BAB IV
HASIL DAN PEMBAHASAN

1.1. Hasil Praproses Data

Proses pengolahan data mentah menjadi data siap merupakan salah satu proses penting yang perlu dilakukan sebelum data siap diolah untuk proses selanjutnya. Langkah ini disebut juga pembersihan data atau *data cleaning*. Data penelitian merupakan data sekunder dimana data telah ada sebelumnya yaitu data milik KPPS. BMT Adil Berkah Sejahtera yang dikumpulkan untuk kebutuhan penelitian sebanyak 1536 data berbentuk tabel berekstensi .xlsx. Sebelum proses implementasi model, dilakukan praproses data diantaranya menghapus data yang memiliki null atau *NaN* atau data kosong serta menghapus data duplikasi. Pada proses ini terdapat 110 data kosong atau *null* sehingga dataset berkurang jumlahnya menjadi 1426 data dengan sebanyak 1311 berlabel lancar dan 115 berlabel macet. Selain itu juga dilakukan perubahan terhadap kolom Masa karena memiliki jenis data yang berbeda yaitu minggu dan bulan sehingga perlu diubah menjadi satuan bulan untuk masa angsuran. Pada proses ini juga dilakukan pemilihan atribut yang akan digunakan pada proses klasifikasi. Potongan data yang siap diproses dapat dilihat pada Tabel 4.1. sedangkan dataset lengkap dapat dilihat pada Lampiran 2.

Tabel 4.1 Tabel potongan dataset yang sudah siap

Masa	Jenis Pembiayaan	Pokok	*Margin*	*Plafond*	Label
4	Modal Usaha	5000000	580000	5580000	Lancar
10	Modal Usaha	4000000	1000000	5000000	Lancar
2	Pendidikan	10000000	580000	10580000	Lancar
5	Multiguna	3000000	435000	3435000	Lancar
15	Pendidikan	15000000	3375000	18375000	Lancar
10	Modal Usaha	2000000	500000	2500000	Lancar
6	Modal Usaha	5000000	405000	5405000	Macet
4	Modal Usaha	2000000	232000	2232000	Lancar
10	Modal Usaha	2000000	500000	2500000	Lancar

1.2. Hasil Pengujian Menggunakan Naive Bayes

Dari hasil pengolahan, atribut serta label yang telah dipilih terdiri dari masa, jenis pembiayaan, pokok, margin, dan plafond yang dinotasikan sebagai X serta label yang dinotasikan sebagai Y. Pada Naive Bayes, fitur dan label tersebut digunakan untuk menghitung nilai probabilitas baik prior maupun *likelihood*. Pada nilai prior atau dalam persamaan disebut P(Y) masing-masing dicari nilai P(Y=macet) dan P(Y=lancar). Berikut adalah perhitungannya.

$$\text{P(Y = lancar)} = \frac{jumlah\ lancar}{jumlah\ total} = \frac{1311}{1426} = 0{,}919354839$$

$$\text{P(Y = macet)} = \frac{jumlah\ macet}{jumlah\ total} = \frac{115}{1426} = 0{,}080645161$$

Untuk perhitungan P(X|Y) atau *likelihood*, pada data kategorikal dicari menggunakan probabilitas atau peluang pada keadaan macet dan keadaan lancar. Berikut perhitungan nilai *likelihood* pada data jenis pembiayaan modal usaha.

$$\text{P(X=modal usaha|Y=lancar)} = \frac{jumlah\ modal\ usaha\ dan\ lancar}{jumlah\ modal\ usaha} = \frac{817}{900} = 0{,}907777778$$

$$\text{P(X=modal usaha|Y=macet)} = \frac{jumlah\ modal\ usaha\ dan\ macet}{jumlah\ modal\ usaha} = \frac{83}{900} = 0{,}092222222$$

Untuk data numerik, nilai probabilitas dicari menggunakan distribusi normalnya. Untuk mencari nilai distribusi normal terlebih dahulu dicari nilai rata-rata dan deviasi standar pada masing-masing atribut numerik. Berikut adalah persamaan untuk mencari rata-rata dan deviasi standar.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5.1}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n-1} \tag{5.2}$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.3}$$

Pada persamaan (5.1), x_i merupakan data ke-i, n merupakan jumlah data, sedangkan μ merupakan nilai rata-rata. Dan pada persamaan (5.2) σ merupakan nilai deviasi standar, dan persamaan (5.3), *f(x)* merupakan persamaan untuk menghitung distribusi normal dari nilai x. Sebagai contoh berikut perhitungan probabilitas pada atribut masa dengan nilai x sebesar 4.

Dari hasil perhitungan menggunakan persamaan sebelumnya, diperoleh perhitungan sebagai berikut.

μ(masa, lancar) = 6,757284516

μ(masa, macet) = 5,088695652

σ(masa, lancar) = 3,306217836

σ(masa, macet) = 2,836379123

$f_{lancar}(4)$ = 0,202149113

$f_{macet}(4)$ = 0,350551147

Setelah diperoleh perhitungan tersebut pada seluruh atribut, kemudain dicari probabilitas atau nilai posteriornya. Sebagai contoh, diberikan keadaan masa 4 bulan, jenis pembiayaan adalah modal usaha, pokok sebesar 5000000, *margin* 250000, *plafond* sebesar 5250000, maka perhitungan posteriornya adalah sebagai berikut.

$$\mathrm{P(Y = lancar|X)}$$

$$= \frac{P(X_1|Y = lanc) \times P(X_2|Y = lanc) \times P(X_3|Y = lanc) \times P(X_4|Y = lanc) \times P(X_5|Y = lanc) \times P(Y = lanc)}{P(X)}$$

$$P(Y = lancar|X)$$

$$= \frac{0{,}20214911 \times 0{,}907777778 \times 0{,}533804804 \times 0{,}291183526 \times 0{,}509041145 \times 0{,}919354839}{P(X)}$$

$$P(Y = lancar|X) = \ 0{,}013348631$$

Untuk posterior dengan Y=macet dilakukan dengan tahapan yang sama, hanya saja nilai prior menggunakan P(Y=macet) dimana dihasilkan nilai sebesar 0,001170933. Dari hasil perhitungan tersebut, nilai P(Y=lancar|X) > P(Y=macet|X) sehingga label pada data tersebut menunjukkan label lancar.

Pada pengujian ini, metode Naive Bayes diimplementasikan sebagai metode klasifikasi yang menggunakan *cross validation* pada pembagian data latih dan data uji sehingga data dibagi menjadi beberapa subset sesuai jumlah *folds* yang ditentukan. Untuk menentukan jumlah *folds* yang menghasilkan akurasi terbaik, dilakukan pengujian terhadap beberasi variasi jumlah *folds*. Pada penelitian ini digunakan jumlah folds sebanyak lima, sepuluh, dan lima belas.

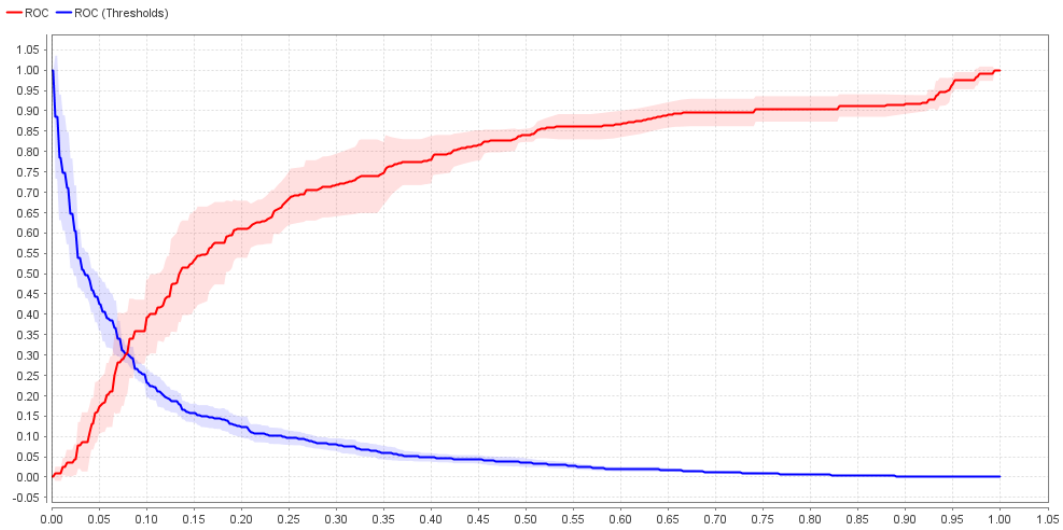
1.2.1. Pengujian Naive Bayes Menggunakan Lima *Folds*

Dari hasil pengujian menggunakan lima folds diperoleh akurasi model sebesar 89.20% dengan score AUC sebesar 0.744 dan nilai *F1-score* sebesar 0.0494. Hasil tangkapan layar confusion matrix dan grafik ROC dapat dilihat pada Gambar 4.1 dan Gambar 4.2.

accuracy: 89.20% +/- 1.54% (micro average: 89.20%)

	true Lancar	true Macet	class precision
pred. Lancar	1268	111	91.95%
pred. Macet	43	4	8.51%
class recall	96.72%	3.48%	

Gambar 4.1 Hasil tangkapan layar *confusion matrix* Naive Bayes dengan lima *folds*



Gambar 4.2 Grafik ROC Naive Bayes dengan lima *fold*

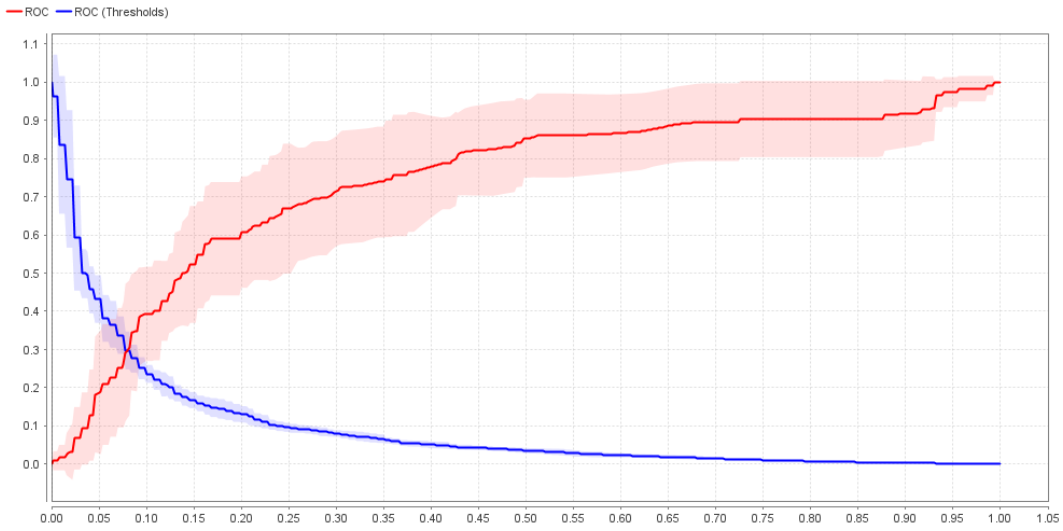
1.2.2. Pengujian Naive Bayes Menggunakan Sepuluh *Folds*

Pada pengujian Naive Bayes menggunakan sepuluh folds diperoleh akurasi yang sedikit lebih tinggi dibanding sebelumnya yaitu sebesar 89.90% dengan skor AUC sebesar 0.743 dan *F1-score* sebesar 0.0769. Hasil *confusion matrix* dan grafik ROC dapat dilihat pada tangkapan layar yang ditampilkan pada Gambar 4.3 dan Gambar 4.4.

accuracy: 89.90% +/- 1.05% (micro average: 89.90%)

	true Lancar	true Macet	class precision
pred. Lancar	1276	109	92.13%
pred. Macet	35	6	14.63%
class recall	97.33%	5.22%	

Gambar 4.3 Hasil tangkapan layar confusion matrix Naive Bayes dengan sepuluh *folds*



Gambar 4.4 Grafik ROC Naive Bayes dengan sepuluh *folds*

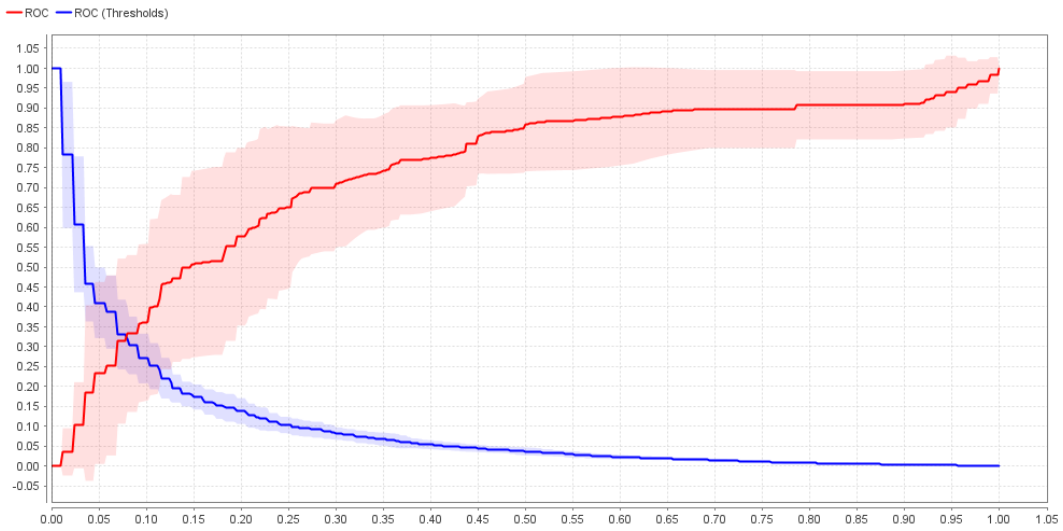
1.2.3. Pengujian Naive Bayes Menggunakan Lima Belas *Folds*

Pengujian Naive Bayes menggunakan 15 *folds* memiliki nilai akurasi yang sama dengan sepuluh *folds* yaitu sebesar 89.90% dan skor AUC sebesar 0.741, serta nilai *F1-score* sebesar 0.1. Hasil tangkapan layar *confusion matrix* dan grafik ROC dapat dilihat pada Gambar 4.5 dan Gambar 4.6.

accuracy: 89.90% +/- 1.18% (micro average: 89.90%)

	true Lancar	true Macet	class precision
pred. Lancar	1274	107	92.25%
pred. Macet	37	8	17.78%
class recall	97.18%	6.96%	

Gambar 4.5 Hasil tangkapan layar *confusion matrix* Naive Bayes dengan lima belas *folds*



Gambar 4.6 Grafik ROC Naive Bayes dengan lima belas *folds*

Dari hasil pengujian metode Naive Bayes, nilai akurasi serta F1-*score* tertinggi diperoleh saat menggunakan sepuluh dan lima belas fold. Perbandingan hasil pengujian dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2. Perbandingan hasil pengujian menggunakan Naive Bayes

No.	Jumlah *Fold*	Akurasi	*Precision*	*Recall*	F1-*score*	AUC
1.	5	89.20%	0.0844	0.0345	0.0494	0.744
2.	10	89.90%	0.1376	0.0515	0.0769	0.743
3.	15	89.90%	0.1778	0.0681	0.1	0.741

1.3. Hasil Klasifikasi Menggunakan Algoritma C4.5

Pada implementasi algoritma C4.5 *criterion* yang digunakan adalah *Gain Ratio* dan *max_depth* yang berjumlah 20. Pengujian juga dilakukan berdasarkan beberapa jumlah folds untuk melihat nilai folds yang menghasilkan akurasi tertinggi. Jumlah folds yang digunakan yaitu lima, sepuluh, dan lima belas.

Pada algoritma C4.5 terlebih dahulu dicari nilai *Gain Ratio* dari seluruh atribut. Untuk mencari *Gain Ratio,* terlebih dahulu dicari nilai *Entropy, Gain information, Split Info,* kemudian dicari nilai *Gain Ratio* masing-masing atribut. Atribut yang memiliki nilai Gain Ratio tertinggi akan menjadi node akar. Proses perhitungan nilai gain ratio kemudian diulang kembali hingga semua variabel pohon memiliki kelas. Berikut adalah contoh perhitungan untuk *Entropy, Gain Information, Split Info,* serta *Gain Ratio* pada atribut jenis pembiayaan.

$$Entropy_{total} = -\left(\frac{jml\ lancar}{jml\ total} \times log_2\left(\frac{jml\ lancar}{jml\ total}\right)\right) + \left(\frac{jml\ macet}{jml\ total} \times log_2\left(\frac{jml\ macet}{jml\ total}\right)\right)$$

$$Entropy_{total} = -\left(\left(\frac{1311}{1426} \times log_2\left(\frac{1311}{1426}\right)\right) + \left(\frac{115}{1426} \times log_2\left(\frac{115}{1426}\right)\right)\right) = 0{,}404448386$$

$$Entropy_{modal\ usaha} = -\left(\left(\frac{817}{900} \times log_2\left(\frac{817}{900}\right)\right) + \left(\frac{83}{900} \times log_2\left(\frac{83}{900}\right)\right)\right) = 0{,}443844129$$

Nilai *entropy* dicari pada seluruh kategori yang ada pada atribut jenis pembiayaan kemudian dicari nilai *information gain.* Berikut adalah perhitungan *information gain* pada atribut jenis pembiayaan.

$$Information\ Gain = Entropy_{total} - \sum_{i=1}^{k} \frac{|S_i|}{S} \times Entropy(S_i)$$

$$Info\ Gain = 0{,}404448386 - \left(\left(\frac{900}{1426} \times 0{,}443844129\right) + \left(\frac{259}{1426} \times 0{,}364047747\right) + \left(\frac{248}{1426} \times 0{,}279505546\right) + \left(\frac{19}{1426} \times 0{,}485460761\right)\right) = 0{,}003123553$$

Dari hasil perhitungan menggunakan persamaan tersebut, dimana dicari total seluruh *Information Gain* pada semua kategori, pada atribut jenis pembiayaan

memiliki *Information Gain* sebesar 0,003123553. Langkah selanjutnya yang dilakukan adalah menghitung nilai *Split Info* dan *Gain Ratio*. Perhitungan *Split Info* dan *Gain Ratio* dapat dilihat pada tahapan berikut ini.

$$SplitInfo(S,A) = -\sum_{i=n}^{n} . \frac{si}{s} \log 2 \frac{si}{s}$$

$$SplitInfo(S,A) = -\left(\left(\frac{900}{1426} \times log_2 \frac{900}{1426}\right) + \left(\frac{259}{1426} \times log_2 \frac{259}{1426}\right) + \left(\frac{248}{1426} \times log_2 \frac{248}{1426}\right) + \left(\frac{19}{1426} \times log_2 \frac{19}{1426}\right)\right) = 1{,}387921151$$

$$Gain\ Ratio = \frac{Gain(S,A)}{Split\ Info(S,A)}$$

$$Gain\ Ratio_{jenis\ pembiayaan} = \frac{0{,}003123553}{1{,}387921151} = \mathbf{0{,}002250526}$$

Sementara itu, perhitungan *Gain Ratio* pada data numerik seperti *plafond*, dilakukan dengan membagi atribut menjadi dua kelas. Berikut adalah contoh perhitungan pada atribut *plafond.*

$$jumlah\ plafond_{>11605000} = 96$$

$$jumlah\ plafond_{<11605000} = 1330$$

$$Entropy_{>11605000} = -\left(\left(\frac{68}{96} \times log_2\left(\frac{68}{96}\right)\right) + \left(\frac{28}{96} \times log_2\left(\frac{28}{96}\right)\right)\right) = 0{,}870864469$$

$$Entropy_{<11605000} = -\left(\left(\frac{1243}{1330} \times log_2\left(\frac{68}{1330}\right)\right) + \left(\frac{87}{1330} \times log_2\left(\frac{28}{1330}\right)\right)\right) = 0{,}348569902$$

$$Info\ Gain = 0{,}404448386 - \left(\left(\frac{96}{1426} \times 0{,}870864469\right) + \left(\frac{1330}{1426} \times 0{,}348569902\right)\right) = 0{,}020716999$$

$$Split\ Info = -\left(\left(\frac{96}{1426} \times log_2 \frac{96}{1426}\right) + \left(\frac{1330}{1426} \times log_2 \frac{1330}{1426}\right)\right) = 0{,}355846341$$

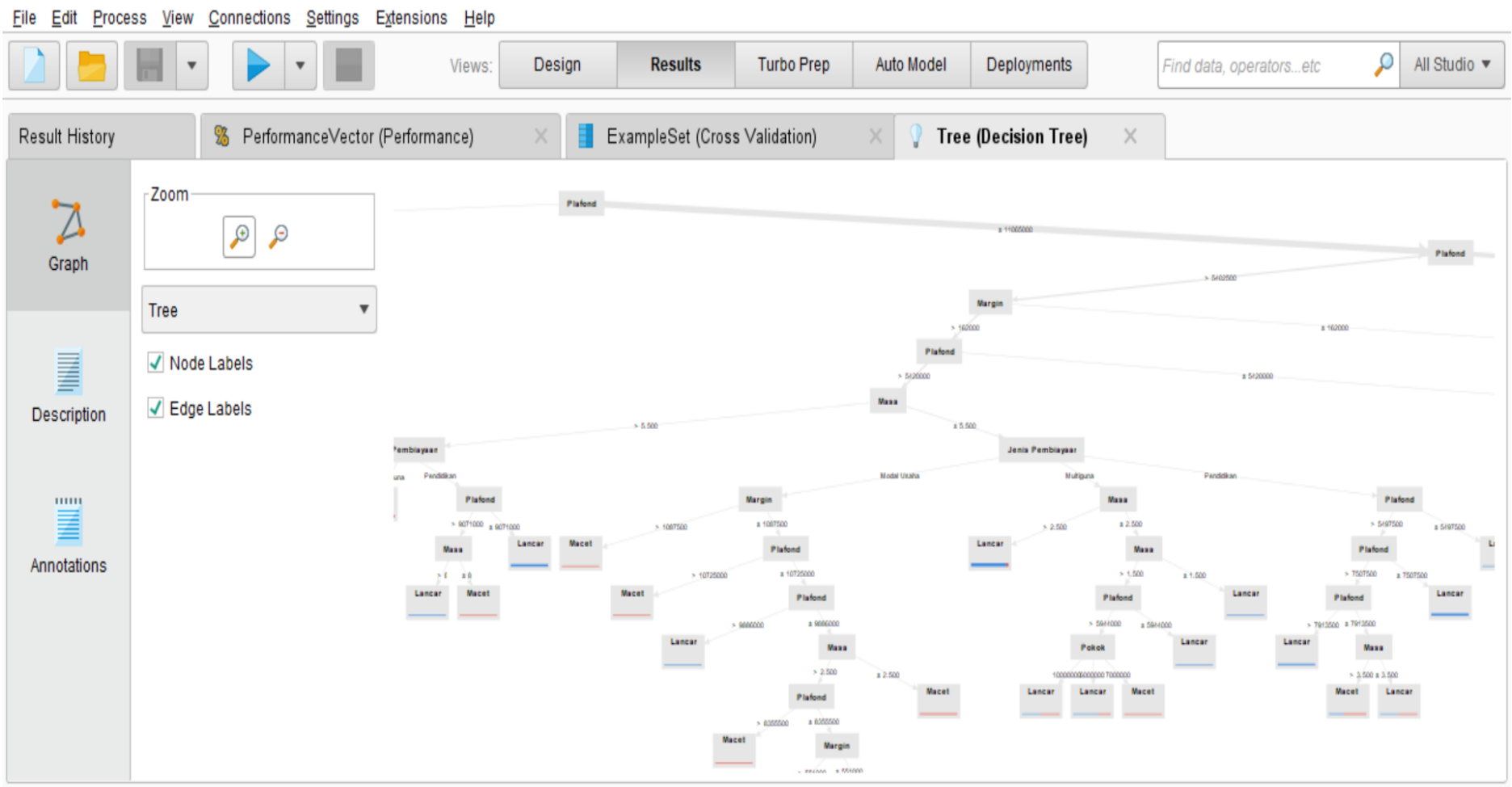
$$Gain\ Ratio_{plafond} = \frac{0{,}020716999}{0{,}355846341} = \mathbf{0{,}058218947}$$

Setelah dilakukan perhitungan terhadap seluruh atribut, *plafond* memiliki nilai *Gain Ratio* tertinggi sehingga atribut *plafond* ditentukan sebagai simpul akar. Hasil Gain ratio pada fitur dapat dilihat pada Gambar 4.7.

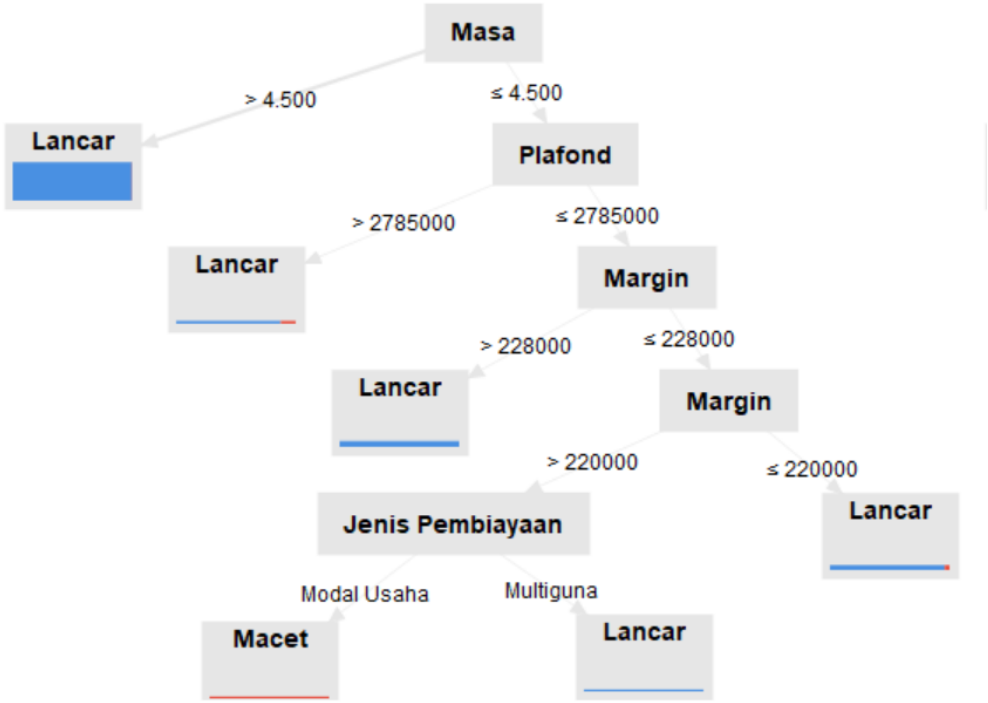
attribute	weight
Plafond	0.058
Margin	0.020
Masa	0.019
Pokok	0.017
Jenis Pe...	0.002

Gambar 4.7. Hasil perhitungan *Gain Ratio* pada RapidMiner

Pada Gambar 4.7 dapat dilihat hasil perhitungan *Gain Ratio* masing-masing atribut menggunakan RapidMiner. Atribut *plafond* yang memiliki nilai tertinggi dijadikan simpul akar. Atribut *margin* memiliki nilai tertinggi kedua, sehingga dijadikan node akar selanjutnya, dilanjutkan dengan atribut masa, lalu pokok, dan terakhir adalah jenis pembiayaan yang memiliki nilai *Gain Ratio* terkecil, sehingga menjadi node akar paling bawah, sedangkan *leaf* atau node daun berisi target kelas, yaitu kelas lancar dan macet Diagram dari pohon atau *tree* yang terbentuk dapat dilihat pada Gambar 4.8 dan Gambar 4.9.



Gambar 4.8 Tampilan *tree* yang terbentuk pada aplikasi RapidMiner (1)



Gambar 4.9 Tampilan *tree* pada *branches* (ranting) dan *leaf node*(daun)

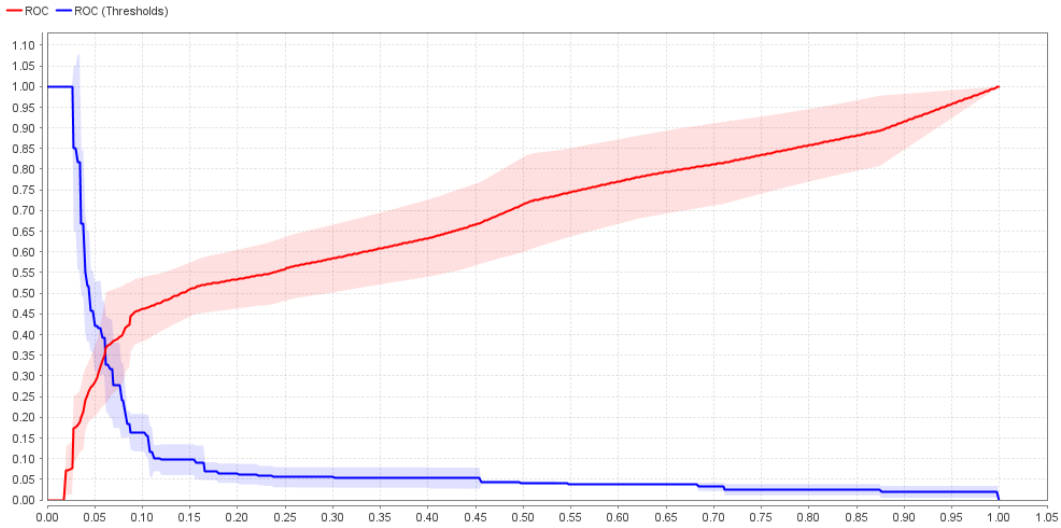
1.3.1. Pengujian Algoritma C4.5 Menggunakan Lima *Folds*

Implementasi algoritma C4.5 menggunakan lima folds menghasilkan akurasi sebesar 90.18% dengan AUC *score* sebesar 0.681, dan nilai F1-*score* sebesar 0.2223. Tampilan hasil *confusion matrix* dan grafik ROC dapat dilihat pada Gambar 4.10 dan Gambar 4.11.

accuracy: 90.18% +/- 0.83% (micro average: 90.18%)

	true Lancar	true Macet	class precision
pred. Lancar	1265	94	93.08%
pred. Macet	46	21	31.34%
class recall	96.49%	18.26%	

Gambar 4.10 Hasil tangkapan layar *confusion matrix* algoritma C4.5 dengan lima *fold*s



Gambar 4.11 Grafik ROC algoritma C4.5 dengan lima *folds*

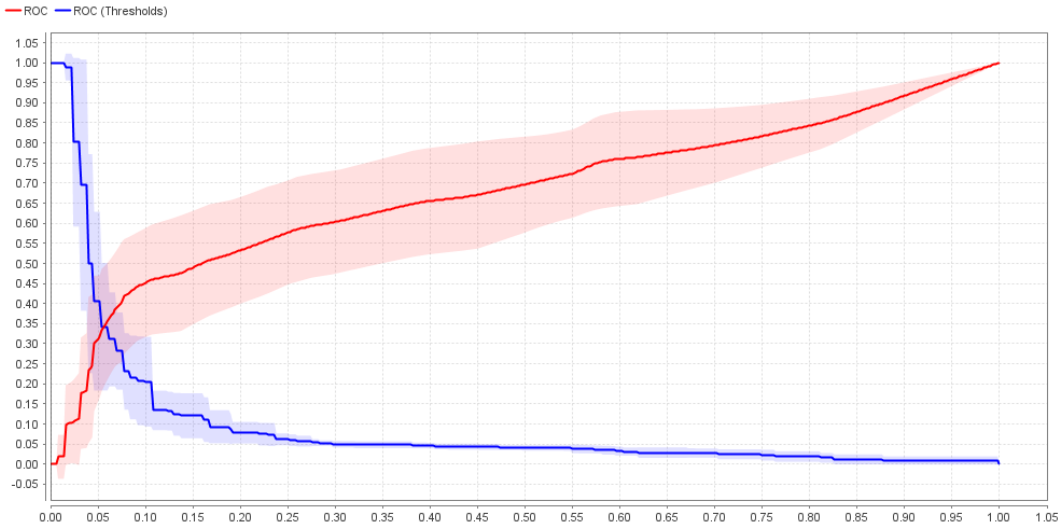
1.3.2. Pengujian Algoritma C4.5 Menggunakan Sepuluh *Folds*

Pengujian algoritma C4.5 menggunakan sepuluh folds menghasilkan akurasi yang lebih tinggi yaitu sebesar 90.60% dengan nilai AUC sebesar 0.679, dan *F1-score* sebesar 0.2299. Hasil perhitungan akurasi dan kurva ROC dapat dilihat pada Gambar 4.12 dan Gambar 4.13.

accuracy: 90.60% +/- 0.69% (micro average: 90.60%)

	true Lancar	true Macet	class precision
pred. Lancar	1272	95	93.05%
pred. Macet	39	20	33.90%
class recall	97.03%	17.39%	

Gambar 4.12 Hasil tangkapan layar *confusion matrix* algoritma C4.5 dengan sepuluh *folds*



Gambar 4.13 Grafik ROC algoritma C4.5 dengan sepuluh *folds*

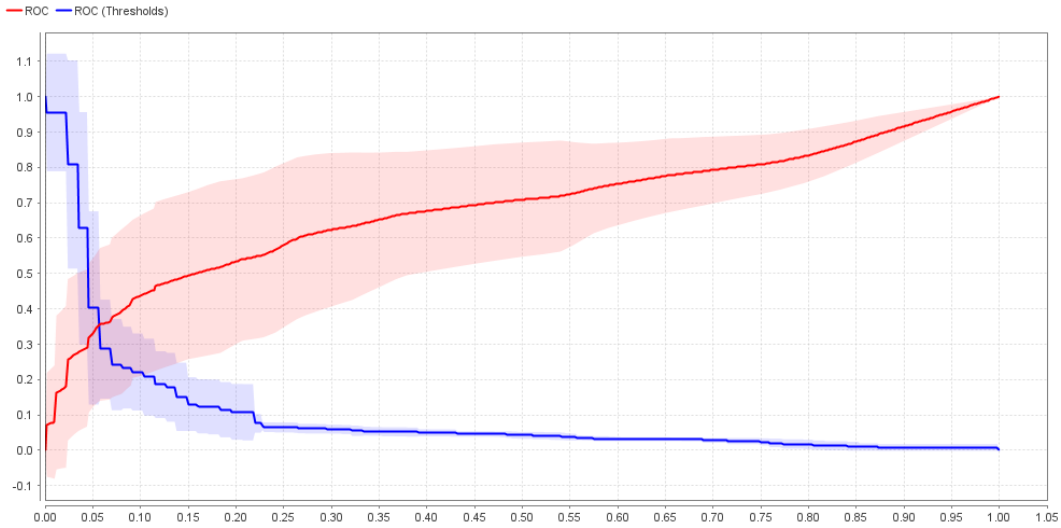
1.3.3. Pengujian Algoritma C4.5 Menggunakan Lima Belas *Folds*

Pada pengujian menggunakan 15 *folds* akurasi yang dihasilkan memiliki nilai paling tinggi yaitu sebesar 91.23% dengan nilai AUC sebesar 0.686, dan nilai *F1-score* sebesar 0.2604. Nilai akurasi ini memiliki nilai yang paling tinggi dibanding pengujian sebelumnya, baik metode Naive Bayes ataupun algoritma C4.5. Hasil *confusion matrix* serta kurva ROC pada pengujian ini dapat dilihat pada Gambra 4.14 dan Gambar 4.15.

accuracy: 91.23% +/- 2.44% (micro average: 91.23%)

	true Lancar	true Macet	class precision
pred. Lancar	1279	93	93.22%
pred. Macet	32	22	40.74%
class recall	97.56%	19.13%	

Gambar 4.14 Hasil tangkapan layar *confusion matrix* algoritma C4.5 dengan lima belas *fold*s



Gambar 4.15 Grafik ROC algoritma C4.5 dengan lima belas *folds*

Dari hasi pengujian menggunakan algoritma C4.5 diperoleh nilai akurasi tertinggi sebesar 91.23% serta nilai F1-score tertinggi sebesar 0.2604 menggunakan 15 fold. Berikut adalah Tabel 4.3 yang berisi perbandingan hasil pengujian algoritma C4.5.

Tabel 4.3. Perbandingan hasil pengujian menggunakan algoritma C4.5

No.	Jumlah *Fold*	Akurasi	*Precision*	*Recall*	F1-*score*	AUC
1.	5	90.18%	0.3024	0.1809	0.2223	0.681
2.	10	90.60%	0.3095	0.1727	0.2299	0.679
3.	15	91.23%	0.4074	0.1967	0.2604	0.686

1.3.4. Perbandingan Akurasi Naive Bayes dan Algoritma C4.5

Dari hasil pengujian terhadap kedua metode, Naive Bayes memiliki nilai akurasi tertinggi pada pengujian menggunakan sepuluh dan lima belas *folds* yaitu sebesar 89.90% sedangkan algoritma C4.5 menghasilkan akurasi tertinggi sebesar 91.23% dengan menggunakan 15 *folds.* Perbandingan besar akurasi kedua metode dapat dilihat pada Tabel 4.4.

Tabel 4.4 Perbandingan performa algoritma C4.5 dan Naive Bayes

No	Metode yang digunakan	Akurasi	AUC	*F1-score*
1	Naive Bayes	89.90%	0.744	0.1000
2	Algoritma C4.5	91.23%	0.686	0.2604

Pada Tabel 4.4 dapat dilihat hasil klasifikasi antara Naive Bayes dengan algoritma C4.5 menggunakan nilai akurasi, AUC *score*, dan F1-*score*. Nilai akurasi hasil klasifikasi mengukur seberapa baik model dalam memprediksi hasil yang sebenarnya. Semakin tinggi nilai akurasi model, maka semakin baik model memprediksi hasil yang sebenarnya. Hanya saja, akurasi tertinggi bukanlah satu-satunya indikator dalam menentukan performa suatu model, karena ada faktor lain yang mempengaruhi hasil akurasi, diantaranya kemampuan model dalam memprediksi hasil data yang memiliki *outlier*, atau data yang tidak seimbang. Pada kedua metode, algoritma C4.5 memiliki nilai akurasi lebih tinggi yaitu sebesar 91.23% dibanding dengan akurasi milik Naive Bayes yaitu sebesar 89.90%.

Nilai AUC atau *Area Under ROC Curve* adalah ukuran performa suatu model klasifikasi binomial. Nilai AUC berkisar antara 0 hingga 1 dengan nilai 1 menunjukan performa model yang sempurna dalam membedakan kelas positif dan negatif sehingga semakin tinggi nilai AUC maka semakin baik model dalam mengklasifikasikan data. Pada hasil penelitian, nilai AUC milik Naive Bayes lebih tinggi yaitu sebesar 0.744 dibanding milik algoritma C.45 yaitu sebesar 0.686.

Perhitungan menggunakan nilai *F-score* juga dilakukan untuk mengevaluasi performa model. Dari hasil penelitian, nilai *F-score* algoritma C4.5 memiliki nilai yang lebih tinggi yaitu sebesar 0.2604 dibanding nilai Naive Bayes yang sebesar 0.1000. Semakin tinggi nilai F-*score* maka semakin baik performa model dalam memprediksi kelas positif dan menemukan semua kelas positif yang sebenarnya sehingga memiliki kemampuan yang lebih baik pada masalah klasifikasi. Hanya saja, nilai *F1-score* dengan nilai akurasi memiliki perbedaan yang cukup jauh dimana nilai akurasi model tinggi tetapi nilai *F1-score* rendah. Perbedaan ini terjadi karena jumlah data antara macet dan lancar yang tidak seimbang. Nilai akurasi diperoleh hanya berasal dari perhitungan nilai *True Positive* dan *True Negative*, sedangkan *F1-score* juga memperhitungkan nilai *False Negative* dan *False Positive.* Nilai F1-*score* memiliki rentang 0 hingga 1. Nilai F1-*score* adalah rasio antara *precision* dan *recall.* Namun, seperti pada akurasi dan nilai AUC, nilai F1-*score* tinggi bukanlah indikator mutlak keberhasilan model, karena masih ada faktor lain yang harus dipertimbangkan.