

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Penerapan Algoritma C4.5

Algoritma C4.5 merupakan sebuah model dengan membentuk pohon keputusan. Untuk membuat sebuah pohon keputusan, berikut langkah-langkah yang harus dilakukan:

1. Menyiapkan data training yang diambil dari data histori yang pernah terjadi sebelumnya dan dikelompokkan ke dalam kelas-kelas tertentu.
2. Menentukan akar dari pohon. Akar diambil dari atribut terpilih dengan cara menghitung nilai gain dari masing-masing atribut, nilai gain yang paling tinggi yang akan menjadi akar pertama. Sebelum menghitung nilai gain, maka perlu dicari dulu nilai entropinya dengan persamaan sebagai berikut:

$$Entropy (S) = \sum_{i=1}^n -p_i * \log_2 p_i$$

Tabel 4.1 Jumlah kasus dari tiap atribut

TOTAL		386	78	308
JK				
	L	106	23	83
	P	280	55	225
ALAMAT				
	METRO	72	10	62
	LAMTENG	87	21	66
	LAMTIM	170	33	137
	TUBA	10	3	7
	TUBABAR	4	4	0
	PESAWARAN	8	2	6
	LAMBAR	2	1	1
	MESUJI	9	5	4

TANGGAMUS	1	1	0
WAY KANAN	5	1	4
PRINGSEWU	1	0	1
ELSE	5	1	4

Jumlah kasus lengkap dari setiap atribut dapat dilihat pada halaman lampiran 1 jumlah kasus setiap atribut.

Berikut contoh cara mencari nilai entropy untuk atribut total kasus dan jenis kelamin pria dan wanita adalah sebagai berikut:

$$E_{totalkasus}(78,308) = \frac{78}{386} * \log\left(\frac{78}{386}\right) + \left(-\frac{308}{386} * \log\left(\frac{308}{386}\right)\right)$$

$$= 0,726053857$$

$$E_{jkpria}(23,83) = \left(-\frac{23}{106} * \log\left(\frac{23}{106}\right)\right) + \left(-\frac{83}{106} * \log\left(\frac{83}{106}\right)\right)$$

$$= 0,754616702$$

$$E_{jkwanita}(55,225) = \left(-\frac{55}{280} * \log\left(\frac{55}{280}\right)\right) + \left(-\frac{225}{280} * \log\left(\frac{225}{280}\right)\right)$$

$$= 0,714727473$$

Dengan perhitungan yang sama dilakukan terhadap tiap atribut dengan berdasarkan pada pengelompokan jumlah kasus pada tiap atribut.

Perhitungan selengkapnya dapat dilihat di lampiran 2 perhitungan manual mencari entropy

- Langkah selanjutnya menghitung nilai gain dengan persamaan sebagai berikut:

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i)$$

Contoh perhitungan nilai gain pada atribut jenis kelamin yaitu:

Information Gain =

$$(0.0726053857 - ((\frac{106}{386}) * 0.751616702) - ((\frac{280}{386}) * 0,714727473))$$

$$= 0,000372347.$$

Dengan perhitungan yang sama, setiap atribut dicair nilai *information gain* nya hingga semua atribut mendapatkan nilainya.

Perhitungan selengkapnya dapat dilihat di lampiran III perhitungan manual mencari nilai gain

4.Ulangi langkah 2 dan 3 untuk mencari *entropy* serta nilai *information gain* pada setiap atribut lainnya hingga semua mendapatkan nilai entropy dan information gain.

Tabel 4.2 Berikut adalah sampel dari hasil perhitungan entropy dan nilai gain

Atribut	Nilai	Jumlah Kasus	Lulus	Tidak Lulus	Entropy	Weight Info Gain
TOTAL		386	78	308	0,726053857	
JK						0,000372347
	L	106	23	83	0,754616702	
	P	280	55	225	0,714727473	
ALAMAT						0,035987247
	METRO	72	10	62	0,581321499	
	LAMTENG	87	21	66	0,79732651	
	LAMTIM	170	33	137	0,710006223	
	TUBA	10	3	7	0,881290899	
	TUBABAR	4	4	0	0	
	PESAWARAN	8	2	6	0,811278124	
	LAMBAR	2	1	1	1	
	MESUJI	9	5	4	0,99107606	
	TANGGAMUS	1	1	0	1	
	WAY KANAN	5	1	4	0,721928095	
	PRINGSEWU	1	0	1	0	
	ELSE	5	1	4	0,721928095	

Tabel hasil perhitungan entropy dan nilai gain selengkapnya dapat dilihat pada halaman lampiran IV tabel hasil nilai entropy dan nilai gain

4.1.2 Model C4.5 dioptimasi PSO

Pencarian bobot setiap atribut didalam algoritma PSO menggunakan persamaan 2.5 dan persamaan 2.6. Bobot dari sampel data ditampilkan dalam tabel berikut:

Tabel 4.3 Contoh sampel perhitungan bobot PSO

	eng_4	jml sdra	gaji	job
iterasi 2	0.363	3.780	2.280	3.515
iterasi 3	0.309	3.705	1.932	3.245
iterasi 4	0.268	3.180	1.620	2.885
iterasi 5	0.237	2.556	1.322	2.405
iterasi 6	0.214	1.800	0	1.760
iterasi 7	0.197	0	0	0
iterasi 8	0.184	0	0	0
iterasi 9	0.174	0	0	0
iterasi 10	0.167	0	0	0

Langkah dalam untuk menghitung jumlah bobot dengan persamaan 2.5 dalam algoritma PSO adalah sebagai berikut:

1. Inisialisasi partikel

Penginisialisasian partikel merupakan proses dimana atribut yang ada dimasukkan kedalam model partikel. Sampel partikel yang diinisialisaikan adalah sebagai berikut:

Tabel 4.4 Contoh sampel inisialisasi partikel

ENG4	Jumlah saudara	Penghasilan	
A	0	Rendah	1000.000-2000.000
A+	1	Sedang	2000.000-3000.000
B+	2	Tinggi	3000.000-4000.000
	3		
	4		

Semua atribut nantinya akan dijadikan sebuah partikel.

Selengkapnya dapat dilihat di halaman lampiran V tabel inisialisasi partikel

2. Pembangkitan posisi dan kecepatan partikel

Posisi dan kecepatan partikel bisa dimisalkan sebagai sebuah slot. Nilai yang dihasilkan posisi dan kecepatan ditentukan secara acak dengan adanya batas minimal dan batas maksimal slot. Nilai posisi dan kecepatan awal sebuah partikel diinisialisasi sama, sehingga hanya melakukan satu kali random untuk mendapatkan nilai kecepatan dan posisi partikel.

Pada tahap ini dilakukan proses pembangkitan posisi dan juga kecepatan sampel partikel yang telah tercantum di tabel atas. Pembangkitan posisi dan kecepatan ini dilakukan secara acak dengan menggunakan persamaan 2.5. Nilai X_{min} (batas terendah) dan X_{max} (batas tertinggi) untuk setiap tabel sampel diatas adalah sebagai berikut:

Batas bawah / Batas atas	Tabel		
	ENG4	Jumlah Saudara	Penghasilan
X_{min}	83	5	3
X_{max}	70	1	1

Contoh perhitungan untuk pembobotan jumlah saudara pada iterasi 2 dengan algoritma PSO sebagai berikut:

Perhitungan hasil awal posisi (v) untuk partikel jumlah saudara berdasarkan persamaan 2.5 adalah sebagai berikut:

Misalnya hasil random adalah 0.8

$$X_o^i = 1 + rand(5 - 1)$$

$$= 1 + 0.8 (5-1)$$

$$= 1 + 4 - 0.8$$

$$= \mathbf{4.2}$$

Dari perhitungan diatas, didapatkan hasil untuk posisi (v) awal dari partikel jumlah saudara, yakni $X = 4.2$. Untuk hasil awal kecepatan (x) partikel jumlah saudara, diinisialisasi sama dengan hasil posisi (v), yakni $V = 4.2$.

Langkah selanjutnya, hasil awal dari posisi (v) dan kecepatan (x) digunakan untuk proses mencari hasil *update* posisi (v) dengan persamaan 2.7 serta *update* kecepatan (x) dengan persamaan 2.8

Untuk menghasilkan hasil yang optimal, diberikan parameter ujicoba sebagai berikut:

$$C_1 = 1.5$$

$$C_2 = 1.5$$

$$W = 0.5$$

Parameter uji coba ini diambil dari hasil penelitian yang dilakukan oleh Irfans Kusmarna(2013) yang mana dengan parameter ini dapat menghasilkan rata-rata hasil yang optimal.

Proses untuk *update* posisi (v) untuk partikel jumlah saudara:

Misal hasil rand 1 = 0.1 dan rand 2 = 0.7

$$V_2 = 0.5 * 4.2 + 1.5 * 0.1(4.2-4.2) + 1.5 * 0.7(4.2-4.2)$$

$$= 0.5 * 4.2 + 1.5 * 0.42 - 0.42 + 1.5 * 2.94 - 2.94$$

$$= \mathbf{3.78}$$

Proses untuk *update* kecepatan (x) untuk partikel jumlah saudara:

$$X_o^i = 4.2 + 3.78$$

= 8.28

Semua partikel dilakukan pencarian seperti langkah-langkat diatas hingga semua partikel mendapatkan hasil.

Langkah untuk mencari bobot C4.5 pada PSO berdasarkan penelitian yang dilakukan oleh Khoirul Muarif(2013) adalah sebagai berikut:

1. Mencari nilai probabilitas dengan persamaan:

Langkah awal untuk mencari bobot C4.5 pada PSO adalah dengan mencari probabilitas dari setiap subset atribut yang dijadikan sampel.

Persamaan yang digunakan untuk mencari nilai probabilitas adalah sebagai berikut:

$$P(j|t) = \frac{n_j(t)}{\sum_i n_i(t)}$$

Dengan:

$P(j|t)$ = Probabilitas sampel

$n_j(t)$ = Banyak Sampel

$\sum_i n_i(t)$ = Total Sampel

Pencarian probabilitas ini berlaku bagi semua sub set atribut yang ada. Misal atribut jenis kelamin mempunyai subset atribut, yakni laki-laki dan perempuan, jadi subset atribut dari laki-laki dan perempuan harus dicari nilai probabilitasnya.

Contoh perhitungan manual untuk mencari nilai probabilitas subset dari atribut jenis kelamin, yakni subset atribut laki-laki dan perempuan:

Subset atribut laki-laki (L):

$$\text{probabilitas sampel JK (L)} = \frac{\text{Jumlah lulus}}{\text{jumlah kasus}}$$

$$\frac{23}{106} = \mathbf{0.216981132}$$

Subset atribut laki-laki (TL):

$$\text{probabilitas sampel JK (TL)} = \frac{\text{Jumlah lulus}}{\text{jumlah kasus}}$$

$$\frac{83}{106} = \mathbf{0.783018868}$$

Berdasarkan perhitungan diatas, dapat dilihat bahwa nilai probabilitas untuk subset atribut jenis kelamin laki-laki yang lulus seleksi adalah **0.216981132**, sedangkan untuk subset atribut jenis kelamin laki-laki yang tidak lulus adalah **0,783018868**.

Sebagian hasil perhitungan probabilitas dari setiap subset atribut akan ditampilkan pada tabel berikut:

		Lulus	Tidak Lulus
JK	L	0.216981132	0.783018868
	P	0.196428571	0.803571429
Pendidikan Ortu	SMP	0.20441989	0.79558011
	SMA	0.176470588	0.823529412
	Diploma	0.125	0.875
	Sarjana	0.297297297	0.702702703
Penghasilan	Level 1	0.179723502	0.820276498

	Level 2	0.2	0.8
	Level 3	0.12	0.88

Daftar hasil pernhitungan probabilitas dari semua subset atribut dapat dilihat di halaman lampiran VI hasil perhitungan probabilitas atribut.

2. Mencari nilai bobot untuk setiap atribut yang menggunakan persamaan:

$$n_w^j(t) = w(j) * n_j(t)$$

Dengan:

$$n_w^j(t) = \text{nilai atribut yang memiliki bobot atribut}$$

$$w(j) = \text{bobot atribut}$$

$$n_j(t) = \text{banyak sampel}$$

Bobot atribut $w(j)$ didapat dari hasil *weight gain* yang sebelumnya sudah dihitung pada persamaan C4.5 sedangkan banyak sampel merupakan total sampel yang digunakan untuk penelitian.

Berikut contoh perhitungan untuk mencari nilai bobot atribut untuk jenis kelamin.

$$\text{Bobot atribut JK} = \text{bobot atribut} * \text{banyak sampel}$$

$$= 0.000372347 * 23$$

$$= \mathbf{0.008563981}$$

Maka nilai atribut jenis kelamin adalah 0.008563981.

Begitupun dengan atribut lainnya untuk dicari nilai bobot atributnya hingga semua atribut memiliki nilai bobot.

3. Setelah nilai probabilitas dari subset atribut dan nilai bobot ditemukan, maka langkah selanjutnya yakni kalikan nilai probabilitas setiap subset atribut dengan hasil nilai bobot atribut.

Berikut akan diperlihatkan contoh perhitungan perkalian untuk atribut jenis kelamin:

Probabilitas dari subset atribut jenis kelamin laki-laki adalah

		Lulus	Tidak Lulus
JK	L	0.216981132	0.783018868
	P	0.196428571	0.803571429

Nilai bobot dari atribut laki-laki adalah **0.008563981**. Maka dapat dihitung menjadi:

$$JK_laki-laki_lulus = 0.216981132 * 0.008563981 = 0.001858222$$

$$JK_laki-laki_tidak\ lulus = 0.783018868 * 0.008563981 = 0.006705759$$

Perhitungan ini dilakukan ke semua atribut hingga kesemuanya mendapatkan hasil dari perkalian keduanya.

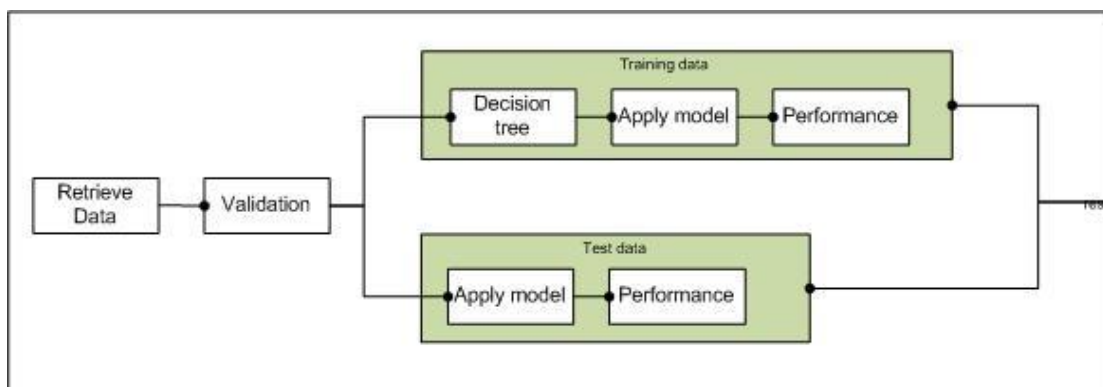
4. Lalu mencari nilai entropy dan gain kembali dengan persamaan 2.1 dan persamaan 2.2

4.2 Pembahasan

4.2.1 Validasi dan evaluasi

Dalam tahap ini dilakukan validasi dan pengukuran keakuratan hasil yang dicapai oleh model C4.5 dan C4.5 berbasis PSO yang dihasilkan dari tool atau alat bantu yang mana akan berpengaruh pada hasil pohon keputusan yang terbentuk.

Adapun desain model C4.5 dari adalah sebagai berikut:



Gambar 4.1 Desain model validasi C4.5

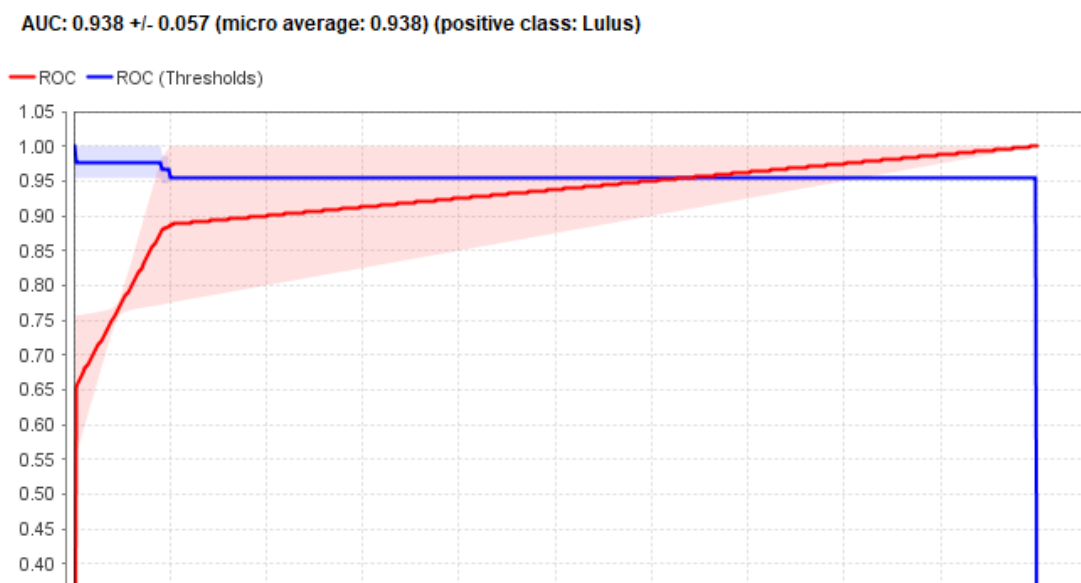
Retrieve berfungsi untuk memasukkan dataset. Validasi menggunakan cross-validation. Didalam validation terdapat dua kolom, yaitu kolom training dan testing. Didalam kolom training terdapat algoritma klasifikasi yang digunakan yakni algoritma C.45, sedangkan di dalam kolom testing terdapat apply model yang berguna untuk menjalankan algoritma C4.5 dan performance yang berguna untuk mengukur performa dari algoritma C4.5.

Adapun hasil dari pemodelan C4.5 yang di proses oleh tools adalah sebagai berikut:

	True Lulus	True Tidak Lulus	Class Precision
Pred.Lulus	61	17	78.21%
Pred.Tidak Lulus	71	291	94.48%
Class recall	78.21%	94.48%	

Accuracy: 91.20% +/- 3.49% (mikro:91.19%)

Gambar 4.2 Nilai akurasi hasil dari C4.5

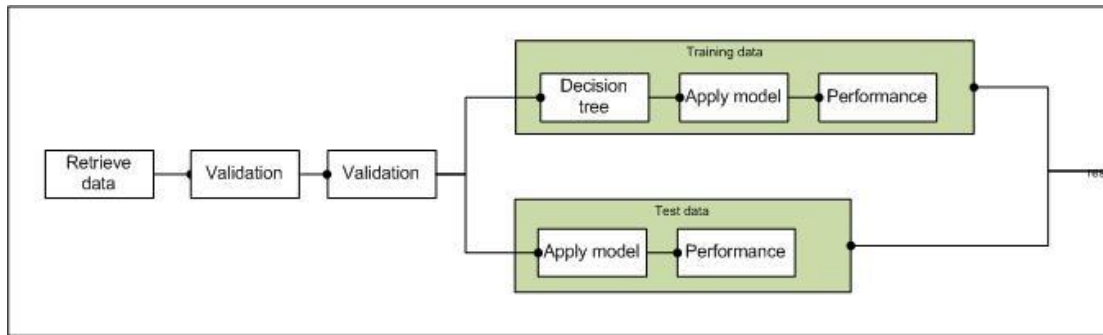


Gambar 4.3 Hasil performa AUC dari C4.5

Dari hasil pemrosesan diatas, dapat dievaluasi nilai akurasinya dengan tabel confusion matrik, sebagai berikut:

$$\text{Akurasi} = \frac{61+17}{61+17+71+291} \times 100\% = 91,20\%$$

Sedangkan desain model C4.5 berbasis PSO pada pembobotan atribut pada RapidMiner adalah sebagai berikut:



Gambar 4.4 Desain model C4.5 berbasis PSO

Optimize weight (PSO) berguna untuk menerapkan algoritma PSO pada pembobotan atribut. Didalam Optimize Weight (PSO) terdapat validation yang menggunakan cross-validation. Didalam validation terdapat dua kolom, yaitu training dan testing. Didalam kolom training terdapat algoritma klasifikasi yang digunakan yakni C4.5, sedangkan didalam kolom testing terdapat apply model untuk menjalankan algoritma C4.5 serta performance untuk mengukur performance dari model tersebut.

Hasil percobaan model C4.5 dengan PSO dapat dilihat pada tabel dibawah ini:

	True Lulus	True Tidak Lulus	Class Precision
Pred.Lulus	63	6	91.30%
Pred.Tidak Lulus	15	302	95.27%
Class recall	80.77%	98.05%	

Accuracy: 94.57% +/- 2.29% (mikro:94.56%)

Dari hasil pemrosesan diatas, dapat dievaluasi nilai akurasinya dengan tabel confusion matrik, sebagai berikut:

$$\text{Akurasi} = \frac{63+6}{63+6+15+302} \times 100\% = 94,57\%$$

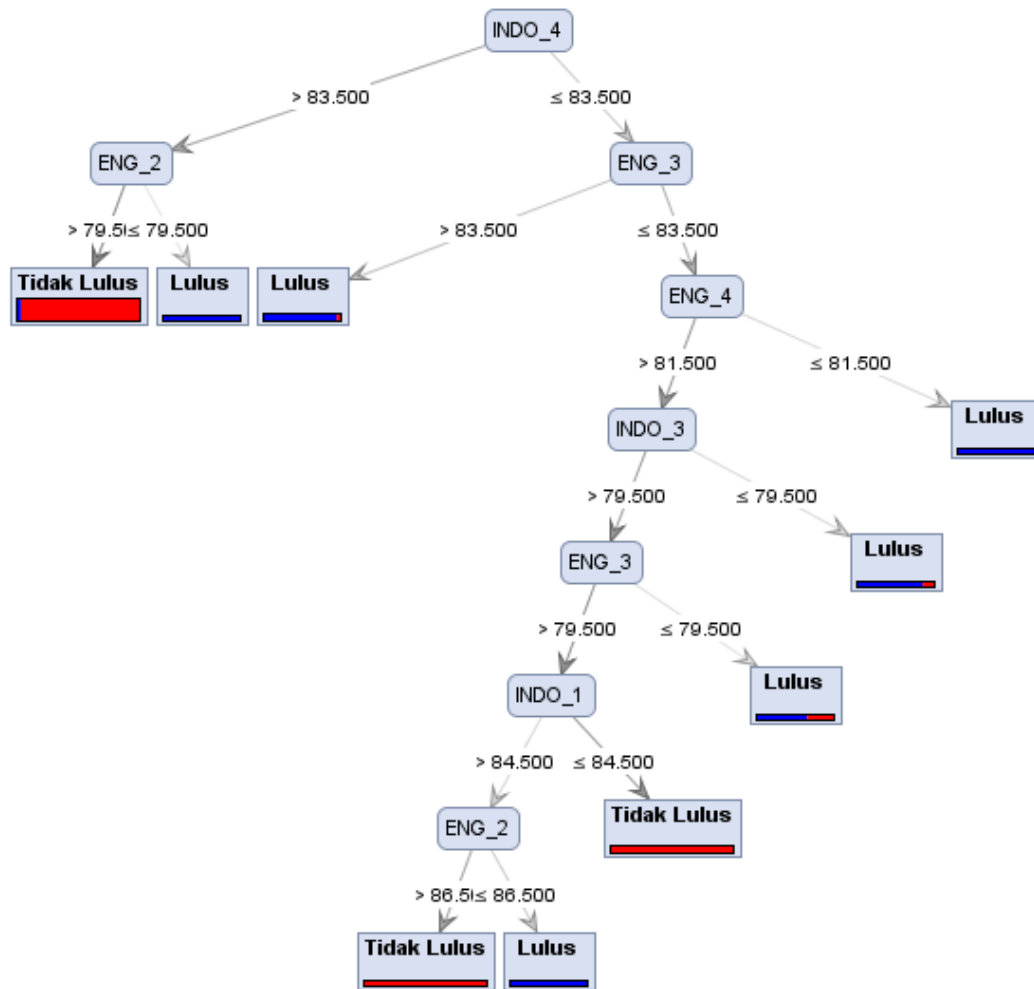
Setelah melakukan pemodelan dan perhitungan berdasarkan pada kedua algoritma, kemudian dilakukan perbandingan hasil yang berupa nilai akurasi, maka diperoleh data perbandingan sebagai berikut:

Perbandingan	C4.5	C4.5 dengan PSO
Performance	91.20%	94,56%
Waktu eksekusi	0 detik	2 menit 23 detik

4.2.2 Hasil pemodelan

4.2.2.1 Tree dan Rule C4.5

Hasil pemodelan C4.5 akan menghasilkan sebuah pohon keputusan. Pohon keputusan yang terbentuk dari hasil pengujian pada penelitian ini dapat dilihat pada gambar dibawah ini:



Dari pohon keputusan yang terbentuk diatas, dapat di artikan sebagai berikut:

1. Jika nilai INDO_4 lebih kecil dari 83.50 dan nilai ENG_2 lebih kecil dari pada 79.50, maka siswa dinyatakan tidak lulus seleksi.
2. Jika nilai INDO_4 lebih kecil dari 83.50 dan nilai ENG_2 lebih besar sama dengan dari pada 79.50, maka siswa dinyatakan lulus seleksi.
3. Jika nilai INDO_4 lebih kecil sama dengan 83.50 dan nilai ENG_3 lebih kecil dari pada 83.50, maka siswa dinyatakan lolos seleksi.

4. Jika nilai INDO_4 lebih besar sama dengan 83.50 dan nilai ENG_3 lebih besar sama dengan 83.50 dan nilai ENG_4 lebih besar sama dengan 81.50 maka siswa dinyatakan lolos seleksi.
5. Jika nilai INDO_4 lebih besar sama dengan 83.50 dan nilai ENG_3 lebih besar sama dengan 83.50 dan nilai ENG_4 lebih kecil dari pada 81.50 dan nilai INDO_3 lebih besar sama dengan 79.50 maka siswa dinyatakan lolos seleksi.
6. Jika nilai INDO_4 lebih besar sama dengan 83.50 dan nilai ENG_3 lebih besar sama dengan 83.50 dan nilai ENG_4 lebih kecil dari pada 81.50 dan INDO_3 lebih kecil dari pada 79.50 dan ENG_3 lebih besar sama dengan 79.50 maka siswa dinyatakan lolos.
7. Jika nilai INDO_4 lebih besar sama dengan 83.50 dan nilai ENG_3 lebih besar sama dengan 83.50 dan nilai ENG_4 lebih kecil dari pada 81.50 dan INDO_3 lebih kecil dari pada 79.50 dan ENG_3 lebih besar 79.50 dan INDO_1 lebih besar sama dengan 84.50 maka siswa tidak lulus seleksi.
8. Jika nilai INDO_4 lebih besar sama dengan 83.50 dan nilai ENG_3 lebih besar sama dengan 83.50 dan nilai ENG_4 lebih kecil dari pada 81.50 dan INDO_3 lebih kecil dari pada 79.50 dan ENG_3 lebih besar 79.50 dan INDO_1 lebih kecil dari pada 84.50 dan nilai ENG_2 lebih kecil dari pada 86.50 maka siswa dinyatakan tidak lulus seleksi.
9. Jika nilai INDO_4 lebih besar sama dengan 83.50 dan nilai ENG_3 lebih besar sama dengan 83.50 dan nilai ENG_4 lebih kecil dari pada 81.50 dan INDO_3 lebih kecil dari pada 79.50 dan ENG_3 lebih besar 79.50 dan INDO_1 lebih kecil

dari pada 84.50 dan nilai ENG_2 lebih besar sama dengan 86.50 maka siswa dinyatakan lulus seleksi.

Adapun rule yang tercipta dari pemodelan pohon keputusan diatas adalah sebagai berikut:

Tree

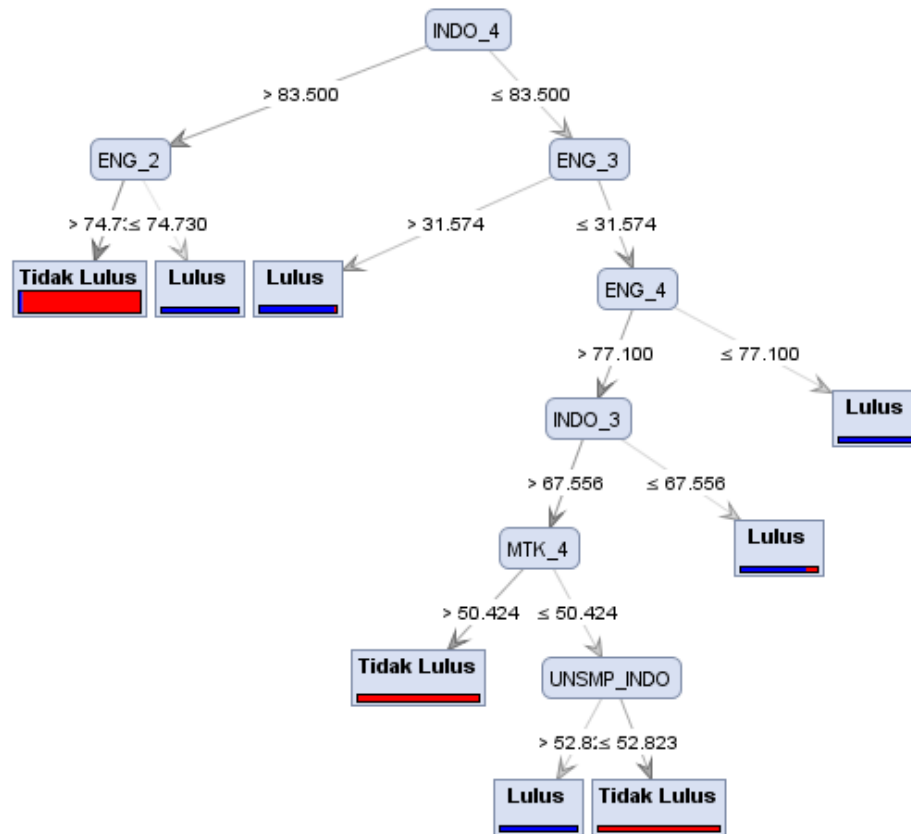
```

INDO_4 > 83.500
|   ENG_2 > 79.500: Tidak Lulus {Lulus=12, Tidak Lulus=262}
|   ENG_2 ≤ 79.500: Lulus {Lulus=3, Tidak Lulus=0}
INDO_4 ≤ 83.500
|   ENG_3 > 83.500: Lulus {Lulus=43, Tidak Lulus=2}
|   ENG_3 ≤ 83.500
|   |   ENG_4 > 81.500
|   |   |   INDO_3 > 79.500
|   |   |   |   ENG_3 > 79.500
|   |   |   |   |   INDO_1 > 84.500
|   |   |   |   |   |   ENG_2 > 86.500: Tidak Lulus {Lulus=0, Tidak
Lulus=5}
|   |   |   |   |   |   ENG_2 ≤ 86.500: Lulus {Lulus=2, Tidak
Lulus=0}
|   |   |   |   |   |   INDO_1 ≤ 84.500: Tidak Lulus {Lulus=0, Tidak
Lulus=37}
|   |   |   |   |   |   ENG_3 ≤ 79.500: Lulus {Lulus=2, Tidak Lulus=1}
|   |   |   |   |   |   INDO_3 ≤ 79.500: Lulus {Lulus=6, Tidak Lulus=1}
|   |   |   |   |   |   ENG_4 ≤ 81.500: Lulus {Lulus=10, Tidak Lulus=0}

```

4.2.2.2 Tree dan Rule C4.5 dengan PSO

Setelah model C4.5 dioptimasi dengan algoritma PSO, maka akan terbentuk pula sebuah pohon keputusan dan rule yang tentunya berbeda dari pohon keputusan dan rule sebelum dioptimasi. Pohon keputusan dan rule perubahan tersebut dapat dilihat pada gambar berikut:



Sedangkan rule yang terbentuk sebagai berikut:

Tree

```

INDO_4 > 83.500
|   ENG_2 > 74.730: Tidak Lulus {Lulus=10, Tidak Lulus=238}
|   ENG_2 <= 74.730: Lulus {Lulus=2, Tidak Lulus=0}
INDO_4 <= 83.500
|   ENG_3 > 31.574: Lulus {Lulus=41, Tidak Lulus=1}
|   ENG_3 <= 31.574
|   |   ENG_4 > 77.100
|   |   |   INDO_3 > 67.556
|   |   |   |   MTK_4 > 50.424: Tidak Lulus {Lulus=0, Tidak
Lulus=29}
|   |   |   |   MTK_4 <= 50.424
|   |   |   |   |   UNSMP_INDO > 52.823: Lulus {Lulus=3, Tidak
Lulus=0}
|   |   |   |   |   UNSMP_INDO <= 52.823: Tidak Lulus {Lulus=0, Tidak
Lulus=7}
|   |   |   |   INDO_3 <= 67.556: Lulus {Lulus=6, Tidak Lulus=1}
|   |   |   ENG_4 <= 77.100: Lulus {Lulus=9, Tidak Lulus=0}

```