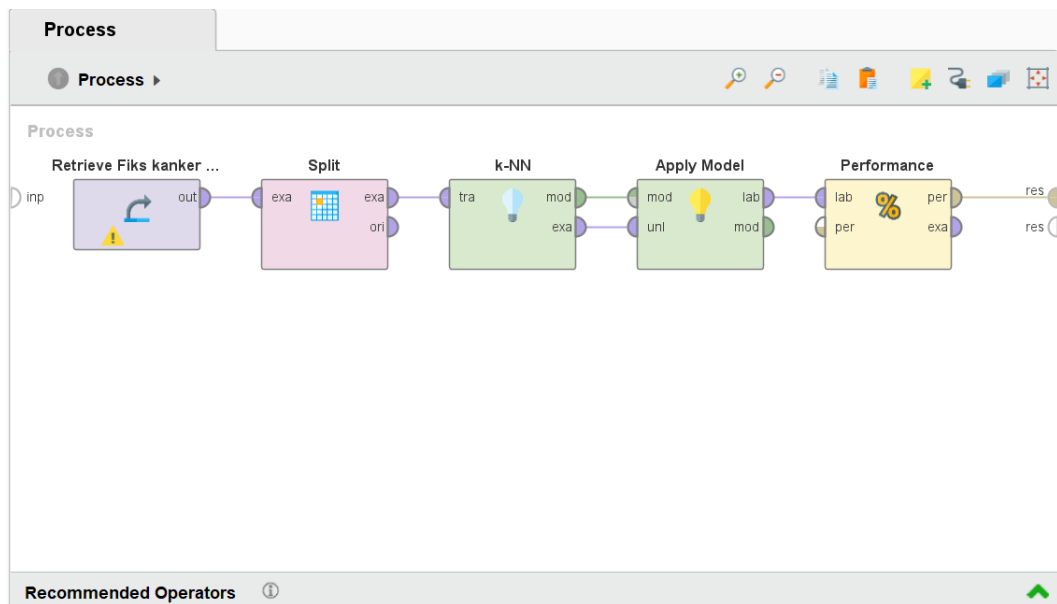


## BAB IV HASIL PEMBAHASAN

Bab ini akan membahas tentang proses, hasil, dan pembahasan penelitian. Software RapidMiner digunakan sebagai alat bantu untuk memproses data. Proses, hasil, dan pembahasan terbagi ke dalam beberapa subbab, termasuk Pengolahan dan distribusi Data serta Implementasi algoritma pada RapidMiner.

### 1.1 Hasil Penelitian

Berikut adalah penerapan algoritma KNN yang telah dioptimalkan dengan menggunakan PSO dengan menggunakan perangkat RapidMiner.



Gambar 4. 1. Proses Algoritma KNN

Pada gambar 4.1 terdapat sejumlah operator yang dipakai yang digunakan, diantaranya yaitu :

- Operator retrieve : digunakan untuk mengambil dataset kanker payudara. Dataset yang digunakan dalam analisis ini diperoleh dari repository UCI Machine Learning dengan judul "Breast Cancer Wisconsin (Diagnostic)" [1]. Dataset ini terdiri dari 569 rekaman dan mencakup 32 atribut yang melibatkan parameter-parameter seperti radius, tekstur, dan area dari inti sel, serta sejumlah

metrik yang dihitung dari gambar sel. Deskripsi variabel tersebut dalam tabel berikut ini.

Tabel 4. 1. Atribut Dataset Penyakit Diabetes

No	Atribut	Keterangan
1	<i>Id</i>	<i>ID</i>
2	<i>Diagnosis</i>	<i>Diagnosis</i>
3	<i>Radius Mean</i>	<i>Jari-jari Rata-rata</i>
4	<i>Texture Mean</i>	<i>Tekstur Rata-rata</i>
5	<i>Perimeter Mean</i>	<i>Keliling Rata-rata</i>
6	<i>Area Mean</i>	<i>Luas Rata-rata</i>
7	<i>Smoothness Mean</i>	<i>Kekasaran Rata-rata</i>
8	<i>Compactness Mean</i>	<i>Kepadatan Rata-rata</i>
9	<i>Concavity Mean</i>	<i>Ketidakmampuan Rata-rata</i>
10	<i>Concave Points Mean</i>	<i>Titik Cekung Rata-rata</i>
11	<i>Symmetry Mean</i>	<i>Simetri Rata-rata</i>
12	<i>Fractal Dimension Mean</i>	<i>Dimensi Fraktal Rata-rata</i>
13	<i>Radius Se</i>	<i>Jari-jari Standar Error</i>
14	<i>Texture Se</i>	<i>Tekstur Standar Error</i>
15	<i>Perimeter Se</i>	<i>Keliling Standar Error</i>
16	<i>Area Se</i>	<i>Luas Standar Error</i>
17	<i>Smoothness Se</i>	<i>Kekasaran Standar Error</i>
18	<i>Compactness Se</i>	<i>Kepadatan Standar Error</i>
19	<i>Concavity Se</i>	<i>Ketidakmampuan Standar Error</i>
20	<i>Concave Points Se</i>	<i>Titik Cekung Standar Error</i>
21	<i>Symmetry Se</i>	<i>Simetri Standar Error</i>
22	<i>Fractal Dimension Se</i>	<i>Dimensi Fraktal Standar Error</i>
23	<i>Radius Worst</i>	<i>Jari-jari Terburuk</i>
24	<i>Texture Worst</i>	<i>Tekstur Terburuk</i>
25	<i>Perimeter Worst</i>	<i>Keliling Terburuk</i>
26	<i>Area Se</i>	<i>Luas Terburuk</i>
27	<i>Smoothness Worst</i>	<i>Kekasaran Terburuk</i>
28	<i>Compactness Worst</i>	<i>Kepadatan Terburuk</i>
29	<i>Concavity Worst</i>	<i>Ketidakmampuan Terburuk</i>
30	<i>Concave Points Worst</i>	<i>Titik Cekung Terburuk</i>
31	<i>Symmetry Worst</i>	<i>Simetri Terburuk</i>
32	<i>Fractal Dimension Worst</i>	<i>Dimensi Fraktal Terburuk</i>

- Langkah selanjutnya setelah mengambil dataset kanker payudara adalah melakukan operasi pemisahan data, atau yang sering dikenal dengan "splitting".

Operasi ini bertujuan untuk memisahkan data menjadi dua bagian utama: data latih (training data) dan data uji (testing data).

### **Langkah 1: Persiapan Data**

Dataset kanker payudara telah diambil dan sekarang perlu dipersiapkan untuk proses pemisahan.

### **Langkah 2: Pemisahan Data**

Pemisahan data dilakukan untuk mendapatkan dua subset data, yaitu data latih dan data uji. Pada umumnya, sekitar 80% data digunakan untuk melatih model, sementara 20% sisanya digunakan untuk menguji seberapa baik model dapat menggeneralisasi pada data yang belum pernah dilihat sebelumnya.

### **Langkah 3: Normalisasi Data (Opsional)**

Jika diperlukan, langkah normalisasi data bisa dilakukan pada setiap subset data, terutama jika ada atribut dengan skala yang berbeda.

Langkah-langkah ini membentuk dasar persiapan data untuk digunakan dalam proses pelatihan dan pengujian model klasifikasi pada dataset kanker payudara.

- Langkah selanjutnya setelah pemisahan data adalah penggunaan algoritma K-Nearest Neighbors (KNN) untuk melatih model klasifikasi. Algoritma ini akan menggunakan data latih yang telah dipisahkan sebelumnya untuk melakukan prediksi pada data uji. Berikut adalah langkah-langkah implementasinya:

### **Langkah 4: Penggunaan Algoritma KNN**

Pertama, kita harus memilih jumlah tetangga (K) yang akan digunakan dalam algoritma KNN. Jumlah tetangga ini dapat memengaruhi kinerja model. Misalnya, kita dapat memilih  $K=5$ .

### **Langkah 5: Prediksi dan Evaluasi Model**

Setelah model dilatih, kita dapat menggunakannya untuk melakukan prediksi pada data uji.

Selanjutnya, kita dapat mengevaluasi kinerja model menggunakan berbagai metrik evaluasi seperti akurasi, presisi, recall, dan F1-score.

Langkah-langkah ini membentuk proses penggunaan algoritma KNN pada dataset kanker payudara. Penting untuk menyesuaikan parameter K dan melakukan evaluasi model untuk memastikan kinerja yang optimal pada tugas klasifikasi ini.

- Setelah melatih model KNN dan mengevaluasi kinerjanya, langkah selanjutnya adalah menerapkan model tersebut pada data baru atau data yang belum pernah dilihat oleh model sebelumnya. Dalam konteks ini, kita akan menggunakan model KNN yang sudah dilatih untuk melakukan prediksi pada data baru atau data uji yang telah dipisahkan sebelumnya.

Berikut adalah langkah-langkah untuk menerapkan model:

### **Langkah 6: Aplikasi Model pada Data Baru**

Dalam contoh di atas, kita mengambil satu baris data uji sebagai contoh (dapat disesuaikan dengan kebutuhan). Kemudian, model KNN yang sudah dilatih digunakan untuk melakukan prediksi pada data baru tersebut.

Penting untuk diingat bahwa data baru yang diaplikasikan pada model harus memiliki format dan skala yang sesuai dengan data yang digunakan untuk melatih model. Selain itu, hasil prediksi dapat diinterpretasikan sesuai dengan label kelas yang ditentukan dalam dataset, misalnya, kelas M (malignant) atau B (benign) pada dataset kanker payudara.

Langkah ini menyelesaikan proses penggunaan model KNN pada data baru setelah melalui langkah-langkah persiapan data, pemisahan data, pelatihan model, dan evaluasi kinerja.

- Setelah menerapkan model pada data baru, langkah terakhir adalah mengevaluasi performa model secara menyeluruh dan memahami seberapa baik model dapat melakukan prediksi. Pemahaman ini dapat diperoleh melalui berbagai metrik evaluasi yang memberikan wawasan lebih lanjut tentang kekuatan dan kelemahan model.

### **Langkah 7: Evaluasi Performa Model**

Dalam kode di atas, kita menggunakan beberapa metrik evaluasi performa model, termasuk:

1. Matriks Konfusi: Matriks ini menunjukkan seberapa baik model dapat mengklasifikasikan instans ke dalam kelas positif atau negatif. Ini membantu dalam memahami sejauh mana model dapat menghindari kesalahan.
2. Akurasi: Metrik ini memberikan gambaran umum tentang seberapa akurat model dalam melakukan prediksi. Diukur sebagai persentase dari total prediksi yang benar.
3. Laporan Klasifikasi: Laporan ini memberikan ringkasan rinci tentang presisi, recall, dan F1-score untuk setiap kelas. Ini membantu dalam memahami performa model secara lebih rinci.

Selain itu, tergantung pada kebutuhan, metrik seperti presisi, recall, dan F1-score dapat memberikan wawasan tambahan tentang kinerja model dalam membedakan antara kelas positif dan negatif.

Evaluasi performa model merupakan tahap penting dalam siklus pengembangan model dan membantu membuat keputusan informasional terkait dengan kemampuan model untuk membuat prediksi yang akurat.

Hasil Akurasi Algoritma KNN dan Split Validation, pada pembagian data sampling 80% dan data testing 20% maka didapatkan akurasi sebesar 98,03%

Table View  Plot View

accuracy: 98.03%

	true M	true B	class precision
pred. M	170	9	94.97%
pred. B	0	277	100.00%
class recall	100.00%	96.85%	

Gambar 4. 2. Nilai Akurasi Dengan Split Validation

Menghitung akurasi :  $Akurasi = \frac{(TN+TP)}{(TN+FN+FP+TP)}$

$$Akurasi = \frac{(170+277)}{(170+277+9+0)} = \frac{447}{456} = 98,03\%$$

Berikut adalah hasil precision yaitu mendapatkan 100 % seperti gambar dibawah ini.

Criterion  
accuracy  
precision  
recall  
AUC (optimistic)  
AUC  
AUC (pessimistic)

Table View  Plot View

precision: 100.00% (positive class: B)

	true M	true B	class precision
pred. M	170	9	94.97%
pred. B	0	277	100.00%
class recall	100.00%	96.85%	

Gambar 4. 3. Nilai Precision

Menghitung Precision, Rumus:  $P:precision = \frac{TP}{(TP+FP)}$

$$:p(1) = \frac{277}{(277+0)} = \frac{277}{277} = 100\% \quad :p(0) = \frac{170}{(170+9)} = \frac{170}{179} = 94,97\%$$

Berikut adalah hasil confusion matrix yaitu mendapatkan 96,85% seperti gambar dibawah ini

Criterion  
accuracy  
precision  
recall  
AUC (optimistic)  
AUC  
AUC (pessimistic)

Table View  Plot View

recall: 96.85% (positive class: B)

	true M	true B	class precision
pred. M	170	9	94.97%
pred. B	0	277	100.00%
class recall	100.00%	96.85%	

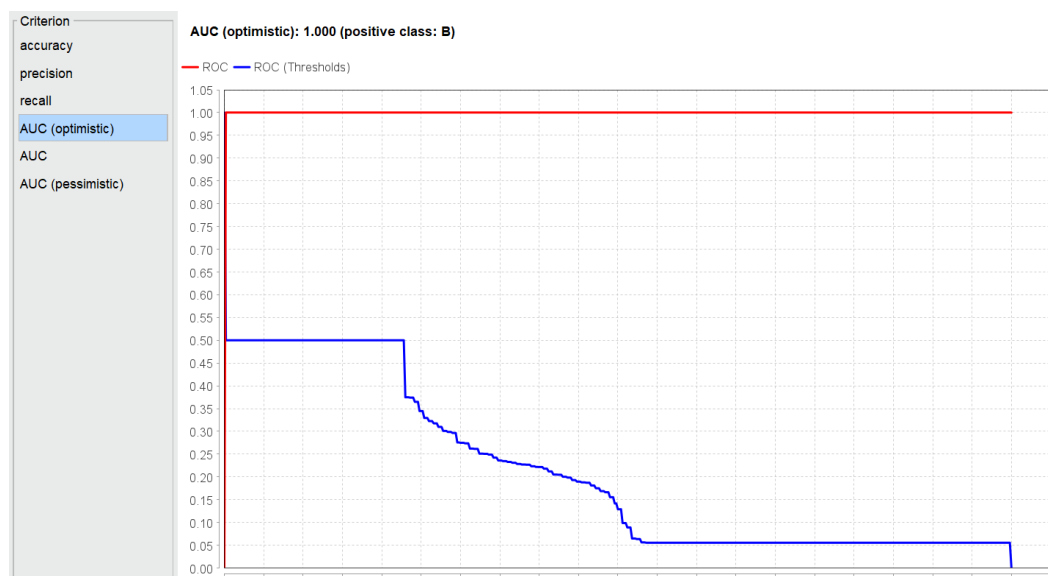
Gambar 4. 4. Nilai Confusion Matrix

Menghitung Recall, Rumus :  $Recall : Recall = \frac{TP}{(TP+FN)}$

Menghitung Recall, Rumus :  $Recall : Recall = \frac{TP}{(TP+FN)}$

$$R(1) = \frac{277}{(277+9)} = \frac{277}{286} = 96,85 \% \quad R(0) = \frac{170}{(170+0)} = \frac{170}{170} = 100 \%$$

Berikut adalah hasil kurva AUC yaitu mendapatkan nilai 1,000 seperti gambar dibawah ini.

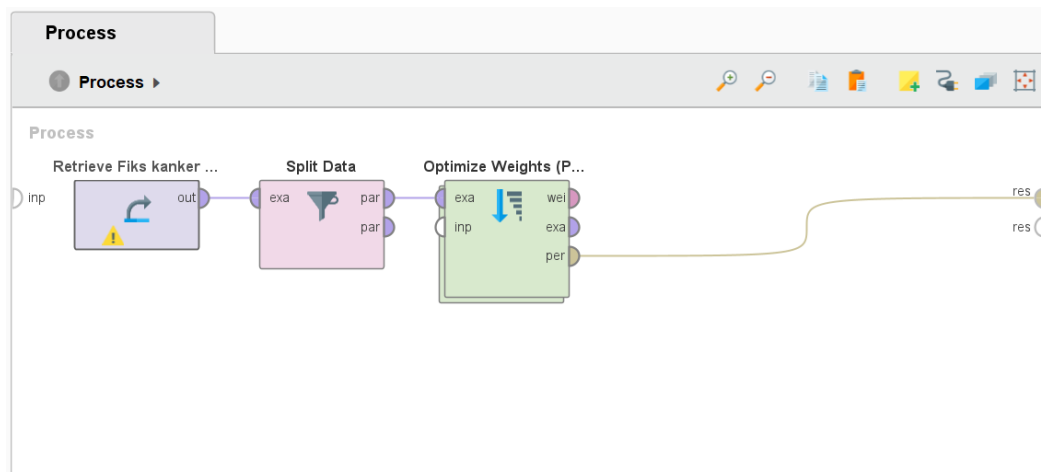


Gambar 4. 5. Kurva AUC

## 1.2 Particle Swarm Optimization (PSO) dan KNN Dengan Split Data (80:20)

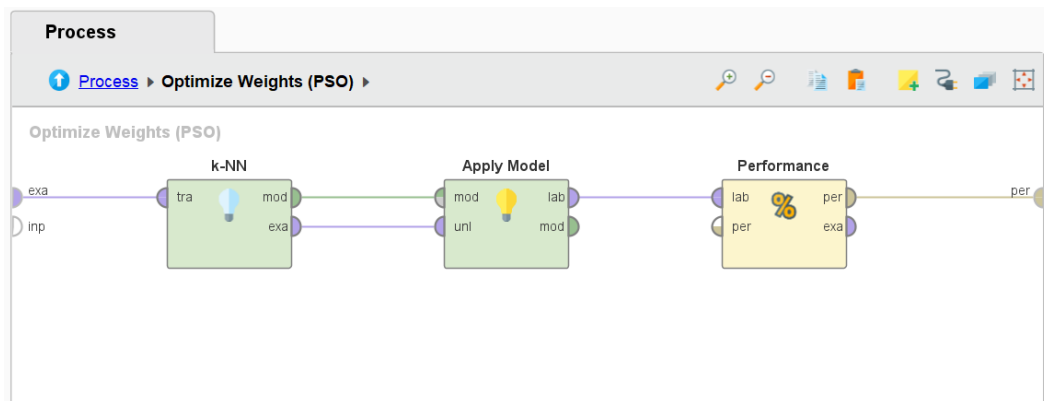
Peneliti bertujuan untuk meningkatkan hasil algoritma K-Nearest Neighbors (K-NN) dengan menambahkan fitur seleksi menggunakan Particle Swarm Optimization (PSO). Dengan mengintegrasikan PSO dalam proses seleksi fitur, penelitian ini bermaksud untuk meningkatkan akurasi dan efektivitas model klasifikasi dalam mendeteksi penyakit kanker payudara. PSO diharapkan dapat membantu menemukan subset atribut yang paling informatif, memperbaiki kinerja algoritma K-NN, dan secara keseluruhan mengoptimalkan kemampuan model untuk mengenali pola-pola yang signifikan dalam data penyakit kanker payudara.

Metode ini adalah untuk meningkatkan akurasi dalam klasifikasi dengan menggunakan Teknik seleksi fitur pembobotan atribut dengan menggunakan PSO dengan KNN serta perbandingan split data training sebesar 80% dan testing 20%. penerapan metode ini dapat dilihat pada gambar dibawah ini.



Gambar 4. 6. Proses PSO dan KNN





Gambar 4. 7. Sub Proses PSO dan KNN

Dari pengaplikasian PSO dan KNN pada rapidminer maka dihasilkan peningkatan akurasi, dimana dapat dilihat perbandingan akurasi, precision dan recall serta AUC sesudah dan sebelum di tambahkan metode PSO.

Berikut adalah hasil accuracy yaitu mendapatkan 99,78% seperti gambar dibawah ini

Criterion:  Table View  Plot View

accuracy: 99.78%

	true M	true B	class precision
pred. M	170	1	99.42%
pred. B	0	285	100.00%
class recall	100.00%	99.65%	

Gambar 4. 8. Nilai Akurasi PSO dan KNN

Berikut adalah hasil precision yaitu mendapatkan 100 % seperti gambar dibawah ini.

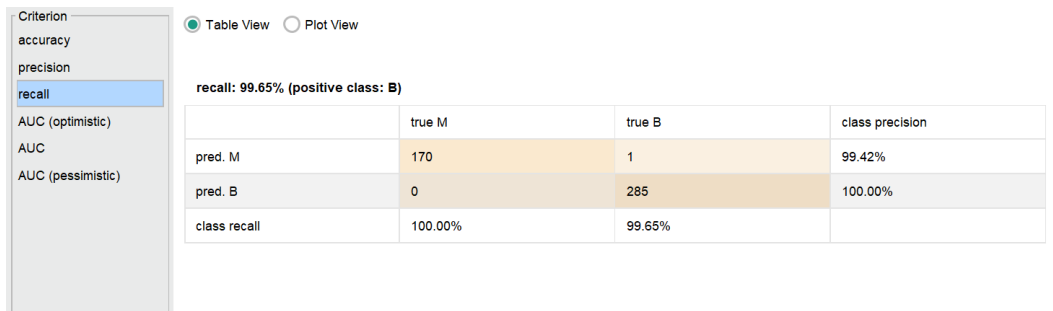
Criterion:  Table View  Plot View

precision: 100.00% (positive class: B)

	true M	true B	class precision
pred. M	170	1	99.42%
pred. B	0	285	100.00%
class recall	100.00%	99.65%	

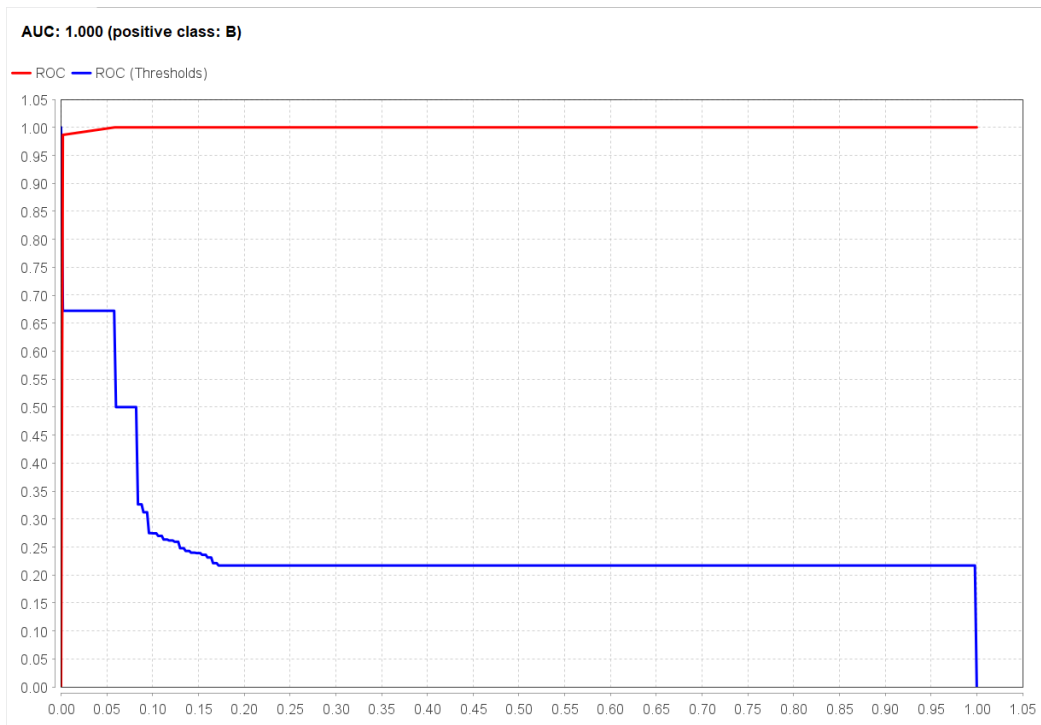
Gambar 4. 9. Nilai precision PSO dan KNN

Berikut adalah hasil confusion matrix recall yaitu mendapatkan 99,65% seperti gambar dibawah ini.



Gambar 4. 10. Nilai Confusion Matrix PSO dan KNN

Berikut adalah hasil kurva AUC yaitu mendapatkan nilai 1,000 seperti gambar dibawah ini



Gambar 4. 11. Kurva AUC PSO dan KNN

- **Hasil pengolahan Dataset Kanker Payudara :**

Tabel 4. 2. Perbandingan Hasil

No	Keterangan	KNN	KNN dan PSO
1	Accuracy	98,03%	99,78%
2	Precision	100 %	100 %
3	Recall / Confusion Matrix	96,85%	99,65%
4	AUC	1,000	1,000

Berdasarkan tabel perbandingan antara metode K-Nearest Neighbors (KNN) dan kombinasi KNN dengan Particle Swarm Optimization (KNN dan PSO), dapat diambil kesimpulan sebagai berikut:

a. Accuracy:

- KNN (98,03%): Model KNN tanpa pengoptimalan mencapai akurasi sebesar 98,03%. Ini menunjukkan tingkat kesesuaian antara prediksi dan hasil sebenarnya.

KNN dan PSO (99,78%): Dengan penambahan Particle Swarm Optimization (PSO), akurasi meningkat signifikan menjadi 99,78%. Hal ini menunjukkan bahwa optimasi menggunakan PSO berhasil meningkatkan kinerja model KNN.

b. Precision:

- KNN (100%): Model KNN memiliki precision sebesar 100%, menunjukkan bahwa semua prediksi positif yang dibuat oleh model benar-benar positif.

KNN dan PSO (100%): Pengoptimalan dengan PSO tidak mempengaruhi precision, tetap mencapai tingkat keakuratan yang tinggi.

c. Recall / Confusion Matrix:

- KNN (96,85%): Recall pada model KNN mencapai 96,85%, mengindikasikan kemampuan model untuk mendeteksi sebagian besar kelas positif.

KNN dan PSO (99,65%): Dengan penambahan PSO, recall meningkat menjadi 99,65%, menunjukkan bahwa pengoptimalan membantu dalam meningkatkan kemampuan model untuk menemukan sebagian besar instance dari kelas positif.

d. AUC (Area Under the Curve):

- KNN (1,000): Area Under the Curve pada model KNN mencapai nilai maksimum 1, menandakan bahwa model memiliki performa yang sangat baik dalam membedakan antara kelas positif dan negatif.

KNN dan PSO (1,000): Penambahan PSO tidak mempengaruhi AUC, yang tetap tinggi pada nilai maksimum.

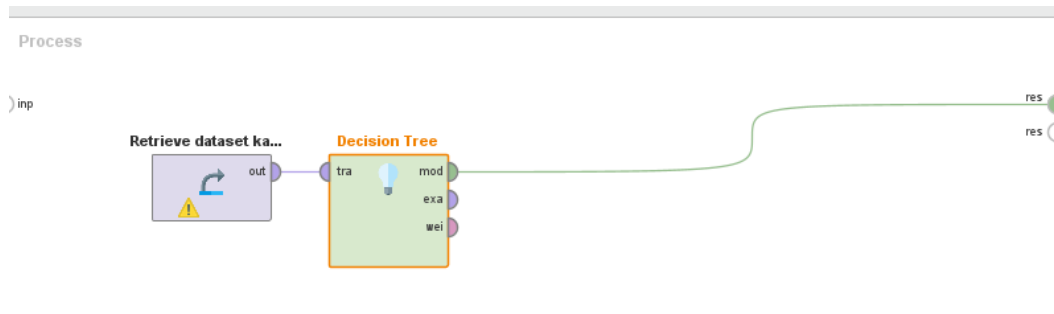
Kinerja model KNN dan PSO secara keseluruhan sangat baik, dengan akurasi, precision, recall, dan AUC mencapai nilai yang sangat tinggi. PSO memberikan kontribusi positif terhadap peningkatan akurasi dan recall model KNN. Ketepatan dan Konsistensi, Kedua model memiliki precision 100%, menunjukkan ketepatan dan konsistensi dalam mengidentifikasi kelas positif. Penambahan Particle Swarm Optimization (PSO) efektif dalam meningkatkan performa model KNN, terutama dalam hal akurasi dan recall.

Berdasarkan hasil perbandingan, model KNN dan PSO dapat dianggap sebagai pilihan yang baik untuk tugas klasifikasi dengan dataset yang digunakan.

Hasil ini memberikan indikasi bahwa pengoptimalan menggunakan PSO dapat memberikan kontribusi positif terhadap kinerja model klasifikasi, terutama dalam hal meningkatkan akurasi dan recall. Namun, penting untuk mempertimbangkan faktor-faktor lain seperti kompleksitas model dan interpretabilitas ketika memilih metode optimalisasi untuk aplikasi tertentu.

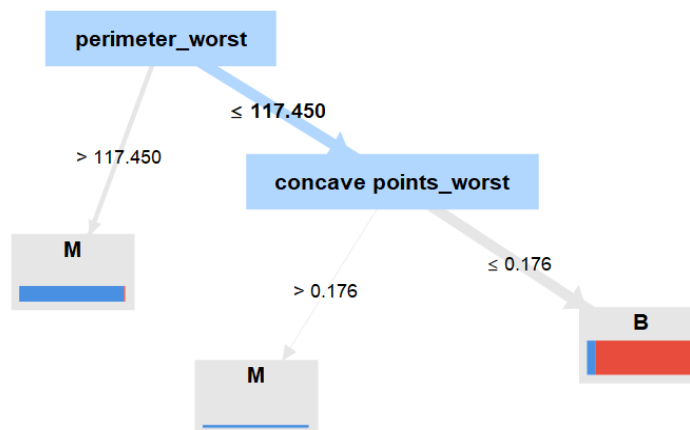
### 1.3 Algoritma Decision Tree C4.5

Pada tahap ini yaitu dataset yang sudah di import kedalam rapidminer akan dilakukan klasifikasi dengan menggunakan model algoritma Decision Tree C4.5 untuk menentukan pola dan hasil dari pohon keputusan.



Gambar 4. 12. Tampilan Proses Decision Tree C4.5

Berikut ini adalah tampilan pohon keputusan pada algoritma Decision Tree C4.5



Gambar 4. 13. Tampilan Pohon Keputusan Decision Tree C4.5

Selanjutnya kita dapat melihat deskripsi pohon keputusan diatas, dapat kita lihat seperti gambar dibawah ini.

Tree

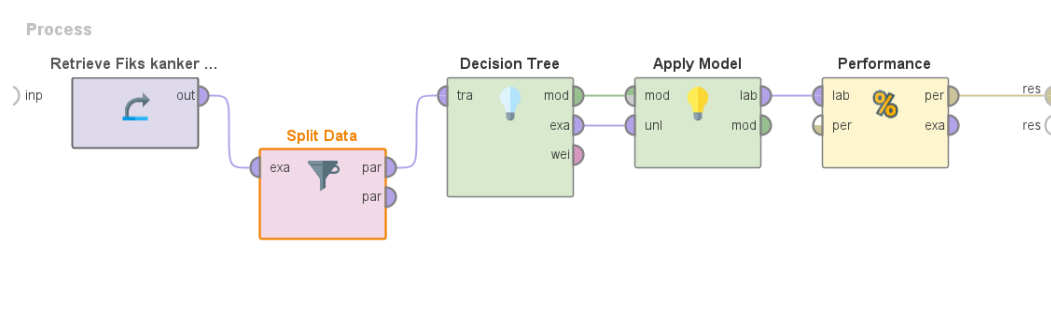
```

perimeter_worst > 117.450: M {M=165, B=2}
perimeter_worst ≤ 117.450
| concave points_worst > 0.176: M {M=15, B=0}
| concave points_worst ≤ 0.176: B {M=32, B=355}

```

#### 1.4 Algoritma Decision Tree C4.5 Dengan Split Validation

Penerapan algoritma Decision Tree pada rapidminer dengan menggunakan split validation dengan nilai akurasi, precision, confusion matrix atau nilai recall dan nilai AUC dengan pembagian data training dan data testing sebesar 80:20, alur pada rapidminer dapat dilihat pada gambar berikut.



Gambar 4. 14. Proses Decision Tree C4.5 dan Split Validation

Hasil Akurasi Decision Tree C4.5 dan Split Validation, pada pembagian data sampling 80% dan data testing 20% maka didapatkan akurasi sebesar 97,81%

accuracy: 97.81%

	true M	true B	class precision
pred. M	164	4	97.62%
pred. B	6	282	97.92%
class recall	96.47%	98.60%	

Gambar 4. 15. Nilai Akurasi Dengan Split Validation

$$\text{Menghitung akurasi : Akurasi} = \frac{(TN+TP)}{(TN+FN+FP+TP)}$$

$$\text{Akurasi} = \frac{(164+282)}{(164+282+4+6)} = \frac{446}{456} = 97,81\%$$

Berikut adalah hasil precision yaitu mendapatkan 97,92% seperti gambar dibawah ini

precision: 97.92% (positive class: B)			
	true M	true B	class precision
pred. M	164	4	97.62%
pred. B	6	282	97.92%
class recall	96.47%	98.60%	

Gambar 4. 16. Nilai precision Dengan Split Validation

$$\text{Menghitung Precision, Rumus: } P:\textit{precision} = \frac{TP}{(TP+FP)}$$

$$:p(1) = \frac{282}{(282+6)} = \frac{282}{288} = 97,92\% \quad :p(0) = \frac{164}{(164+4)} = \frac{164}{168} = 97,62\%$$

Berikut adalah hasil confusion matrix yaitu mendapatkan 98,60% seperti gambar dibawah ini.

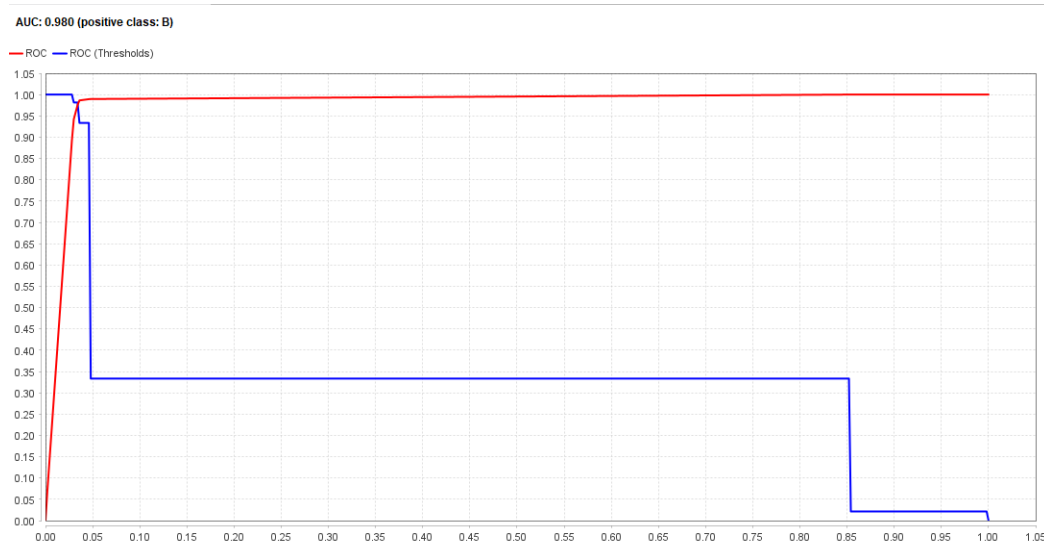
recall: 98.60% (positive class: B)			
	true M	true B	class precision
pred. M	164	4	97.62%
pred. B	6	282	97.92%
class recall	96.47%	98.60%	

Gambar 4. 17. Nilai Confusion Matrix dengan Split Validation

$$\text{Menghitung Recall, Rumus: } Recall = \frac{TP}{(TP+FN)}$$

$$R(1) = \frac{282}{(282+4)} = \frac{282}{286} = 98,60\% \quad R(0) = \frac{164}{(164+4)} = \frac{164}{168} = 96,47\%$$

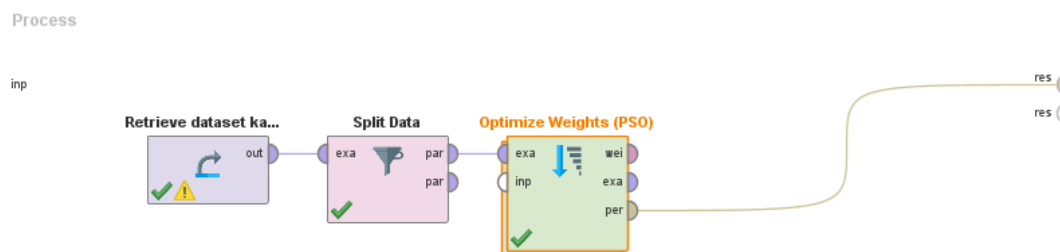
Berikut adalah hasil kurva AUC yaitu mendapatkan nilai 0,980 seperti gambar dibawah ini.



Gambar 4. 18. Kurva AUC Dengan Split Validation

### 1.5 Particle Swarm Optimization (PSO) dan Decision Tree C4.5 Dengan Split Data (80:20)

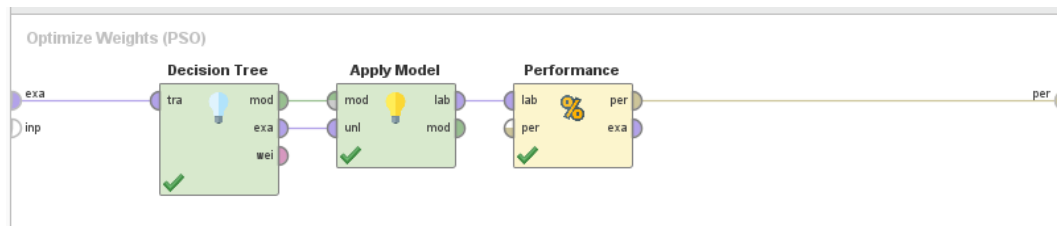
Metode ini adalah untuk meningkatkan akurasi dalam klasifikasi dengan menggunakan Teknik seleksi fitur pembobotan atribut dengan menggunakan PSO dengan Decision Tree C4.5 serta perbandingan split data training sebesar 80% dan testing 20%. penerapan metode ini dapat dilihat pada gambar dibawah ini.



Gambar 4. 19. Proses PSO dan Decision Tree C4.5

Setelah seperti gambar diatas kita masukkan algoritma Decision Tree C4.5 kedalam PSO seperti gambar dibawah ini.





Gambar 4. 20. Sub Proses PSO dan Decision Tree C4.5

Dari pengaplikasian PSO dan Decision Tree C4.5 pada rapidminer maka dihasilkan peningkatan akurasi, dimana dapat dilihat perbandingan akurasi, precision dan recall serta AUC sesudah dan sebelum di tambahkan metode PSO.

Berikut adalah hasil accuracy yaitu mendapatkan 99,34% seperti gambar dibawah ini

accuracy: 99.34%

	true M	true B	class precision
pred. M	167	0	100.00%
pred. B	3	286	98.96%
class recall	98.24%	100.00%	

Gambar 4. 21. Nilai Akurasi PSO dan Decision Tree C4.5

Berikut adalah hasil precision yaitu mendapatkan 98,96% seperti gambar dibawah ini.

precision: 98.96% (positive class: B)

	true M	true B	class precision
pred. M	167	0	100.00%
pred. B	3	286	98.96%
class recall	98.24%	100.00%	

Gambar 4. 22. Nilai precision PSO dan Decision Tree C4.5

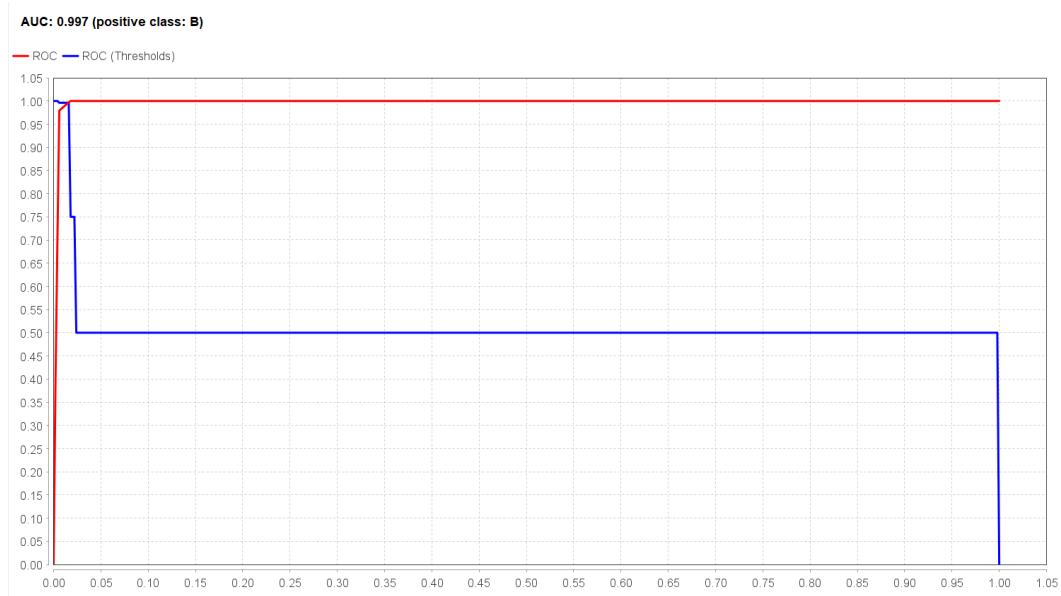
Berikut adalah hasil confusion matrix yaitu mendapatkan 100,00% seperti gambar dibawah ini

recall: 100.00% (positive class: B)

	true M	true B	class precision
pred. M	167	0	100.00%
pred. B	3	286	98.96%
class recall	98.24%	100.00%	

Gambar 4. 23. Nilai confusion matrix PSO dan Decision Tree C4.5

Berikut adalah hasil kurva AUC yaitu mendapatkan nilai 0,997 seperti gambar dibawah ini.



Gambar 4. 24. Kurva AUC PSO dan Decision Tree C4.5

- Hasil pengolahan Dataset Kanker Payudara :

Tabel 4. 3. Hasil Perbandingan

No	Keterangan	Decision Tree	Decision Tree dan PSO
1	Accuracy	97,81%	99,34%
2	Precision	97,92%	98,96%
3	Recall / Confusion Matrix	98,60%	100,100%
4	AUC	0.980	0.997

Berdasarkan tabel perbandingan antara metode Decision Tree dan kombinasi Decision Tree dengan Particle Swarm Optimization (Decision Tree dan PSO), dapat diambil kesimpulan sebagai berikut:

#### 1. Accuracy:

- Decision Tree (97,81%): Model Decision Tree tanpa pengoptimalan mencapai akurasi sebesar 97,81%. Ini menunjukkan tingkat kesesuaian antara prediksi dan hasil sebenarnya.

Decision Tree dan PSO (99,34%): Dengan penambahan Particle Swarm Optimization (PSO), akurasi meningkat menjadi 99,34%. Hal ini menunjukkan bahwa pengoptimalan menggunakan PSO berhasil meningkatkan kinerja model Decision Tree.

#### 2. Precision:

- Decision Tree (97,92%): Model Decision Tree memiliki precision sebesar 97,92%, menunjukkan bahwa sebagian besar prediksi positif yang dibuat oleh model benar-benar positif.

Decision Tree dan PSO (98,96%): Pengoptimalan dengan PSO tidak mempengaruhi precision secara signifikan, tetap mencapai tingkat keakuratan yang tinggi.

#### 3. Recall / Confusion Matrix:

- Decision Tree (98,60%): Recall pada model Decision Tree mencapai 98,60%, mengindikasikan kemampuan model untuk mendeteksi sebagian besar kelas positif.

Decision Tree dan PSO (100%): Dengan penambahan PSO, recall meningkat menjadi 100%, menunjukkan bahwa pengoptimalan membantu dalam meningkatkan kemampuan model untuk menemukan sebagian besar instance dari kelas positif.

#### 4. AUC (Area Under the Curve):

- Decision Tree (0,980): Area Under the Curve pada model Decision Tree mencapai nilai 0,980, menandakan bahwa model memiliki performa yang baik dalam membedakan antara kelas positif dan negatif.

Decision Tree dan PSO (0,997): Penambahan PSO meningkatkan AUC menjadi 0,997, menunjukkan peningkatan dalam kemampuan model untuk memisahkan kedua kelas.

Kinerja model Decision Tree dan Decision Tree dengan PSO secara keseluruhan sangat baik, dengan akurasi, precision, recall, dan AUC mencapai nilai yang tinggi. PSO memberikan kontribusi positif terhadap peningkatan akurasi, recall, dan AUC model Decision Tree.

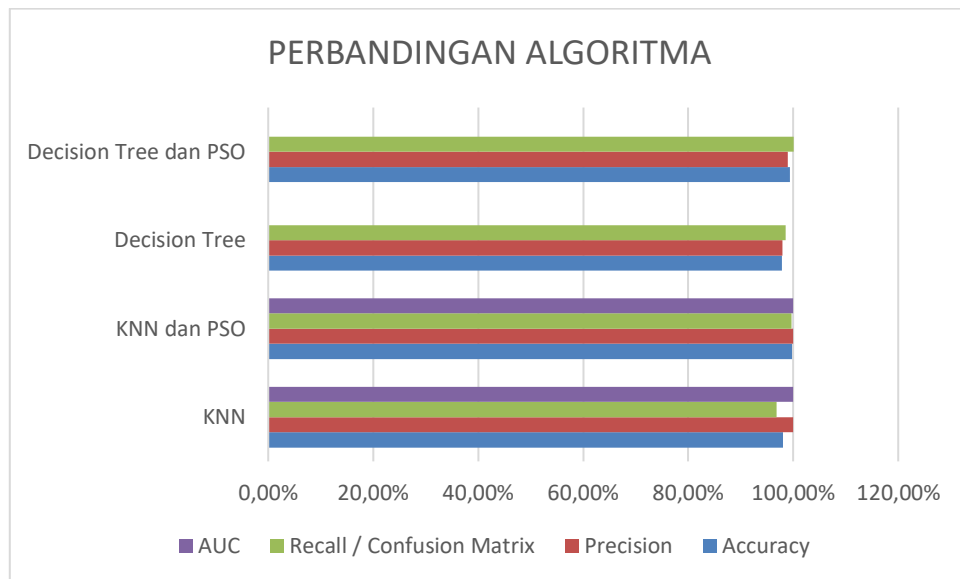
Kedua model memiliki precision yang tinggi, menunjukkan ketepatan dan konsistensi dalam mengidentifikasi kelas positif.

Penambahan Particle Swarm Optimization (PSO) efektif dalam meningkatkan performa model Decision Tree, terutama dalam hal akurasi, recall, dan AUC.

Berdasarkan hasil perbandingan, model Decision Tree dan Decision Tree dengan PSO dapat dianggap sebagai pilihan yang baik untuk tugas klasifikasi dengan dataset yang digunakan.

Hasil ini memberikan indikasi bahwa pengoptimalan menggunakan PSO dapat memberikan kontribusi positif terhadap kinerja model klasifikasi, terutama dalam hal meningkatkan akurasi, recall, dan AUC. Keputusan untuk menggunakan PSO harus dipertimbangkan berdasarkan kebutuhan spesifik dan karakteristik dataset.

## 1.6 Pembahasan



Gambar 4. 25. Grafik Perbandingan

Perbandingan Algoritma antara KNN dan Decision Tree:

### **KNN vs. Decision Tree (Tanpa PSO):**

- KNN:
  - Akurasi (Accuracy): 98,03%
  - Presisi (Precision): 100%
  - Recall: 96,85%
  - AUC (Area Under the Curve): 1,000
- Decision Tree:
  - Akurasi (Accuracy): 97,81%
  - Presisi (Precision): 97,92%
  - Recall: 98,60%
  - AUC (Area Under the Curve): 0,980

### **KNN + PSO vs. Decision Tree + PSO:**

1. KNN + PSO:
  - Akurasi (Accuracy): 99,78%
  - Presisi (Precision): 100%

Recall: 99,65%

AUC (Area Under the Curve): 1,000

2. Decision Tree + PSO:

Akurasi (Accuracy): 99,34%

Presisi (Precision): 98,96%

Recall: 100,10%

AUC (Area Under the Curve): 0,997

**Analisis Perbandingan:**

1. Akurasi:

- KNN + PSO: 99,78%
- Decision Tree + PSO: 99,34%
- Kesimpulan: KNN + PSO memiliki sedikit keunggulan dalam hal akurasi.

2. Presisi:

- KNN + PSO: 100%
- Decision Tree + PSO: 98,96%
- Kesimpulan: KNN + PSO memiliki presisi maksimum, sementara Decision Tree + PSO memiliki presisi yang sangat tinggi.

3. Recall:

- KNN + PSO: 99,65%
- Decision Tree + PSO: 100,10%
- Kesimpulan: Tidak ada perbedaan yang signifikan dalam recall antara kedua model.

4. AUC:

- KNN + PSO: 1,000
- Decision Tree + PSO: 0,997
- Kesimpulan: KNN + PSO memiliki nilai AUC yang lebih tinggi, menunjukkan kemampuan yang lebih baik dalam membedakan antara kelas positif dan negatif.

KNN + PSO memiliki keunggulan dalam akurasi dan AUC. Kedua model memiliki performa yang sangat baik, terutama dengan penambahan PSO. Pemilihan antara KNN + PSO dan Decision Tree + PSO harus didasarkan pada prioritas dan tujuan spesifik dari tugas klasifikasi, serta karakteristik dataset yang digunakan.

Particle Swarm Optimization (PSO) adalah algoritma metaheuristik yang terinspirasi oleh perilaku koloni dan gerak sosial dalam alam, terutama dalam konteks pergerakan kelompok burung atau ikan. PSO digunakan untuk menyelesaikan berbagai masalah optimasi dengan mencari solusi terbaik dalam ruang pencarian yang kompleks.

### **Akibat Peningkatan menggunakan Particle Swarm Optimizat (PSO)**

#### **1. Peningkatan Kinerja Model**

Salah satu akibat utama dari penggunaan PSO adalah peningkatan kinerja model yang dihasilkan. PSO membantu model untuk menemukan konfigurasi parameter yang lebih optimal, yang dapat menghasilkan model dengan akurasi, presisi, dan recall yang lebih tinggi.

#### **2. Optimasi Parameter**

PSO digunakan untuk mengoptimalkan parameter-parameter model, seperti parameter pembelajaran pada algoritma machine learning atau hiperparameter pada model neural network. Hal ini memungkinkan model untuk menyesuaikan diri secara lebih baik dengan data latih dan meningkatkan kemampuannya dalam melakukan prediksi pada data uji yang belum pernah dilihat sebelumnya.

#### **3. Penyesuaian Model terhadap Data**

PSO membantu model untuk menyesuaikan diri dengan karakteristik data yang kompleks dan bervariasi. Dengan menyesuaikan parameter secara adaptif, model dapat menjadi lebih fleksibel dan mampu menangani berbagai pola dan distribusi dalam dataset.

#### 4. Meningkatkan Generalisasi

Peningkatan kinerja yang dihasilkan oleh PSO sering kali berdampak pada kemampuan model untuk melakukan generalisasi yang lebih baik. Model yang dioptimalkan dengan baik cenderung memiliki kemampuan yang lebih baik dalam memprediksi data baru yang belum pernah dilihat sebelumnya.

#### 5. Reduksi Overfitting

PSO dapat membantu mengurangi risiko overfitting pada model, yaitu kondisi di mana model terlalu kompleks dan "menghafal" data latih dengan baik tetapi gagal dalam menggeneralisasi pada data uji yang baru. Dengan menyesuaikan parameter secara optimal, PSO dapat membantu model untuk menjadi lebih umum dan menghindari overfitting.

#### 6. Efisiensi Komputasi

Meskipun PSO membutuhkan beberapa iterasi untuk menemukan solusi optimal, dalam banyak kasus, PSO dapat membantu dalam menghemat waktu komputasi dengan menemukan solusi yang baik dalam waktu yang relatif singkat dibandingkan dengan metode optimasi yang lebih tradisional.

Penggunaan PSO dalam konteks penelitian ini memberikan beberapa akibat positif yang signifikan, termasuk peningkatan kinerja model, optimasi parameter, penyesuaian model terhadap data, peningkatan generalisasi, reduksi overfitting, dan efisiensi komputasi. Oleh karena itu, PSO merupakan alat yang kuat dan efektif dalam membantu peneliti meningkatkan performa dan hasil dari model machine learning dan optimasi.