

## BAB IV

### HASIL DAN PEMBAHASAN

Pada tahap ini bertujuan untuk mengimplemantasikan *Particel Sward Optimazion* dalam tingkatan akurasi prediksi penyakit stroke dengan memakai metode *Algoritma C4. 5*. dan *Support Vector Machine (SVM)*, dengan menggunakan bantuan tool *RapidMiner Studio 9.10*. Hasil dari riset ini sendiri berbentuk hasil pengolahan informasi kualitatif serta kuantitatif dengan perhitungan bersumber pada model yang diusulkan dan disajikan menggunakan *Confusion Matrik*. Riset dicoba pada suatu dataset yang telah ada yang bersifat publik yang bersumber pada *kaggle.com*. Eksperimen serta pengujian dalam riset ini dilakukan dengan menggunakan Metode *Algoritma C4. 5* dan *Support Vector Machine (SVM)* tanpa *Particel Sward Optimazion* dan dengan menggunakan *Particel Sward Optimazion*.

#### 4.1. Pengujian Algoritma C4.5

##### 4.1.1. Read Dataset

Tahapan ini dilakukan pada Rapid Miner dengan menggunakan *Operator Retrive* dan mengexport dataset yang sudah didapatkan melalui *Kaggle.com*. Tahapan ini berfungsi sebagai pemanggilan dan pembacaan data yang akan di olah, Pada tahap *Result* kita juga dapat mengetahui ad atau tidaknya *Missing Value* pada data set yang kita gunakan. Dari 5110 data yang penulis gunakan tidak ada *Missing Value* dari dataset yang digunakan seperti yang terlihat pada gambar 4.1.

Row No.	id	gender	age	hype...	heart_disea...	ever_married	work_type	Residence_L...	avg_glucos...	bmi	smoking_status	stroke ↓
1	9046	Male	67	0	1	Yes	Private	Urban	228.690	36.6	formerly smoked	1
2	51676	Female	61	0	0	Yes	Self-employed	Rural	202.210	N/A	never smoked	1
3	31112	Male	80	0	1	Yes	Private	Rural	105.920	32.5	never smoked	1
4	60182	Female	49	0	0	Yes	Private	Urban	171.230	34.4	smokes	1
5	1665	Female	79	1	0	Yes	Self-employed	Rural	174.120	24	never smoked	1
6	56669	Male	81	0	0	Yes	Private	Urban	186.210	29	formerly smoked	1
7	53882	Male	74	1	1	Yes	Private	Rural	70.090	27.4	never smoked	1
8	10434	Female	69	0	0	No	Private	Urban	94.390	22.8	never smoked	1
9	27419	Female	59	0	0	Yes	Private	Rural	76.150	N/A	Unknown	1
10	60491	Female	78	0	0	Yes	Private	Urban	58.570	24.2	Unknown	1
11	12109	Female	81	1	0	Yes	Private	Rural	80.430	29.7	never smoked	1
12	12095	Female	61	0	1	Yes	Govt_job	Rural	120.460	36.8	smokes	1
13	12175	Female	54	0	0	Yes	Private	Urban	104.510	27.3	smokes	1
14	8213	Male	78	0	1	Yes	Private	Urban	219.840	N/A	Unknown	1

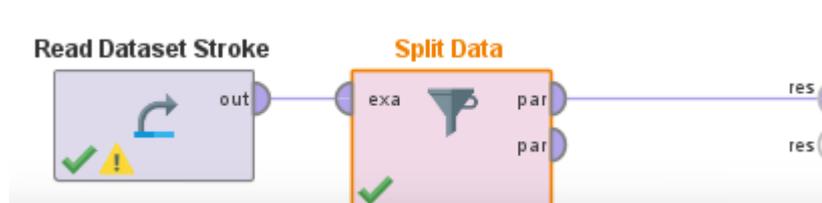
ExampleSet (5,110 examples, 0 special attributes, 12 regular attributes)

**Gambar 4.1** Dataset yang digunakan

### 4.1.2. Implementasi Algoritma C4.5

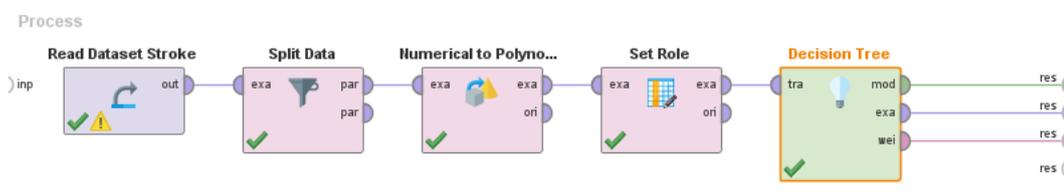
Pengelolaan informasi memakai model C4. 5 dilakukan pada dataset yang terdiri dari 1150 data dan 12 atribut dengan format *CSV* yang dimana atribut dari prakiraan penderita stroke dengan tabel terakhir sebagai nilai class (stroke)

Model dari *Algoritma C4. 5* berbentuk pohon keputusan, supaya lebih mempermudah dalam membuat pohon keputusan, langkah awal merupakan menghitung jumlah *class* yang berpotensi terserang penyakit stroke serta tidak stroke, dari tiap- tiap class bersumber pada atribut yang sudah ditetapkan dengan melakukan *split data training* dan *testing* pada tahap ini penulis menggunakan 0.7 (70%) *data training* dan 0.3 (30%) *data testing*.



Gambar 4.2. Proses Split data

Setelah dilakukannya split data terjadi pengurangan jumlah data menjadi 3577 data, kemudian dilanjutkan dengan tahapan selanjutnya yaitu mengubah *type data label class* (stroke) menjadi nominal, untuk selanjutnya diolah dengan *Decision Tree*.



Gambar 4.3. Proses Decision Tree

*Operator Numerical to Polynominal* digunakan karena *label class* data awal masih memiliki *type interger*, untuk dirubah ke *type nominal* karena pengolahan *Decision Tree* harus menggunakan *type label class* nominal.

*Operator Set Role* tahap ini menetapkan *atribut name stroke* sebagai target *role* dengan mengubah menjadi *label*.

*Decission Tree* kumpulan node seperti pohon yang dimaksudkan untuk membuat keputusan tentang afiliasi nilai ke kelas atau perkiraan nilai target numerik. Setiap node mewakili aturan pemisahan untuk satu Atribut tertentu.

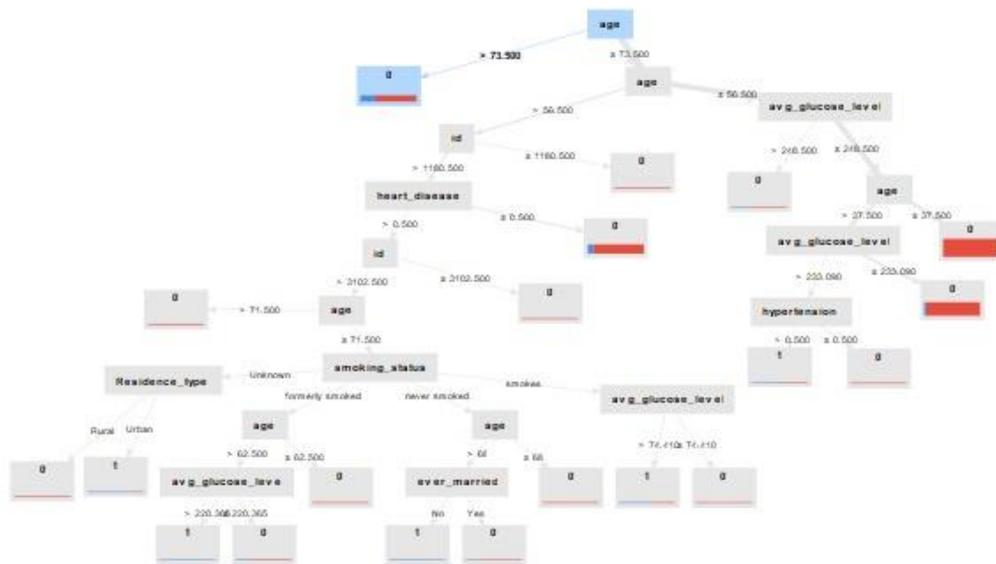
#### **4.1.3. Hasil Implementasi Metode *Decission Tree* Pada Tahap *Gain Ratio*.**

Hasil Pohon keputusan berikut merupakan hasil dari implementasi *Algortima Decission Tree* pada pemrosesan *Gain Ratio*. Sebelum dilakukan pemodelan pohon keputusan dilakukan pencarian nilai *Gain* Atribut, dimana atribut yang memiliki nilai tertinggi akan menjadi akar pertama.

attribute	weight
smoking_status	0.008
Residence_type	0.203
id	0.031
heart_disease	0.006
avg_glucose_level	0.255
hypertension	0.114
age	0.225
ever_married	0.158

**Gambar 4.4** Bobot atribut *Gain ratio*

Terlihat dari 12 atribut menjadi 8 atribut berpengaruh, pada atribut *avg\_glucose\_level* (kadar gula darah) memiliki nilai yang sama dengan atribut *age* (umur) yaitu dengan nilai 0.255 dengan demikian kedua atribut ini menjadi akar pertama pada pohon keputusan.



Gambar 4.5. Bentuk Pohon Keputusan *Gain Ratio*

#### 4.1.4. Hasil Implementasi Metode *Decision Tree* Pada Tahap *Information Gain*

Hasil Pohon keputusan berikut merupakan hasil dari implementasi *Algoritma Decision Tree* pada pemrosesan *Information Gain*. Sebelum dilakukan pemodelan pohon keputusan dilakukan pencarian nilai *Gain* Atribut, dimana atribut yang memiliki nilai tertinggi akan menjadi akar pertama.

attribute	weight
smoking_status	0.082
gender	0.302
work_type	0.122
Residence_type	0.356
hypertension	0.107
heart_disease	0.026
ever_married	0.004

Gambar 4.6. Bobot atribut *Information Gain*



attribute	weight
smoking_status	0.008
gender	0.235
work_type	0.071
Residence_type	0.423
heart_disease	0.002
hypertension	0.112
ever_married	0.148

Gambar 4.8. Bobot Atribut *Gini Index*

Terlihat dari 12 atribut menjadi 10 atribut berpengaruh, pada atribut *id* (*id* memiliki nilai yang sama dengan atribut *avg\_glucose\_level* (kadar gula dalam darah) yaitu dengan nilai 0.255 dan 0.296 dengan demikian kedua atribut ini menjadi akar pertama pada pohon keputusan.



Gambar 4.9. Pohon Keputusan *Gini Index*

Bobot atribut pada proses *gain ratio*, *information gain*, dan *gini index* berbeda-beda karena metode-metode tersebut memiliki perbedaan dalam cara mereka mengukur kualitas pemisahan antara kelas-kelas yang berbeda pada suatu atribut.

*Information gain*, misalnya, menggunakan pengukuran reduksi entropi (ketidakpastian) pada data setelah dipisahkan dengan suatu atribut. Semakin banyak informasi yang dapat dipisahkan atau dihilangkan, maka semakin besar pula nilai *information gain* dari atribut tersebut.

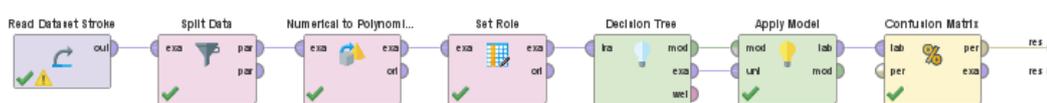
*Gini index*, di sisi lain, mengukur seberapa baik sebuah atribut memisahkan dua kelas dengan menghitung *impurity* dari setiap kelas pada atribut tersebut. Semakin kecil *impurity*-nya, maka semakin baik pula atribut tersebut dalam memisahkan kelas-kelas.

*Gain ratio* mengambil pendekatan yang lebih kompleks dengan memperhitungkan informasi yang terkandung dalam atribut serta kompleksitas atribut tersebut. Dengan kata lain, *gain ratio* tidak hanya memperhatikan seberapa baik atribut dapat memisahkan kelas-kelas, namun juga mempertimbangkan seberapa banyak kemungkinan nilai atribut tersebut dapat dibagi.

Karena setiap metode tersebut memiliki pendekatan yang berbeda untuk mengukur kualitas pemisahan atribut, maka bobot atribut pada masing-masing metode pun akan berbeda-beda.

#### 4.1.6. Confusion Matrix Pada Algoritma C4.5

Tahap ini dilakukan penyajian informasi tentang *Accuracy* dan nilai *AUC (Area Under Curve)* dari pemrosesan data set stroke menggunakan *Algoritma C4.5* yang sudah dilakukan pada tahap sebelumnya dengan disajikan melalui *Confusion Matrix*.



**Gambar 4.10.** Tahapan Penyajian *Confusion Matrix*

*Apply Model* dilakukan agar dapat menerapkan model *Algoritma C4.5* untuk dilakukan prediksi pada *Confusion Matrix*, bertujuan untuk mendapatkan prediksi pada data atau untuk mengubah data dengan menerapkan model *preprocessing*.

*Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai actual.

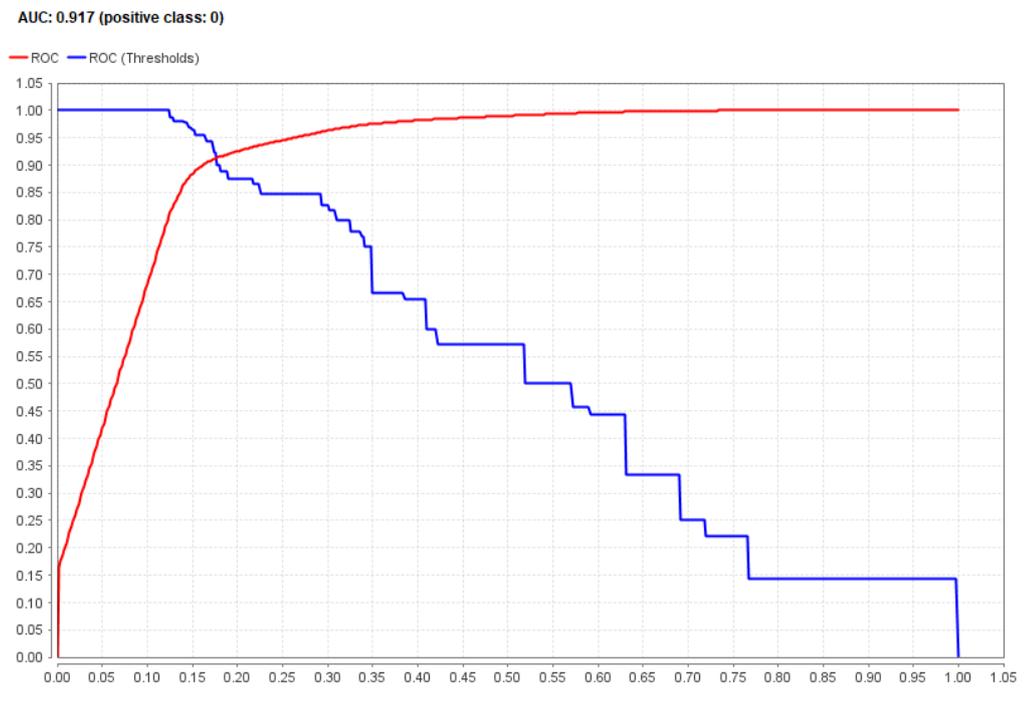
**Tabel 4.1.** Tabel Hasil *Confusion Matrix C4.5*

---

***Accuracy: 95,58%***  
***Precision: 96.89%***  
***Recall: 98.41%***

	True 1	True 0	Class precision
Pred.1	144	53	73.10%
Pred.0	105	3275	96.89%
Class recall	57.83%	98.41%	

---



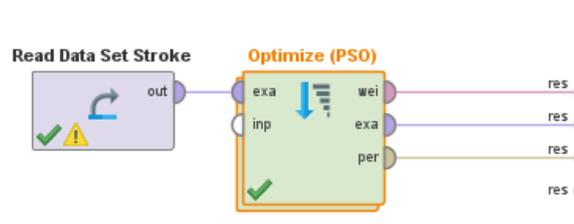
**Gambar 4.11.** Nilai Diagram AUC

Berdasarkan tabel Confusion Matrix yang didapatkan dengan melakukan pemodelan pemrosesan dataset dengan metode *Algoritma C4.5* dengan menggunakan data set stroke didapatkan nilai *accuracy* 95,58%, *Precision* 96.89%, dan *Recall* 98.41%, dengan nilai *AUC (Area Under Curve)* 0,917.

## 4.2. Pengujian Algoritma C4.5 Dengan Particel Sward Optimazion

### 4.2.1. Particle Swarm Optimization (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 seleksi atribut/menghilangkan atribut pada data set yang penulis gunakan, dimana hanya atribut yang berpengaruh yang digunakan pada proses *Algoritma C4.5* yang sudah di optimasi selanjutnya.



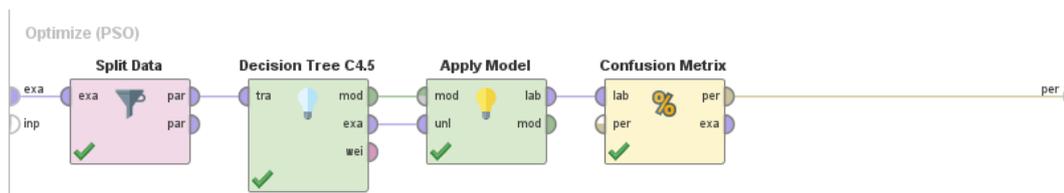
**Gambar 4.12.** Tahap Optimasi dataset dengan PSO

attribute	weight ↓
id	1
age	0.596
ever_married	0.462
smoking_status	0.452
bmi	0.342
gender	0.225
Residence_type	0.061
work_type	0.036
avg_glucose_level	0.020
hypertension	0
heart_disease	0

**Gambar 4.13** Atribut yang sudah di PSO

Dari 11 atribut dengan 1 lebel class dari data set sebelum di optimasi dengan PSO, tersisa 9 atribut yang memiliki bobot nilai di atas 0 yaitu *Id* 1, *Age* 0.596, *ever\_married* 0.462, *smokingstatus* 0.0452, *bmi* 0.342, *gender* 0.225, *Residenttype* 0.061, *work\_type* 0.036, *avg\_glucoselvl* 0.020. Dengan sisa atribut memiliki nilai 0.

Selanjutnya data set yang sudah di optimasi menggunakan *Particel Sward Optimazion*, di olah menggunakan *Algoritma Decision Tree C4.5*. Seperti pada gambar pemrosesan berikut.



**Gambar 4.14.** Tahap Pemrosesan dataset PSO dengan C4.5

Setelah data set di optimasi menggunakan *Particel Sward Optimazion*, dilakukan proses *Split Data* yaitu membagi data *testing* dan *training* dengan perbandingan 70:30, dengan data traning 0.7(70%) dan data testing 0.3(30%).

Kemudian diproses dengan menggunakan *Algoritma Decision Tree C4.5*. dimana penelitian ini menggunakan perhitungan *Gain Ratio* dengan mengatur kedalaman pohon sebanyak 10, dengan mengaktifkan *apply pruning* dengan tujuan (alpaydin 2010) memangkas cabang agar tak terlalu rumit dan besar agar dihasilkan diagram pohon yang lebih akurat dan simple, dan *apply prepruning* dimana proses ini menghentikan proses pembuatan cabang pada titik tertentu. semakin besar cabang yang diulang semakin besar kompleksitas dari pohon keputusan namun jika terlalu sedikit makan pohon keputusan juga kurang akurat. Salah satu faktor [24] yang dapat menurunkan akurasi adalah proses pruning cabang (*node*) dengan nilai informasi yang tinggi.

*Apply Model* dilakukan agar dapat menerapkan model *Algoritma C4.5* untuk dilakukan prediksi pada *Confussion Matrix*, bertujuan untuk mendapatkan prediksi pada data 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 4 kombinasi berbeda dari nilai prediksi dan nilai actual.

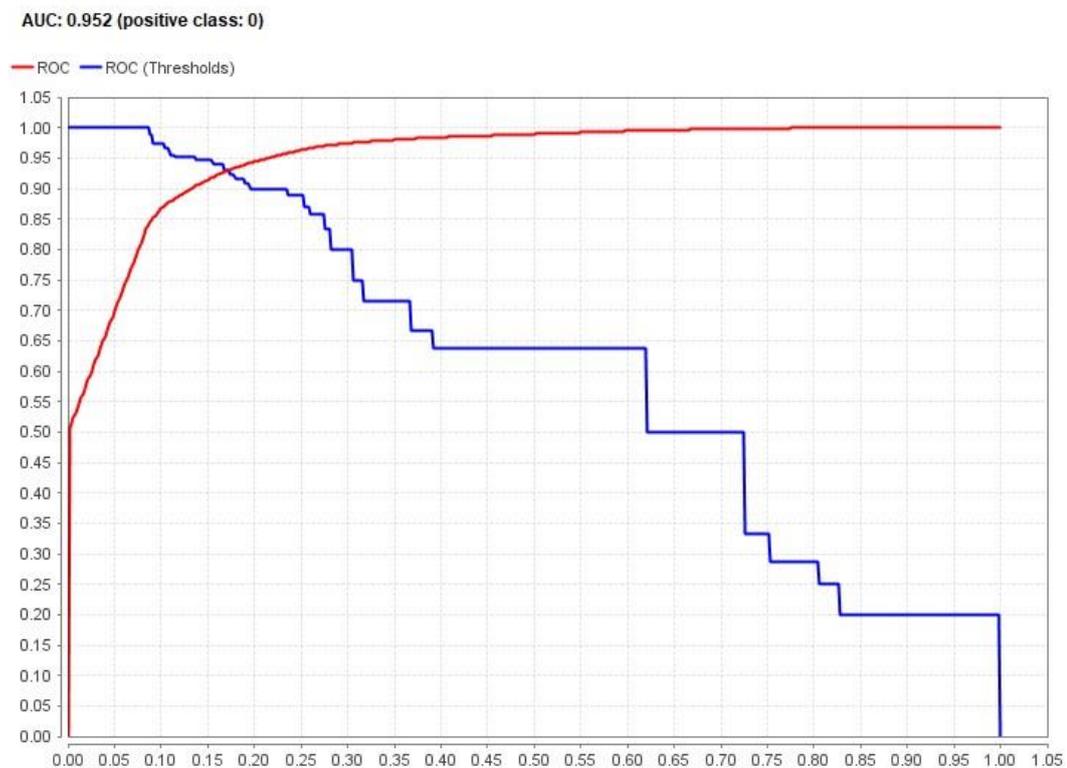
**Tabel 4.2.** Tabel hasil Confusion Matrix C4.5 dengan PSO

**Accuracy: 96.56%**

**Precision: 98.01%**

**Recall: 98.38%**

	True 1	True 0	Class precision
Pred.1	106	55	65.84%
Pred.0	68	3348	98.01%
Class recall	60.92%	98.38%	



**Gambar 4.15** Nilai AUC Decission Tree C4.5 dengan PSO

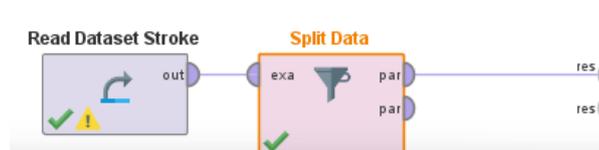
Berdasarkan tabel Confusion Matrix yang didapatkan dengan melakukan pemodelan pemrosesan dataset dengan metode *Algoritma C4.5* yang di optimasi menggunakan *Particel Sward Optimazion (PSO)* dengan menggunakan data set stroke didapatkan nilai *accuracy* 96,56%, *Precision* 98.01%, dan *Recall* 98.38%, dengan nilai *AUC (Area Under Curve)* 0,952.

### 4.3. Pengujian Algoritma Support Vector Machine (SVM)

#### 4.3.1. Memasukan dan Proses Splitdata Pada SVM

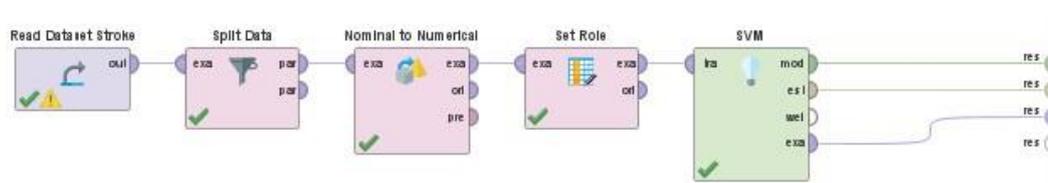
Pengelolaan informasi memakai model *Support Vector Machine (SVM)* dilakukan pada dataset yang terdiri dari 1150 data dan 12 atribut dengan format *CSV* yang dimana atribut dari prakiraan penderita stroke dengan tabel terakhir sebagai nilai class (stroke)

Tahapan awal yang dilakukan dengan menggunakan *Operator Rtrieve* untuk memanggil/membaca *dataset* melakukan *split data training* dan *testing* pada tahap ini penulis menggunakan 0.7 (70%) *data training* dan 0.3 (30%) *data testing* sehingga dihasilkan sisa data sebagai *data testing* dengan jumlah 3,577 data.



**Gambar 4.16.** Proses Split Data

Setelah dilakukannya split data terjadi pengurangan jumlah data menjadi 3.577 data, kemudian dilanjutkan dengan tahapan selanjutnya yaitu mengubah *type* data *label class* (stroke) menjadi nominal, untuk selanjutnya diolah dengan *Support Vector Machine*.



**Gambar 4.17.** Pemrosesan *Support Vector Machine (SVM)*

*Operator Nominal to Numerical* digunakan untuk mengubah tipe atribut non-numerik menjadi tipe numerik. *Operator* ini tidak hanya mengubah jenis atribut yang dipilih tetapi juga memetakan semua nilai atribut tersebut ke nilai numerik. Nilai atribut biner dipetakan ke 0 dan 1, pada tahap ini mengubah atribut *Nominal* pada dataset menjadi *Numerical*

*Operator Set Role* tahap ini menetapkan *atribut name stroke* sebagai *target role* dengan mengubah menjadi *label \*.

#### 4.3.2. Hasil Dataset dan Plot Daigram yang Sudah diklasifikasi dengan SVM

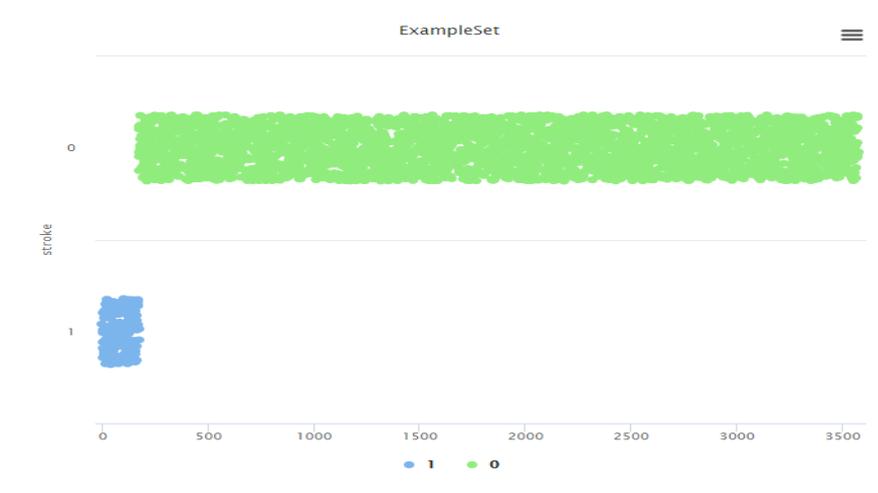
*Support Vector Machine (SVM)* mengambil satu set data input dan memprediksi, untuk setiap input yang diberikan, Model SVM adalah representasi dari contoh-contoh sebagai titik-titik dalam ruang, dipetakan sedemikian rupa sehingga contoh-contoh dari kategori-kategori terpisah dipisahkan oleh celah yang jelas seluas mungkin. Contoh-contoh baru kemudian dipetakan ke dalam ruang yang sama dan diprediksi termasuk dalam kategori berdasarkan di sisi celah mana mereka berada berbagai jenis kernel yang ada pada SVM termasuk *dot*, *radial*, *polynomial*, *neural*, *anova*, *epachnenikov*, *kombinasi gaussian* dan *multiquadric*. Pada tahapan ini digunakan karnel *Polynominal* sebagai parameter uji uang dilakukan, dalam machine learning, *kernel polynomial* adalah fungsi *kernel* yang cocok untuk digunakan dalam SVM dan kernelisasi lainnya, di mana kernel mewakili kesamaan vektor sampel pelatihan dalam ruang fitur.

Row No.	stroke	gender = Male	gender = Fe...	gender = Ot...	age = 67	age = 61	age = 80	age = 49
1	1	1	0	0	1	0	0	0
2	1	0	1	0	0	1	0	0
3	1	0	1	0	0	0	0	1
4	1	1	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0
6	1	0	1	0	0	0	0	0
7	1	0	1	0	0	0	0	0
8	1	0	1	0	0	1	0	0
9	1	1	0	0	0	0	0	0
10	1	0	1	0	0	0	0	0
11	1	1	0	0	0	0	0	0
12	1	1	0	0	0	0	0	0
13	1	0	1	0	0	0	0	0
14	1	1	0	0	0	0	0	0

ExampleSet (3,577 examples, 1 special attribute, 4,523 regular attributes)

**Gambar 4.18.** Hasil Klasifikasi Dataset Menggunakan SVM

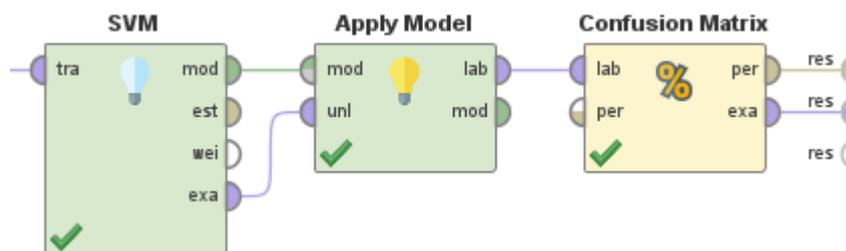
Perubahan pada atribut data set yang menggunakan nilai *Numerical* karena Algoritma *Support Vector Machine (SVM)* hanya dapat memproses tipe data *Numerical* untuk dilakukan penyebaran *plot* (titik-titik) data untuk dibagi kalsifikasi berdasarkan sebaran data yang di proses oleh model SVM.



**Gambar 4.19** Diagram Plot SVM

Diagram tersebut merupakan hasil sebaran plot dataset yang sudah di klasifikasi menggunakan *Support Vector Machine* dengan jumlah data yang sudah dilakukan *Splitdata* menjadi 3.577. Dengan nilai 1 sebagai terindikasi menderita Stroke dan nilai 0 terklasifikasi tidak stroke.

### 4.3.3. Proses Confusion Matrik Pada Support Vector Machine



**Gambar 4.20.** Pemrosesan Confusion Matrix

*Apply Model* dilakukan agar dapat menerapkan model *Support Vector Machine* untuk dilakukan prediksi pada *Confusion Matrix*, bertujuan untuk mendapatkan prediksi pada data atau untuk mengubah data dengan menerapkan model *preprocessing*.

*Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai actual.

Row No.	stroke	prediction(s...	confidence(1)	confidence(0)	gender = Male	gender = Fe...	gender = Ot...	age = 67
1	1	1	0.560	0.440	1	0	0	1
2	1	1	0.512	0.488	0	1	0	0
3	1	1	0.544	0.456	0	1	0	0
4	1	1	0.553	0.447	1	0	0	0
5	1	1	0.551	0.449	1	0	0	0
6	1	1	0.513	0.487	0	1	0	0
7	1	0	0.299	0.701	0	1	0	0
8	1	0	0.325	0.675	0	1	0	0
9	1	1	0.513	0.487	1	0	0	0
10	1	1	0.531	0.469	0	1	0	0
11	1	1	0.583	0.417	1	0	0	0
12	1	1	0.546	0.454	1	0	0	0
13	1	1	0.579	0.421	0	1	0	0
14	1	1	0.514	0.486	1	0	0	0

ExampleSet (3,577 examples, 4 special attributes, 4,523 regular attributes)

**Gambar 4.21.** Tabel Hasil Prediksi SVM pada *Confusion Matrix*

Pada table di atas ditunjukkan 2 nilai *class* pada dataset (hijau) muncul variable baru dengan nama *prediction stroke* dimana table tersebut menunjukkan label awal dataset dengan hasil yang prediksi dengan *support vector machine*. Tabel ketepatan/nilai keyakinan (kuning) merupakan nilai kepercayaan dari hasil prediksi yang dilakukan *Support Vector Machine*.

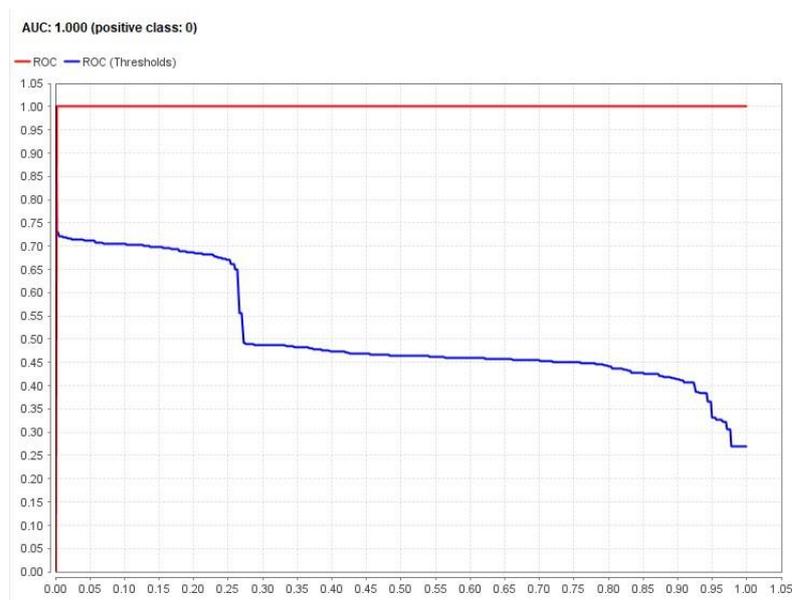
**Tabel 4.3.** Tabel Hasil *Confusion Matrix Support Vector Machine*

***Accuracy:98.71%***

***Precision:98.67%***

***Recall:100.00%***

	True 1	True 0	Class precision
Pred.1	128	0	100.00%
Pred.0	46	3403	98.67%
Class recall	73.56%	100.00%	

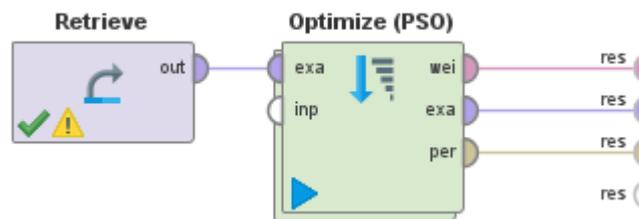


**Gambar 4.22.** Grafik Nilai AUC

Berdasarkan tabel *Confusion Matrix* yang didapatkan dengan melakukan pemodelan pemrosesan dataset dengan metode *Support Vector Machine*, dengan menggunakan dataset stroke didapatkan nilai *accuracy* 98,71%, *Precision* 98.67%, dan *Recall* 100.00%, dengan nilai *AUC* (*Area Under Curve*) 1,000.

#### 4.4. Pengujian Algoritma Support Vector Machine Dengan Particel Sward Optimazion

Pada pengujian ini dilakukan pengujian algoritma SVM dengan melakukan seleksi fitur pada *dataset*, dengan tujuan mengurangi nilai variable agar menghasilkan variable yang dikira nilainya lebih optimal untuk melakukan deteksi penyakit stroke.

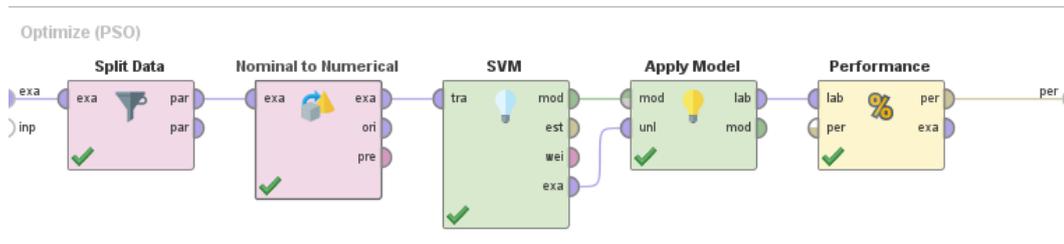


**Gambar 4.23.** Proses Seleksi Atribut Pada SVM

attribute	weight ↓
gender	1
Residence_type	1
hypertension	0.914
heart_disease	0.722
ever_married	0.702
id	0.348
smoking_status	0.317
age	0.159
avg_glucose_level	0.103
bmi	0.043
work_type	0

**Gambar 4.24.** Nilai bobot atribut yang di optimasi dengan PSO

Dari 11 atribut dengan 1 lebel class dari data set sebelum di optimasi dengan PSO, tersisa 10 atribut yang memiliki bobot nilai di atas 0 yaitu *Gender* 1, *Residence\_type* 1, *Hypertension* 0.914, *heart\_disease* 0.722, *ever\_meried* 0.702, *id* 0.348, *smoking\_stats* 0.317, *age* 0.159, *avg\_glucose\_lvl* 0.103, *bmi*0.043 dengan sisa atribut *work\_type* memiliki nilai 0.



**Gambar 4.25.** Proses optimasi data set PSO dan Confusion Matrix

*Apply Model* dilakukan agar dapat menerapkan model *Support Vector Machine* untuk dilakukan prediksi pada *Confusion Matrix*, bertujuan untuk mendapatkan prediksi pada data atau untuk mengubah data dengan menerapkan model *preprocessing*.

*Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai actual.

**Tabel 4.4** Tabel Hasil Confusion Matrix SVM dengan PSO

**Accuracy: 99.11%**

**Precision: 99.07%**

**Recall: 100.00%**

	True 1	True 0	Class precision
Pred.1	142	0	100.00%
Pred.0	32	3403	99.07%
Class recall	81.61%	100.00%	



**Gambar 4.26.** Grafik Nilai AUC

Berdasarkan tabel *Confusion Matrix* yang didapatkan dengan melakukan pemodelan pemrosesan dataset dengan metode *Support Vector Machine (SVM)* yang di optimasi menggunakan *Particel Sward Optimazion (PSO)* dengan menggunakan data set stroke didapatkan nilai *accuracy* 99.11%, *Precision* 99.07%, dan *Recall* 100.00%, dengan nilai *AUC (Area Under Curve)* 1.000.

#### 4.5. Komparasi Hasil Pengujian

Pada tahap ini penulis melakukan komparasi terhadap hasil penelitian percobaan 2 metode yang digunakan yaitu *Decission Tree C4.5* dan *Support Vector Machine (SVM)* yang masing-masing tahapannya dilakukan dengan menggunakan seleksi fitur *Particel Sward Optimazion*. Penulis menyajikan hasil komparasi melalui table:

**Tabel 4.5.** Tabel Hasil Komparasi

Metode	Nilai Accuracy	Nilai AUC	F <sub>1</sub> -Score
<i>C4.5</i>	95,58%	0.917	0.9918
<i>SVM</i>	96,59%	1.000	1
<i>C4.5+PSO</i>	96,56%	0.952	0.9918
<i>SVM+PSO</i>	99,11%	1.000	1

#### 4.6. 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 Decision Tree C4.5 dan Support Vector Machine (SVM) dalam Deteksi Penderita Stroke 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 4.5), percobaan yang dilakukan pada *dataset* stroke yang diolah menggunakan *Decision Tree C4.5* tanpa menggunakan seleksi fitur didapatkan nilai akurasi 95,58% dengan nilai *AUC* 0.917, pada metode *Support Vector Machine (SVM)* tanpa menggunakan seleksi fitur didapatkan nilai akurasi 96,59%, dengan nilai *AUC* 1.000.

Selanjutnya jika kedua metode tersebut di optimasi dengan menggunakan *Particel Sward Optimazion (PSO)* maka akan menunjukkan hasil yang cukup signifikan dimana metode *Decision Tree C4.5* dengan fitur *Particel Sward Optimazion (PSO)* mendapatkan nilai akurasi 96.56% dan nilai *AUC* 0.952, hasil nilai akurasi yang didapat meningkat sebesar 0,98% dan nilai *AUC* 0.035.

Untuk metode *Support Vector Machine (SVM)* dengan *Particel Sward Optimazion (PSO)* mendapatkan peningkatan nilai akurasi yang sangat baik dimana metode ini mendapatkan nilai akurasi 99,11% dan nilai *AUC* tetap di 1.000, hasil nilai akurasi yang didapat meningkat sebesar 2,52% dan nilai *AUC* tetap

Dengan demikian Metode *Support Vector Machine (SVM)* berbasis *Particel Sward Optimazion (PSO)* menjadi yang lebih baik dalam penelitian ini, dibanding algoritma pembandingnya *Decision Tree C4.5*. Dengan hasil yang diperoleh SVM pada penelitian ini dengan nilai *accuracy* mencapai 99,11% dan Nilai *AUC* 1.000

Mengacu pada [19] *performance* keakurasian *AUC* dapat diklasifikasikan menjadi lima kelompok yaitu:

1.  $0,90 - 1,00 = \textit{Excellent Classification}$
2.  $0,80 - 0,90 = \textit{Good Classification}$
3.  $0,70 - 0,80 = \textit{Fair Classification}$
4.  $0,60 - 0,70 = \textit{Poor Classification}$
5.  $0,50 - 0,60 = \textit{Failure Classification}$

Algoritma *Decision Tree C4.5* dengan nilai *AUC* 0.917 termasuk algoritma yang *Excellent* (unggul) untuk memprediksi penyakit penderita stroke karena nilai *AUC* termasuk dalam predikat *Excellent Classification* ( $0,90 - 1,00$ ).

Algoritma *Decision Tree C4.5 Particel Sward Optimazion (PSO)* dengan nilai *AUC* 0.952 termasuk algoritma yang *Excellent* (unggul) untuk memprediksi penyakit penderita stroke karena nilai *AUC* termasuk dalam predikat *Excellent Classification* ( $0,90 - 1,00$ ).

Algoritma *Support Vector Machine (SVM)* dengan nilai *AUC* 1.000 termasuk algoritma yang *Excellent* (unggul) untuk memprediksi penyakit penderita stroke karena nilai *AUC* termasuk dalam predikat *Excellent Classification* ( $0,90 - 1,00$ ).

Berdasarkan klasifikasi *Performance* tersebut maka dapat disimpulkan bahwa Algoritma *Support Vector Machine (SVM)* berbasis *Particel Sward Optimazion (PSO)*, termasuk algoritma yang *Excellent* (unggul) untuk memprediksi penyakit penderita stroke karena nilai *AUC* termasuk dalam predikat *Excellent Classification* ( $0,90 - 1,00$ ).

Pendapat tersebut diperkuat dengan hasil *F<sub>1</sub>-Score*, dari metode *Support Vector Machine* dengan dua pemrosesan mendapat nilai sama yaitu 1, dan metode *C4.5* dengan dua pemrosesan mendapat nilai sama 0,9918. Dikarenakan semakin besar nilai dari *F<sub>1</sub>-Score* menunjukkan kinerja yang lebih baik dari system klasifikasi.