

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Dataset yang digunakan dalam penelitian ini di ambil dari *crawling* media social *Twitter* menggunakan *Software RapidMiner 9.10*. Data yang diambil dibatasi berdasarkan *tweet* yang memiliki tagar #kurikulummerdeka. Data *tweet* yang *crawling* dengan aplikasi *RapidMiner* tidak semuanya dapat diklasifikasikan kedalam kelas positif, negative dan netral. Maka diperlukan proses klasifikasi secara manual, dengan melakukan proses *Labeling* untuk memberikan label, label dikategorikan kedalam tiga kelas yaitu kelas Positif, Negatif dan juga Netral. Jumlah data yang didapatkan dari social media *Twitter* berjumlah sekitar 1.600 data *tweet*. Dari data tersebut kemudian dilakukan tahapan *preprocessing* dan selanjutnya dilakukan pembagian data yaitu data *testing* dan data *training*. Setelah dilakukan tahapan *preprocessing* dataset berkurang dan menjadi 1.217 dataset yang kemudian dibagi menjadi data *Traning* dan Data *Testing* yang masing-masing berjumlah 30% dan 70%. Adapun berikut rincian hasil pemrosesan.

4.1.1 Klasifikasi Text dengan Algoritma *Naïve Bayes* dan *Suport Vector Machine*

Dalam proses pengklasifikasian tekt dengan menggunakan Algoritma *Naïve Bayes* dan *Suport Vector Machine* terdapat beberapa tahap yang dilakukan, dalam penelitian ini penulis mengolah data dengan menggunakan beberapa tahapan yang dijabarkan secara rinci yaitu sebagai berikut :

1. Pengumpulan Data

Pengambilan data pada penelitian ini dibatasi pada *tweet-tweet* yang menggunakan tagar atau #kurikulummerdeka yang didapatkan menggunakan bantuan aplikasi *RapidMiner 9.10*. Data *Tweet* yang didapatkan kemudian disimpan kedalam *Microsoft Excel* untuk dilakukan tahap

selanjutnya yaitu *labelling* atau mengklasifikasikan *tweet* tersebut kedalam kelas positif, negative atau netral tahap ini dilakukan secara manual dan hanya 30% dari keseluruhan data untuk dijadikan data latih.

2. Pengolahan data awal (*Preprocessing*)

Tahapan yang dilakukan dalam pengolahan data awal atau preprocessing adalah:

a. *Case Folding*

Proses *Case Folding* dilakukan menggunakan layanan yang disediakan oleh *RapidMiner*. Karakter yang tidak perlu dalam dokumen dihapus atau dihilangkan. Ini dapat mengakibatkan *noice* seperti *emoticon*, tanda baca, dll. Proses ini juga mengubah huruf dalam dokumen menjadi huruf kecil. Hasil perbandingan teks sebelum dan sesudah dilakukan proses *Case Folding* dapat dilihat pada table 4.1 berikut ini:

Tabel 4.1 Perbandingan teks sebelum dan sesudah dilakukan proses *Case Folding*

Komentar	Komentar (Hasil <i>Case Folding</i>)
Kurikulum Merdeka Jadikan Siswa Lebih Kreatif :)	kurikulum merdeka jadikan siswa lebih kreatif
Ini kurikulum merdeka tujuannya nyiksa batin kali ya??	ini kurikulum merdeka tujuannya nyiksa batin kali ya

b. *Stopword Removal* dimana tahapan ini akan menghilangkan kata yang tidak penting seperti kata “di”, “dan”, “karena” “oleh” dan lain sebagainya. Dalam proses ini, kata-kata sambung atau kata-kata yang tidak relevan akan dihapus, seperti kata tetapi, untuk, dengan, yang, pada dan kata sambung lainnya. Kata-kata tersebut merupakan kata-kata yang tidak mempunyai makna tersendiri jika dipisahkan dengan kata yang lain dan tidak terkait dengan kata sifat yang berhubungan dengan sentiment. Tahap ini dilakukan agar dapat memperbesar nilai akurasi. Hasil perbandingan teks sebelum dan setelah dilakukan proses *stopword Removal* dapat dilihat pada table 4.2 berikut :

Tabel 4.2 Perbandingan teks sebelum dan sesudah dilakukan proses *Stopword Removal*

Komentar	Komentar (Hasil <i>Stopword Removal</i>)
Kurikulum Merdeka Jadikan Siswa Lebih Kreatif	kurikulum merdeka jadikan siswa kreatif
Ini kurikulum merdeka tujuannya nyiksa batin kali ya	ini kurikulum merdeka tujuannya nyiksa batin ya

- c. *Tokenisasi*, Proses Tokenisasi menyeleksi membagi dan memisah kata pada dokumen menjadi term berdasarkan spasi. Hasil perbandingan teks sebelum dan sesudah dilakukan proses *Tokenize* dapat dilihat pada table 4.3 berikut ini :

Tabel 4.3 Perbandingan teks sebelum dan sesudah dilakukan proses *Tokenize*

Komentar	Hasil <i>Tokenisasi</i>
kurikulum merdeka jadikan siswa lebih kreatif	'kurikulum' 'merdeka' 'jadikan' 'siswa' 'lebih' 'kreatif'
ini kurikulum merdeka tujuannya nyiksa batin kali ya	'ini' 'kurikulum' 'merdeka' 'tujuannya' 'nyiksa' 'batin' 'kali' 'ya'

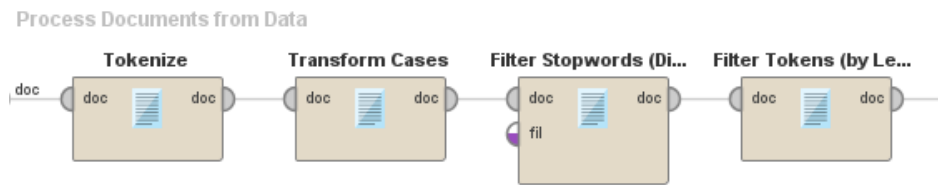
- d. *Filter Token (By Leght)*

Dalam proses ini, kata-kata yang memiliki panjang kurang dari 4 dan lebih dari 25 akan dihapus, seperti kata yg, tdk, jd, ga, ane, gan yang merupakan kata-kata yang tidak mempunyai makna tersendiri jika dipisahkan dengan kata yang lain dan tidak terkait dengan kata sifat yang berhubungan dengan sentiment. Hasil perbandingan teks sebelum dan sesudah dilakukan proses *Filter Token (By Leght)* dapat dilihat pada tabel 4.4 berikut ini :

Tabel 4.4 Perbandingan teks sebelum dan sesudah dilakukan proses *Filter Token (By Leght)*

Komentar	Hasil <i>Tokenisasi</i>
kurikulum merdeka jadikan siswa lebih kreatif	kurikulum merdeka jadikan siswa lebih kreatif
ini kurikulum merdeka tujuannya nyiksa batin kali ya	kurikulum merdeka tujuannya nyiksa batin kali

Desain model proses pengolahan data awal (*Preprocessing*) menggunakan *Rapid Miner* pada penelitian ini dapat dilihat pada gambar 4.1 berikut ini:



Gambar 4.1 Desain Model *Preprocessing Data*

3. Klasifikasi

Jika hasil klasifikasi pada kelas positif lebih besar dari pada kelas negatif, tentukan klausa sebagai anggota kelas positif, negatif, dan netral berdasarkan nilai *Support Vector Machine* dan proses komputasi *Naive Bayes* yang harus dilakukan. , maka apakah proposisi berada pada kelas positif. Jika peluang kelas positif lebih kecil dari kelas negatif, maka proposisi berada pada kelas negatif. Jika kelas positif dan negatif lebih kecil dari kelas netral, maka proposisi berada di kelas negatif, di mana proposisi berada di kelas netral.

4.1.2 Hasil Eksperimen Pengujian Metode

Hasil eksperimen dari metode yang digunakan pada penelitian ini adalah sebagai berikut:

1. Algoritma *Naive Bayes*

Hasil dari pengujian yang dilakukan dari dataset yang berjumlah 1217 data dengan menggunakan Algoritma *Naïve Bayes* menghasilkan nilai *Accuracy* sebesar 75.24% dan nilai *kappa* sebesar 0.626. Hasil Pengujian dapat dilihat pada tabel 4.5 dan 4.6 berikut ini:

Tabel 4.5 Hasil pengujian data menggunakan *Algoritma Naïve Bayes*

accuracy: 75.24% +/- 4.25% (micro average: 75.23%)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	276	43	57	73.40%
pred. Netral	50	376	49	79.16%
pred. Positif	47	55	262	71.98%
class recall	73.99%	79.23%	71.20%	

Tabel 4.6 Nilai *Kappa* dari pengujian *Algoritma Naïve Bayes*

kappa: 0.626 +/- 0.063 (micro average: 0.626)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	276	43	57	73.40%
pred. Netral	50	376	49	79.16%
pred. Positif	47	55	262	71.98%
class recall	73.99%	79.23%	71.20%	

2. Algoritma *Naïve Bayes* berbasis *Particle Swarm Optimization*

Hasil dari pengujian data dengan menggunakan Algoritma *Naïve Bayes* berbasis *Particle Swarm Optimization* menghasilkan nilai *Accuracy* sebesar 76.29 % dan nilai *kappa* sebesar 0.641. Hasil pengujian dapat dilihat pada tabel 4.7 dan 4.8 berikut ini.

Tabel 4.7 Hasil pengujian data menggunakan *Naïve Bayes* berbasis *Particle Swarm Optimization*

accuracy: 76.29% +/- 5.24% (micro average: 76.30%)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	285	41	53	75.20%

pred. Netral	48	383	56	78.64%
pred. Positif	40	50	259	74.21%
class recall	76.41%	80.80%	70.38%	

Tabel 4.8 Nilai kappa dari pengujian *Algoritma Naïve Bayes berbasis Particle Swarm Optimization*

kappa: 0.641 +/- 0.078 (micro average: 0.641)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	285	41	53	75.20%
pred. Netral	48	383	56	78.64%
pred. Positif	40	50	259	74.21%
class recall	76.41%	80.80%	70.38%	

3. Algoritma Support Vector Machine

Hasil dari pengujian data dengan menggunakan Algoritma *Support Vector Machine* menghasilkan nilai *Accuracy* sebesar 66.51% dan nilai kappa sebesar 0.477. Hasil pengujian dapat dilihat pada tabel 4.9 dan 4.10 berikut ini.

Tabel 4.9 Hasil Pengujian Data Menggunakan *Algoritma Support Vector Machine*.

accuracy: 66.51% +/- 3.13% (micro average: 66.50%)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	285	41	53	75.20%
pred. Netral	48	383	56	78.64%
pred. Positif	40	50	259	74.21%
class recall	76.41%	80.80%	70.38%	

Tabel 4.10 Nilai kappa dari Pengujian Data Menggunakan *Algoritma Support Vector Machine*.

kappa: 0.477 +/- 0.049 (micro average: 0.477)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	285	41	53	75.20%
pred. Netral	48	383	56	78.64%

pred. Positif	40	50	259	74.21%
class recall	76.41%	80.80%	70.38%	

4. Algoritma *Support Vector Machine* berbasis *Particle Swarm Optimization*

Hasil dari pengujian data dengan menggunakan Algoritma *Support Vector Machine* berbasis *Particle Swarm Optimization* menghasilkan nilai *Accuracy* sebesar 68.16 % dan nilai kappa sebesar 0.504. Hasil pengujian dapat dilihat pada tabel 4.11 dan 4.12 berikut ini.

Tabel 4.11 Hasil Pengujian menggunakan *Algoritma Support Vector Machine* berbasis *Particle Swarm Optimization*

accuracy: 66.51% +/- 3.13% (micro average: 66.50%)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	196	14	33	80.66%
pred. Netral	155	448	151	59.42%
pred. Positif	22	12	184	84.40%
class recall	52.55%	94.51%	50.00%	

Tabel 4.12 Nilai Kappa dari Pengujian menggunakan *Algoritma Support Vector Machine* berbasis *Particle Swarm Optimization*

kappa: 0.504 +/- 0.079 (micro average: 0.504)				
	true Negatif	true Netral	true Positif	class precision
pred. Negatif	196	14	33	80.66%
pred. Netral	155	448	151	59.42%
pred. Positif	22	12	184	84.40%
class recall	52.55%	94.51%	50.00%	

4.2 Analisis Evaluasi Hasil dan Validasi Model

Validasi adalah proses untuk mengevaluasi keakuratan prediksi dari model yang diterapkan. Hasil pengujian model yang dilakukan adalah mengklasifikasikan komentar dari Implementasi Kurikulum Merdeka kedalam komentar negative, positif dan netral menggunakan algoritma *Naïve*

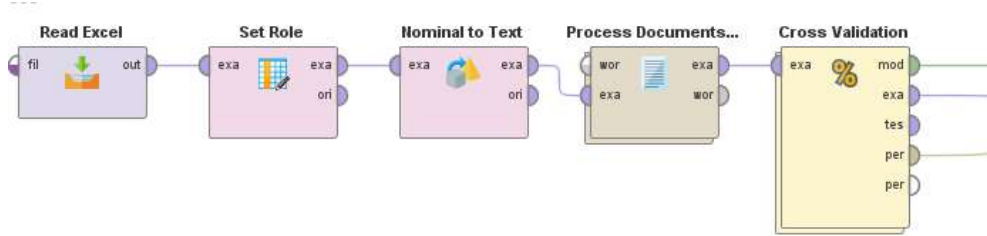
Bayes dan Naïve Bayes berbasis *Particle Swarm Optimization*, *Algoritma Suport Vector Machine* dan *Suport Vector Machine* berbasis *Particle Swarm Optimization* untuk mengetahui nilai akurasi terbaik pada RapidMiner dan desain model berikut ini:

4.2.1 Hasil Pengujian Algoritma Naïve Bayes

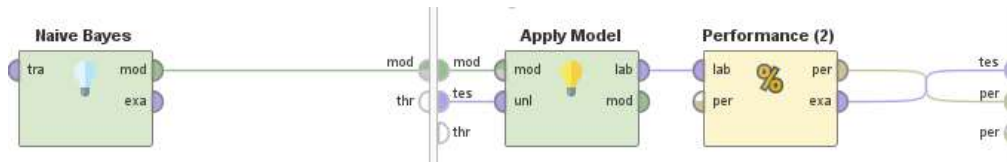
Pengelolaan informasi menggunakan model *Naïve Bayes* dilakukan pada dataset yang terdiri dari 1217 data. Tahapan awal yang dilakukan menggunakan operator *Read Excel* untuk memanggil atau memasukan dataset kedalam model *Naïve Bayes*. Selanjutnya menggunakan operator *Set Role* dimana tahap ini digunakan untuk menetapkan *atribut* label sebagai target role dengan masuk kepengaturan *Chane Role* dan pilih Label.

Operator Nominal to Text digunakan untuk merubah tipe data nominal menjadi teks. Serta menyertakan semua nilai atribut kedalam nilai string yang sesuai. Dalam operator *Process Documents from Data*, Ada beberapa proses yang digunakan untuk membersihkan data menjadi vektor yang dapat digunakan dalam komputasi algoritmik, seperti *Tokenize*, *Transform Case*, *Tokens Filter*, dan *Stopword Filter*. Setelah dokumen menjadi vektor yang dapat dihitung, karena nilai teks berubah menjadi nilai nominal, langkah selanjutnya adalah proses validasi dengan *cross validation*, dengan syarat *K-fold* adalah *K-10*, aplikasi ini memerlukan dua proses disana adalah. Mereka digunakan untuk proses validasi, termasuk proses validasi menggunakan algoritma *Naive Bayes*. *Apply Model* dilakukan agar dapat menerapkan model *Algoritma Naïve Bayes* untuk dilakukan prediksi pada *Confussion Matrix*, tujuannya adalah untuk mendapatkan prediksi pada data yang tidak terlihat atau untuk mengubah data dengan menerapkan model *preprocessing*. *Confussion Matrix* adalah ukuran kinerja untuk masalah klasifikasi pembelajaran mesin yang keluarannya terdiri dari dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 6 kombinasi

nilai prediksi dan aktual yang berbeda. Keseluruhan proses menggunakan Algoritma *Naïve Bayes* dijelaskan pada gambar 4.2 dan 4.3 berikut ini:



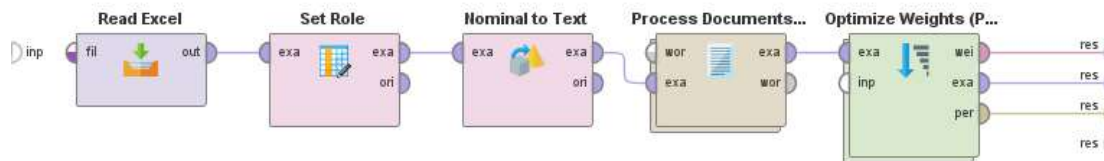
Gambar 4.2 Proses pengujian menggunakan *Algoritma Naïve Bayes*



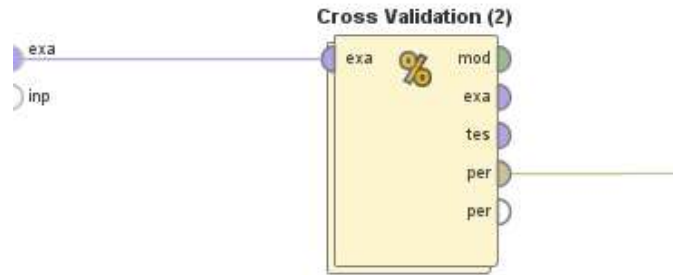
Gambar 4.3 Lanjutan proses pengujian menggunakan *Algoritma Naïve Bayes*

4.2.2 Hasil Pengujian Algoritma *Naïve Bayes* berbasis *PSO*

Particle Swarm Optimization (PSO) adalah salah satu teknik optimasi dan termasuk jenis teknik komputasi evolusi yang dikembangkan oleh *Dr. Eberhart* dan *Dr. Kennedy* pada tahun 1995. PSO digunakan untuk pembobotan atribut pada data set yang penulis gunakan, dimana hanya atribut yang berpengaruh yang digunakan pada proses Algoritma *Naïve Bayes* yang sudah di optimasi selanjutnya. Proses implementasi *Particle Swarm Optimization* dijelaskan pada gambar 4.4 Berikut ini:



Gambar 4.4 Proses pengujian *Algoritma Naïve Bayes* berbasis *Particle Swarm Optimization*



Gambar 4. 5 Proses lanjutan pengujian *Algoritma Naïve Bayes berbasis Particle Swarm Optimization*



Gambar 4. 6 Proses lanjutan pengujian *Algoritma Naïve Bayes berbasis Particle Swarm Optimization*

Tahapan awal yang dilakukan menggunakan operator *Read Excel* untuk memanggil atau memasukkan dataset kedalam model *Naïve Bayes*. Selanjutnya menggunakan operator *Set Role* dimana tahap ini digunakan untuk menetapkan *atribut* label sebagai target role dengan masuk kepengaturan *Chane Role* dan pilih Label.

Operator Nominal to Text digunakan untuk merubah tipe data nominal menjadi teks. Serta menyertakan semua nilai atribut kedalam nilai string yang sesuai. Dalam operator *Process Documents from Data*, terdapat beberapa proses yang digunakan untuk membersihkan data agar menjadi *vector* yang dapat digunakan untuk perhitungan algoritma diantaranya *Tokenize*, *Transform Case*, *Filter Tokens*, *Filter Stopword*. Setelah dokumen di *Particle Swarm Optimization* maka selanjutnya adalah proses validasi menggunakan *Cross Validation*, dengan ketentuan K-fold sebesar K-10, dalam penerapan ini terdapat 2 proses yang digunakan untuk proses validasi diantaranya adalah proses validasi menggunakan algoritma *Naive Bayes*.

Apply Model dilakukan agar dapat menerapkan model *Algoritma Naïve Bayes* untuk dilakukan prediksi pada *Confussion Matrix*, tujuannya adalah untuk mendapatkan prediksi pada data yang tidak terlihat atau untuk mengubah data dengan menerapkan model preprocessing.

Confussion Matrix adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 6 kombinasi berbeda dari nilai prediksi dan nilai actual.

4.2.3 Hasil Pengujian Algoritma *Support Vector Machine*

Pengelolaan informasi menggunakan model *Support Vector Machine* dilakukan pada dataset yang terdiri dari 1217 data. Tahapan awal yang dilakukan menggunakan operator *Read Excel* untuk memanggil atau memasukan dataset kedalam model *Naïve Bayes*. Selanjutnya menggunakan operator *Set Role* dimana tahap ini digunakan untuk menetapkan *atribut* label sebagai target role dengan masuk kepengaturan *Change Role* dan pilih Label.

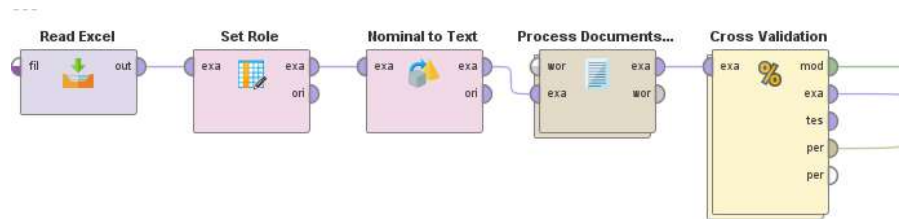
Operator Nominal to Text digunakan untuk merubah tipe data nominal menjadi teks. Serta menyertakan semua nilai atribut kedalam nilai string yang sesuai.

Dalam operator *Process Documents from Data*, terdapat beberapa proses yang digunakan untuk membersihkan data agar menjadi *vector* yang dapat digunakan untuk perhitungan algoritma diantaranya *Tokenize*, *Transform Case*, *Filter Tokens*, *Filter Stopword*.

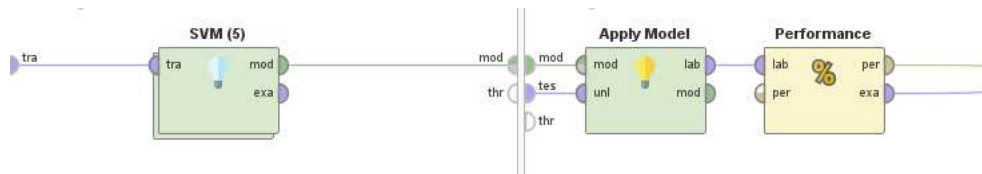
Setelah dokumen menjadi vector yang dapat dihitung karena nilai text tersebut berubah jadi nominalmaka selanjutnya adalah proses validasi menggunakan *Cross Validation*, dengan ketentuan K-fold sebesar K-10, dalam penerapan ini terdapat 2 proses yang digunakan untuk proses validasi diantaranya adalah proses validasi menggunakan algoritma *Support Vector Machine*.

Apply Model dilakukan agar dapat menerapkan model *Algoritma Suport Vector Machine* untuk dilakukan prediksi pada *Confussion Matrix*, tujuannya adalah untuk mendapatkan prediksi pada data yang tidak terlihat atau untuk mengubah data dengan menerapkan model *preprocessing*.

Confussion Matrix adalah pengukuran performa untuk masalah klasifikasi *Machine Learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confussion Matrix* adalah tabel dengan 6 kombinasi berbeda dari nilai prediksi dan nilai aktual. Keseluruhan proses menggunakan *Algoritma Suport Vector Machine* dijelaskan pada gambar 4.7 dan 4.8 berikut ini:



Gambar 4.7 Pengujian *Algoritma Suport Vector Machine*

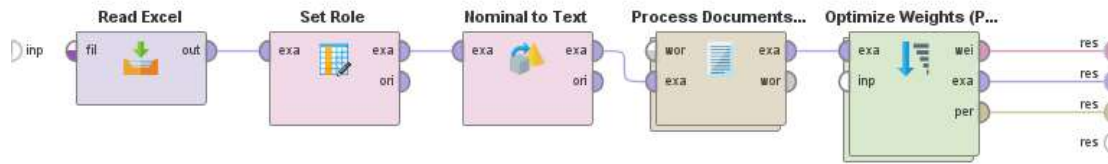


Gambar 4.8 Lanjutan pengujian *Algoritma Suport Vector Machine*

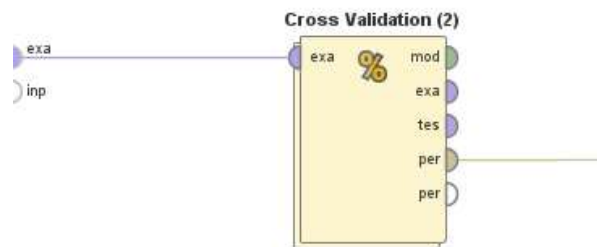
4.2.4 Hasil Pengujian *Algoritma Suport Vector Machine* berbasis PSO

Particle Swarm Optimization (PSO) adalah salah satu teknik optimasi dan termasuk jenis teknik komputasi evolusi yang dikembangkan oleh *Dr. Eberhart* dan *Dr. Kennedy* pada tahun 1995.

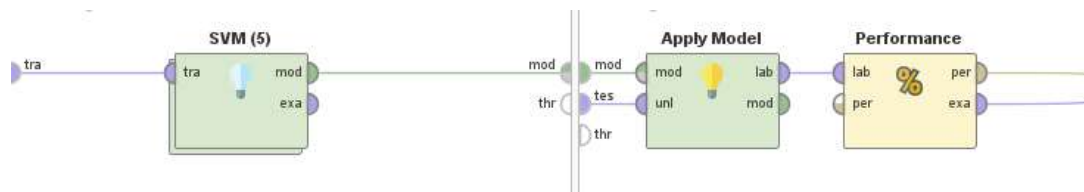
Particle Swarm Optimization (PSO) digunakan untuk pembobotan atribut pada data set yang penulis gunakan, dimana hanya atribut yang berpengaruh yang digunakan pada proses *Algoritma Suport Vector Machine* yang sudah di optimasi selanjutnya. Proses implementasi *Particle Swarm Optimization (PSO)* dijelaskan pada gambar 4.9 sampai 4.11 Berikut ini:



Gambar 4.9 Proses pengujian *Algoritma Support Vector Machine* berbasis *Particle Swarm Optimization (PSO)*



Gambar 4.10 Proses lanjutan pengujian *Algoritma Support Vector Machine* berbasis *Particle Swarm Optimization (PSO)*



Gambar 4.11 Proses lanjutan pengujian *Algoritma Support Vector Machine* berbasis *Particle Swarm Optimization (PSO)*

4.3 Analisis Evaluasi Hasil dan Validasi Model

Pada tahap ini penulis melakukan komparasi terhadap hasil penelitian percobaan 2 metode yang digunakan yaitu *Decision Tree C4.5* dan *Support Vector Machine (SVM)* yang masing-masing tahapannya dilakukan dengan menggunakan seleksi fitur *Particle Swarm Optimization (PSO)*.

Penulis menyajikan hasil komparasi melalui table 4.13 berikut:

Tabel 4.13 Tabel Hasil Komparasi

Metode	Nilai Accuracy	Nilai Kappa
<i>Naïve Bayes</i>	75.24%	0.626
<i>Support Vector Machine</i>	66.51%	0.477
<i>Naïve Bayes + PSO</i>	76.29%	0.641
<i>Support Vector Machine + PSO</i>	68.16%	0.504

4.3.1 Analisis Hasil

Pada tahap ini penulis melakukan evaluasi hasil dari penelitian yang telah dilakukan, untuk mengetahui hasil dari percobaan menggunakan 2 metode klasifikasi untuk menyelesaikan permasalahan penelitian yaitu, manakah metode terbaik antara *Naïve Bayes* dan Support Vector Machine (SVM) dalam Analisis Sentiment Implementasi Kurikulum Merdeka Berbasis *Particel Sward Optimazion (PSO)*. Hasil menunjukkan perbedaan yang cukup signifikan jika metode klasifikasi menggunakan seleksi fitur *Particel Sward Optimazion (PSO)*. Jika dilihat dari hasil komparasi (table 4.13), percobaan yang dilakukan pada *dataset* yang diolah menggunakan *Naïve Bayes* tanpa menggunakan seleksi fitur didapatkan nilai akurasi 75.24% dan nilai kappa sebesar 0.626, pada metode *Support Vector Machine (SVM)* tanpa menggunakan seleksi fitur didapatkan nilai akurasi 66.51% dan nilai kappa sebesar 0.477.

Selanjutnya jika kedua metode tersebut di optimasi dengan menggunakan *Particel Sward Optimazion (PSO)* maka akan menunjukkan hasil peningkatan dimana metode *Naïve Bayes* dengan fitur *Particel Sward Optimazion (PSO)* mendapatkan nilai akurasi 76.29, hasil nilai akurasi yang didapat meningkat sebesar 1% dan nilai kappa juga meningkat menjadi 0.641. Untuk metode *Support Vector Machine (SVM)* dengan *Particel Sward Optimazion (PSO)* mendapatkan peningkatan nilai akurasi yang sangat baik dimana metode ini mendapatkan nilai akurasi 68.16%, hasil nilai akurasi yang didapat meningkat sebesar 2% dan nilai kappa sebesar 0.504. Dengan demikian Metode *Naïve Bayes* berbasis *Particel Sward Optimazion (PSO)* memiliki performa yang lebih baik jika dibandingkan dengan Algoritma *Support Vector Machine* berbasis *Particle Swarm*, dengan hasil yang diperoleh Algoritma *Naïve Bayes* pada penelita ini dengan *accuracy* mencapai 76.29%.

Besar nilai akurasi dipengaruhi oleh beberapa faktor diantaranya adalah jumlah data latih dan data uji, semakin banyak data latih yang digunakan semakin baik model dapat mempelajari pola dalam data dan dapat meningkatkan nilai akurasi. Selain itu juga kejelasan dan keterkaitan data latih juga dapat mempengaruhi nilai akurasi pada algoritma yang digunakan. Kemudian kelas distribusi yang tidak seimbang juga dapat mempengaruhi performa algoritma yang berdampak pada nilai akurasinya.

Performance nilai kappa [21] dapat dikalsifikasikan menjadi lima kelompok yaitu:

Tabel 4.14 Klasifikasi nilai kappa

No	Nilai Kappa	Klasifikasi
1	0.81 – 1.00	Very Good
2	0.61 – 0.80	Good
3	0.41 – 0.60	Moderate
4	0.21 – 0.40	Fair
5	0.00 – 0.20	Poor

Jika dilihat dari tabel 4.11 maka Algoritma *Naïve Bayes* termasuk algoritma yang Baik jika digunakan untuk analisis sentiment dengan nilai kappa yang dihasilkan algoritma *Naïve Bayes* yaitu sebesar 0.626, sedangkan untuk algoritma *Support Vector Machine* termasuk kedalam kategori yang Sedang jika digunakan untuk analisis sentiment dengan nilai kappa yang dihasilkan yaitu 0.41. Kemudian untuk algoritma *Naïve Bayes* berbasis *Particel Sward Optimazion (PSO)* juga termasuk kedalam kategori Baik dengan nilai kappa yang dihasilkan sebesar 0.641, dan algoritma *Support Vector Machine* berbasis *Particel Sward Optimazion (PSO)*,

Berdasarkan klasifikasi diatas maka dapat disimpulkan bahwa Algoritma *Naïve Bayes* berbasis *Particel Sward Optimazion (PSO)* merupakan algoritma yang Baik dan bisa digunakan untuk melakukan analisis sentiment.