0	210000	1.9	137	1	0	7	1
4	5	65	1	160	1	20	0	327000	2.7	116	0	0	8	1
<hr/>														
294	295	62	0	61	1	38	1	155000	1.1	143	1	1	270	0
295	296	55	0	1820	0	38	0	270000	1.2	139	0	0	271	0
296	297	45	0	2060	1	60	0	742000	0.8	138	0	0	278	0

### 3.2. Data Training Dan Data Testing

Untuk menentukan data yang akan dianalisis pada Naïve Bayes terlebih dahulu tentukan data training dan data testing. Data training digunakan untuk pembentukan model data mining sedangkan data testing digunakan untuk mengetahui performa model pada proses testing. Penentuan data training dan data testing pada data pasien penyakit jantung dilakukan secara random sebanyak tiga kali percobaan dengan presentase yang berbeda yaitu 70% dan 30%, 80% dan 20%, 90% dan 10% seperti pada tabel 4.3 dibawah ini. Untuk skrip pembagian data training dan testing pada tabel 4.3 dibawah ini akan dilengkapi pada lampiran.

**Tabel 4. Pembagian data training dan testing 70% dan 30%, 80% dan 20%, 90% dan 10%**

Proporsi 70% Data Training dan 30% Data Testing	Proporsi 80% Data Training dan 20% Data Testing	Proporsi 90% Data Training dan 10% Data Testing
Jumlah data training 70% : 210	Jumlah data training 80% : 240	Jumlah data training 90 : 270
Jumlah data testing 30% : 90	Jumlah data testing 20% : 60	Jumlah data testing 10% : 30

### 3.3. Klasifikasi Perhitungan Berdasarkan Proporsi data training dan testing

**Tabel 5. Klasifikasi Perhitungan**

Proporsi 70% : 30%	Proporsi 80% : 20%	Proporsi 90% : 10%
Proporsi 70% Data Training dan 30% Data Testing	Proporsi 80% Data Training dan 20% Data Testing	Proporsi 90% Data Training dan 10% Data Testing
Jumlah data training 70% : 210	Jumlah data training 80% : 240	Jumlah data training 90 : 270
Jumlah data testing 30% : 90	Jumlah data testing 20% : 60	Jumlah data testing 10% : 30

### 3.3. Menghitung Akurasi Menggunakan Naïve Bayes

Dalam menghitung akurasi menggunakan algoritma naïve bayes hal pertama yang perlu kita siapkan yaitu data kemudian mengimport modul

*from sklearn.naive\_bayes import GaussianNB*

```
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

- GaussianNB adalah kelas dari model Naive Bayes yang menggunakan distribusi Gaussian. Ini adalah algoritma pembelajaran mesin yang sering digunakan untuk klasifikasi.
- accuracy\_score adalah fungsi dari sklearn.metrics yang digunakan untuk menghitung akurasi prediksi dari model klasifikasi.

Kemudian, terdapat sebuah fungsi bernama ‘evaluate\_model’ yang menerima empat parameter: ‘X\_train, y\_train’, X\_test, dan ‘y\_test’. Fungsi ini bertujuan untuk mengukur performa model secara menyeluruh dan menghitung akurasi prediksi menggunakan model Naive Bayes.

1. Inisialisasi model Naive Bayes menggunakan GaussianNB().  
 $clf = GaussianNB()$
2. Melatih model menggunakan data latih (X\_train dan y\_train) dengan menggunakan metode fit().  
 $clf.fit(X\_train, y\_train)$
3. Menguji model yang telah dilatih pada data uji (X\_test) dengan menggunakan metode predict() untuk memprediksi kelas dari data uji.  
 $y\_pred = clf.predict(X\_test)$
4. Menghitung akurasi prediksi menggunakan label sebenarnya (y\_test) dan prediksi yang dihasilkan (y\_pred) menggunakan fungsi accuracy\_score.  
 $accuracy = accuracy\_score(y\_test, y\_pred)$
5. Mengembalikan nilai akurasi.

Setelah itu, percobaan dilakukan dengan membagi data menjadi tiga proporsi yang berbeda, 70% untuk data latih dan 30% untuk data uji, 80% untuk data latih dan 20% untuk data uji, serta 90% untuk data latih dan 10% untuk data uji. Setelah melalui proses tersebut, akurasi dari setiap percobaan kemudian dicetak, dan hasilnya seperti pada tabel berikut ini, untuk skripnya akan di lengkap pada lampiran.

**Tabel 6. Akurasi dari tiga proporsi data menggunakan naïve bayes**

Akurasi dengan 70% Data Training dan 30% Data Testing	0.78
Akurasi dengan 80% Data Training dan 20% Data Testing	0.81
Akurasi dengan 90% Data Training dan 10% Data Testing	0.8

6. classification\_report

Fungsi ini digunakan untuk menghasilkan metrik klasifikasi yang lebih rinci, seperti presisi (precision), recall, f1-score, dan support, untuk setiap kelas dalam data.

7. Output Laporan Klasifikasi

Laporan klasifikasi ini memberikan informasi tentang performa model pada setiap kelas dan metrik agregat.

- Presisi mengukur seberapa baik model memprediksi instance positif.
- Precision untuk kelas 0 dan 1 ditampilkan.

8. Recall (Sensitivitas atau True Positive Rate):

- Recall mengukur seberapa baik model dapat mendeteksi instance positif.
- Recall untuk kelas 0 dan 1 ditampilkan.

9. F1-Score:

- F1-score adalah rata-rata harmonik dari precision dan recall.
- F1-score untuk kelas 0 dan 1 ditampilkan.

10. Support:

- Jumlah instance dalam setiap kelas.

11. Accuracy (Akurasi):

- Akurasi adalah rasio prediksi benar terhadap total instance.

## 12. Macro Avg dan Weighted Avg:

- Macro avg adalah rata-rata dari metrik evaluasi untuk setiap kelas, tanpa memperhatikan proporsi instance dalam setiap kelas.
- Weighted avg adalah rata-rata yang diberi bobot berdasarkan jumlah instance dalam setiap kelas.

Hasil dari laporan klasifikasi memberikan pemahaman yang lebih mendalam tentang seberapa baik model dapat memprediksi setiap kelas, serta keseimbangan antara precision dan recall. Semua metrik ini membantu dalam mengevaluasi performa model klasifikasi secara holistik.

**3.4. Confusion Matrix**

1. Menghitung Confusion Matrix: Pertama-tama, confusion matrix dihitung menggunakan `confusion_matrix(y_test, y_pred)`. Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi. Ini menunjukkan jumlah prediksi yang benar dan salah yang dibuat oleh model dalam setiap kelas target.

`cm = confusion_matrix(y_test, y_pred)`

2. Proporsi Pembagian Data: Data dibagi menjadi tiga proporsi yang berbeda: 70% data latih dan 30% data uji, 80% data latih dan 20% data uji, serta 90% data latih dan 10% data uji. Ini dilakukan untuk menguji performa model dengan variasi proporsi data latih dan data uji.
3. Menghitung Akurasi dan Confusion Matrix: Untuk setiap proporsi pembagian data, fungsi `evaluate_model_and_confusion_matrix()` digunakan untuk menghitung akurasi dan confusion matrix. Akurasi diukur menggunakan fungsi `accuracy_score()` yang mengukur persentase prediksi yang benar dari model. Confusion matrix kemudian dicetak untuk menunjukkan distribusi prediksi yang benar dan salah pada setiap kelas.

`plt.subplot(1, 3, 1)`

```
sns.heatmap(cm_70, annot=True,
fmt="d", cmap="Blues", cbar=False, annot_kws={"size": 10})
plt.title("Confusion Matrix (70:30)")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
```

**Tabel 7. Hasil Akurasi dan Confusion Matrix**

Hasil Akurasi	Confusion Matrix
Akurasi dengan 70% Data Training dan 30% Data Testing: 0.78	<code>[[44 9] [10 27]]</code>
Akurasi dengan 80% Data Training dan 20% Data Testing: 0.81	<code>[[35 5] [6 14]]</code>
Akurasi dengan 90% Data Training dan 10% Data Testing: 0.8	<code>[[19 3] [ 3 5]]</code>

4. Plotting Confusion Matrix: Setelah mendapatkan confusion matrix untuk setiap proporsi pembagian data, keseluruhan confusion matrix diplot menggunakan seaborn. Ini memberikan visualisasi yang lebih jelas tentang bagaimana model melakukan prediksi pada setiap kelas target, seperti pada gambar dibawah ini

Dalam confusion matrix tersebut dapat menafsirkan hasil prediksi dari model klasifikasi. Hasil prediksi dan model klasifikasi dari 70% data training dan 30% data testing adalah :

1. True Positif (TP): Ada 27 kasus dimana model dengan benar memprediksi pasien yang benar-benar meninggal dan diprediksi meninggal
2. True Negatif (TN): Sebanyak 44 kasus dimana model dengan benar memprediksi pasien yang benar-benar hidup dan diprediksi hidup
3. False Positif (FP): Terdapat 10 kasus dimana model salah memprediksi pasien yang

benar-benar hidup tetapi diprediksi meninggal

False Negatif (FN): Ditemukan 9 kasus di mana model salah memprediksi pasien yang benar-benar meninggal tetapi diprediksi tidak meninggal

Dari hasil yang didapatkan terlihat bahwa model memiliki kemampuan untuk mengenali kasus positif dengan baik (27 TP), namun masih ada ruang untuk memperbaiki pengenalan kasus positif yang lebih baik (FN yang lebih rendah). Model juga cukup baik dalam mengenali kasus negatif (44 TN), tetapi ada beberapa kasus negatif yang salah dikenali sebagai positif. Oleh karena itu, dalam mengevaluasi model, tidak hanya akurasi yang perlu diperhatikan, tetapi juga metrik lain seperti presisi, recall, dan F1-score, yang dihasilkan oleh fungsi `classification_report`. Berikut adalah rumus untuk metrik evaluasi

- **Precision (Presisi):**

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

$$\text{Precision}_0 = \frac{43}{43+10} = \frac{43}{53} = 0.81$$

$$\text{Precision}_1 = \frac{28}{28+9} = \frac{28}{37} = 0.75$$

- **Recall :**

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

$$\text{Recall}_0 = \frac{44}{44+9} = \frac{44}{53} = 0.83$$

$$\text{Recall}_1 = \frac{28}{28+10} = \frac{28}{38} = 0.73$$

- **Support:**

Support adalah jumlah masing-masing kelas dalam data testing.

Support untuk kelas positif (1): 36 (27 (TP) + 9(FN))

Support untuk kelas negatif (0): 54 (44 (TN) + 10(FP))

Jadi, jumlah pasien yang meninggal pada data 70% dan 30% sebanyak 36 orang.

Sedangkan pasien yang hidup sebanyak 54 orang.

- **Akurasi :**

$$\text{Accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{Total Data}}$$

$$\text{Accuracy} = \frac{44+27}{90+90} = \frac{71}{90} = 0.78$$

### 3.5.Penerapan Algoritma Naïve Bayes

Penerapan Naïve Bayes dalam penelitian digunakan untuk mengklasifikasi harapan hidup pasien penyakit jantung dengan pengujian akurasi menggunakan Confusion Matrix. Pada klasifikasi Naïve Bayes data dibagi menjadi data training dan data testing dengan masing-masing presentase berbeda yaitu jumlah data 210 data dan 90 data (70% dan 30%), 240 data dan 60 data (80% dan 20%), 270 data dan 30 data (90% dan 10%). Pada pengujian Naïve Bayes akan ditentukan akurasi yang paling tinggi dari masing-masing presentase data training dan data testing. Hasil uji coba menghasilkan akurasi yang berbeda yaitu data training 70% dan 30% data testing menghasilkan akurasi 0.78, data training 80% dan 20% data testing

menghasilkan akurasi 0.81 dan data training 90% dan 10% data testing menghasilkan akurasi 0.8. Dapat dilihat pada tabel 5.1.

**Tabel 8. Hasil Akurasi Pengujian Naïve Bayes**

Data Training Dan Data Testing	Akurasi
70% dan 30%	0.78
80% dan 20%	0.81
90% dan 10%	0.8

### 3.6.Penerapan Algoritma Particle Swarm Optimization

Penerapan Particle Swarm Optimization (PSO) digunakan untuk mengoptimasi nilai akurasi pada Naïve Bayes sehingga mendapatkan akurasi yang lebih tinggi. Pengujian akurasi pada Particle Swarm Optimization (PSO) dilakukan dengan tiga persentase data training dan data testing berbeda sesuai dengan pengujian sebelumnya. Setelah dilakukan optimasi data training 70% dan data testing 30% akurasinya menjadi 0.83, data training 80% dan data testing 20% akurasinya menjadi 0.86 dan data training 90% dan data testing 10% akurasinya menjadi 0.83. Berdasarkan hasil uji coba setelah dilakukan optimasi, algoritma Particle Swarm Optimization bisa menghasilkan akurasi yang lebih tinggi dari algoritma Naïve Bayes. Dapat dilihat pada tabel 5.2

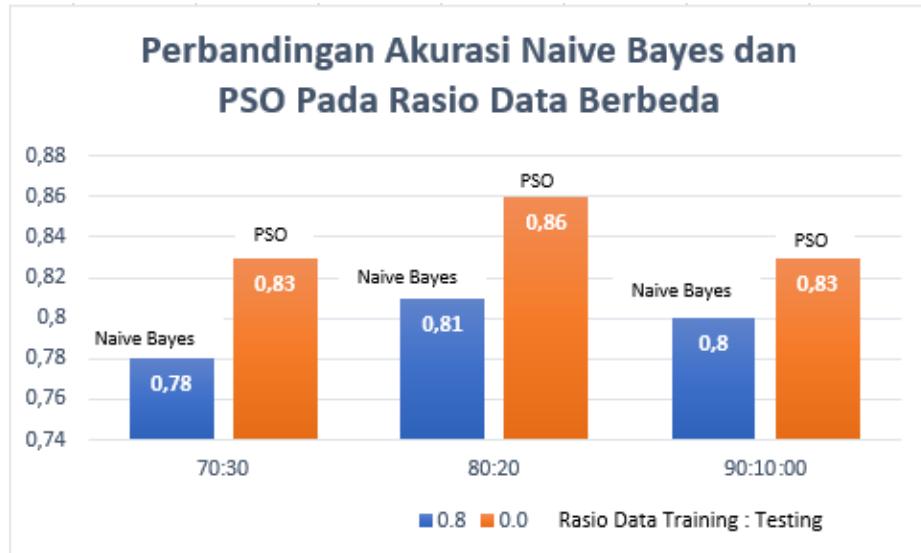
**Tabel 9. Hasil Pengujian Particle Swarm optimization (PSO)**

Data Training Dan Data Testing	Akurasi
70% dan 30%	0.83
80% dan 20%	0.86
90% dan 10%	0.83

### 3.7.Perbandingan Naïve Bayes dengan Particle Swarm Optimization

Penerapan algoritma Naïve Bayes dioptimasi dengan Particle Swarm Optimization menggunakan data pasien penyakit jantung dengan jumlah 300 data. Data dipisahkan menjadi data training dan data testing dengan persentase berbeda, dengan jumlah atribut 13.

Berdasarkan hasil optimasi akurasi yang dihasilkan semakin tinggi. Dapat dilihat pada diagram dibawah ini :

**Gambar 2. Perbandingan Akurasi Menggunakan Naïve Bayes dan PSO**

Pengujian akurasi pada penelitian ini menggunakan model *Confusion Matrix* dengan tingkat akurasi yang berbeda dari pembagian data training dan data testing. Tingkat akurasinya adalah 0.83, 0.86 dan 0.83.

#### 4. KESIMPULAN

Setelah melakukan analisis dan pengujian pada klasifikasi data pasien penyakit jantung pada Penerapan Naïve Bayes dengan Particle Swarm Optimization maka dapat diperoleh kesimpulan sebagai berikut:

1. Penerapan Algoritma Particle Swarm Optimization pada algoritma Naive Bayes, secara konsisten meningkatkan akurasi prediksi. Peningkatan akurasi yang diamati (5%, 5%, dan 3%) menunjukkan bahwa PSO efektif dalam mengoptimalkan parameter model, sehingga model mampu memprediksi data dengan lebih akurat daripada sebelumnya. Hal ini menggambarkan bahwa pendekatan optimasi seperti PSO dapat menjadi strategi yang berguna untuk meningkatkan kinerja model machine learning, terutama dalam konteks klasifikasi menggunakan Naive Bayes.
2. Peningkatan akurasi yang dihasilkan dari algoritma Particle Swarm Optimization (PSO) berhasil meningkatkan akurasi model Naive Bayes dalam tiga kasus yang diberikan. Berikut adalah ringkasan peningkatan akurasi yang dicapai:
  1. Kasus Pertama: a) Akurasi Naive Bayes sebelum PSO: 0.78, b) Akurasi setelah PSO: 0.83, c) Peningkatan akurasi: 5%
  2. Kasus Kedua: a) Akurasi Naive Bayes sebelum PSO: 0.81, b) Akurasi setelah PSO: 0.86, c) Peningkatan akurasi: 5%
  3. Kasus Ketiga: a) Akurasi Naive Bayes sebelum PSO: 0.8, b) Akurasi setelah PSO: 0.83, c) Peningkatan akurasi: 3%

#### 5. REFERENSI

- [1] Amillina, I., & Qoiriah, A. (2021). Penerapan Algoritma Naïve Bayes dalam Klasifikasi Tingkat Kepuasan Siswa terhadap Pembelajaran Daring. *Jurnal Ilmiah Teknologi Informasi dan Robotika*, 3(2).
- [2] Fathurrahman, F., & Wasil, M. (2019). Analisa Penentuan Kelayakan Kredit BUMDES di Desa Gerung Permai Kecamatan Suralaga Kabupaten Lombok Timur Menggunakan Metode C. 45 Berbasis PSO (Particle Swarm Optimization). *Infotek: Jurnal Informatika dan Teknologi*, 2(2), 156-164.
- [3] Hadikristanto W, Asti Setyaningsih. (2020). Penerapan Algoritma Naive Bayes Untuk Menentukan Klasifikasi Produk Terlaris Pada Penjualan Pulsa. *Jurnal SIGMA*, 11(2), 115-122.
- [4] Hadikristanto, W., & Fitri, V. A. (2018). Penerapan Algoritma Naive Bayes Berbasis Particle Swarm Optimization (Pso) Dalam Menangani Kasus Kanker Payudara. *Jurnal SIGMA*, 9(1), 111-120
- [5] Indrayuni, E. (2019). Klasifikasi Text Mining Review Produk Kosmetik Untuk Teks Bahasa Indonesia Menggunakan Algoritma Naive Bayes. *Jurnal Khatulistiwa Informatika*, 7(1).
- [6] Romli, I., & Putra, B. M. (2020). Evaluasi Penilaian Kinerja Dalam Klasifikasi Data Mining Dengan Metode Naive Bayes. *Jurnal Teknik Industri*, 1(2), 36-45.
- [7] Setiawan, D. A., Halilintar, R., & Wahyuniar, L. S. (2021, August). Penerapan Metode Naive Bayes Untuk Klasifikasi Penentuan Penerima Bantuan PKH. In Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi) (Vol. 5, No. 2, pp. 249-254).
- [8] Susilowati, D., Sutrisno, S., & Yunus, M. (2023). Penerapan Particle Swarm Optimization Untuk Meningkatkan Kinerja Algoritma K-Nearest Neighbor Dalam Klasifikasi Penyakit Diabetes. *J-REMI: Jurnal Rekam Medik dan Informasi Kesehatan*, 4(3), 176-184.
- [9] Taghfirul Azhima Y. S, Prihandoko, P. (2018). Penerapan Optimasi Berbasis Particle

Swarm Optimization (Pso) Algoritma Naïve Bayes Dan K-Nearest Neighbor Sebagai Perbandingan Untuk Mencari Kinerja Terbaik Dalam Mendeteksi Kanker Payudara. Metik Jurnal, 2(2), 38-45.

- [10] Wajhillah, R.(2014). Optimasi Algoritma Klasifikasi C4.5 Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Jantung. *Swabumi*, 1(1), 26- 36.