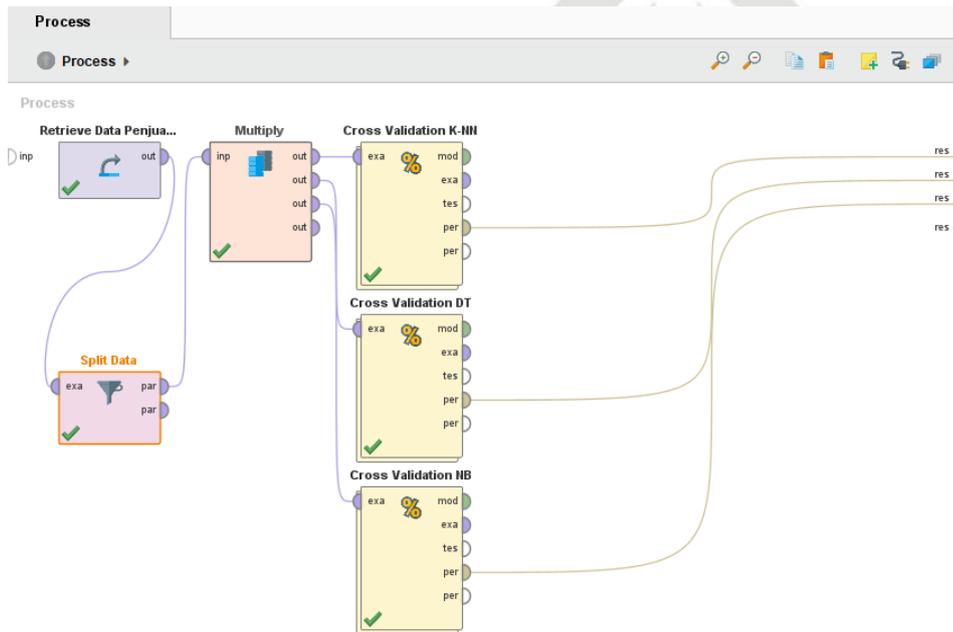


BAB IV PEMBAHASAN

4.1 Klasifikasi Algoritma

Berdasarkan aplikasi yang digunakan untuk menentukan pengujian klasifikasi data dari masing-masing algoritma akan diimplementasikan ke dalam skema RapidMiner dengan skema sebagai berikut :



Gambar 4. 1 Skema Algoritma K-NN, Decision Tree dan Naïve Bayes

Pada Gambar 4.1 menunjukkan operator yang digunakan untuk proses pengolahan data dari *retrive* yang digunakan untuk membaca data set yang telah diimport pada RapidMiner, Split data merupakan operator yang digunakan untuk membagi data menjadi dua yaitu data *training* dan data *testing*. Dalam operator tersebut digunakan pembagian 80:20. Dan operator *cross validation* yang digunakan untuk menguji akurasi dari masing-masing algoritma yang digunakan untuk pengolahan data.

4.1.1 K-Fold Cross Validation

K-Fold Cross Validation adalah salah satu dari jenis pengujian *cross validation* yang berfungsi untuk menilai kinerja proses sebuah metode algoritma

© Ha

dengan membagi sampel data secara acak dan mengelompokkan data tersebut sebanyak nilai K k-fold [47]. Pada metode *K-Fold Cross Validation*, dataset yang digunakan akan dibagi menjadi beberapa partisi secara random. Setelah dibagi ke dalam beberapa partisi maka data tersebut akan diolah sebanyak K kali percobaan dengan setiap K kali percobaan, *data testing* yang digunakan adalah data partisi ke-K dan sisa partisi yang lain digunakan sebagai *data training*. Begitu seterusnya hingga pengolahan berakhir sesuai dengan jumlah K dalam k-fold yang digunakan pada penelitian [48]. Pada penelitian ini digunakan K=10, berikut tabel *10-fold cross validation* :

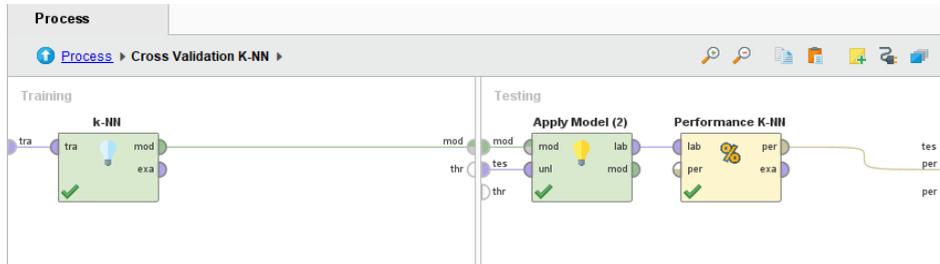
Tabel 4.1 10-Fold Cross Validation

	FOLD 1	FOLD 2	FOLD 3	FOLD 4	FOLD 5	FOLD 6	FOLD 7	FOLD 8	FOLD 9	FOLD 10
1	TEST	TRAIN								
2	TRAIN	TEST	TRAIN							
3	TRAIN	TRAIN	TEST	TRAIN						
4	TRAIN	TRAIN	TRAIN	TEST	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN
5	TRAIN	TRAIN	TRAIN	TRAIN	TEST	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN
6	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TEST	TRAIN	TRAIN	TRAIN	TRAIN
7	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TEST	TRAIN	TRAIN	TRAIN
8	TRAIN	TEST	TRAIN	TRAIN						
9	TRAIN	TEST	TRAIN							
10	TRAIN	TEST								

Pada Tabel 4.1 dapat dilihat bahwa jika 1 data menjadi *data testing* maka data lainnya menjadi *data training* dan pada *10-fold cross validation* maka iterasinya dilakukan sebanyak 10 kali untuk mendapatkan hasil akhirnya.

1. Diarangi mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

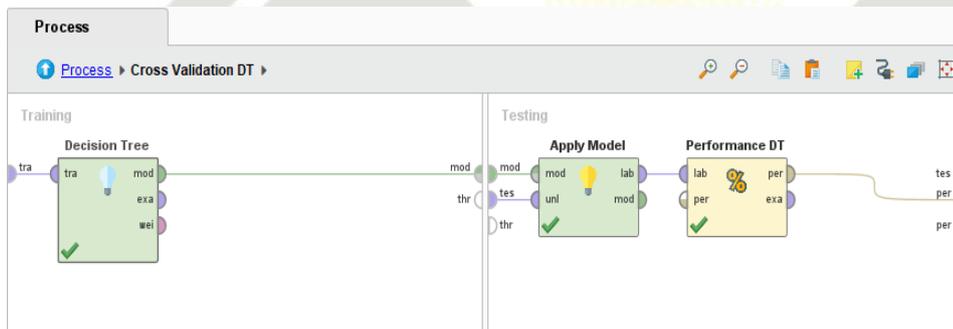
4.1.2 Cross Validation K-NN



Gambar 4. 2 Proses Cross Validation K-NN

Dapat dilihat pada Gambar 4.2 tahapan yang digunakan untuk menguji algoritma yang digunakan dan bertujuan untuk mengetahui tingkat akurasi algoritma K-NN. Terdapat operator algoritma K-NN pada kolom data *training* dan operator operator *apply model* dan *performance* pada kolom data *testing*.

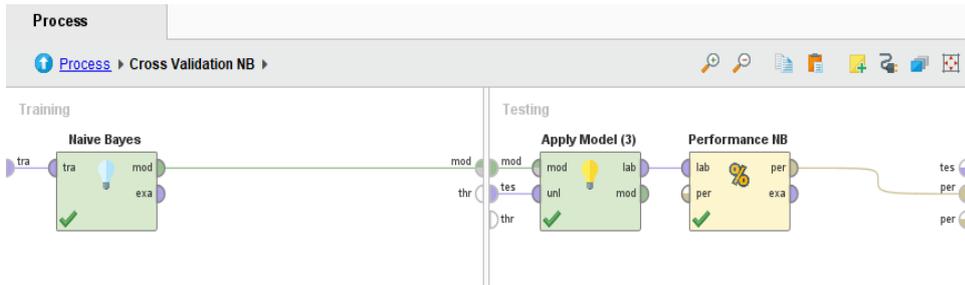
4.1.3 Cross Validation Decision Tree



Gambar 4. 3 Proses Cross Validation Decision Tree

Dapat dilihat pada Gambar 4.3 tahapan yang digunakan untuk menguji algoritma yang digunakan dan bertujuan untuk mengetahui tingkat akurasi algoritma *Decision Tree*. Terdapat operator algoritma *Decision Tree* pada kolom data *training* dan operator operator *apply model* dan *performance* pada kolom data *testing*.

4.1.4 Cross Validation Naïve Bayes



Gambar 4. 4 Proses Cross Validation Naïve Bayes

Dapat dilihat pada Gambar 4.4 tahapan yang digunakan untuk menguji algoritma yang digunakan dan bertujuan untuk mengetahui tingkat akurasi algoritma *Naïve Bayes*. Terdapat operator algoritma *Naïve Bayes* pada kolom data *training* dan operator operator *apply model* dan *performance* pada kolom data *testing*.

4.1.5 Pemilihan Model Terbaik

Untuk memilih model terbaik dapat ditentukan dari akurasi yang sudah di dapatkan dari pengolahan data menggunakan aplikasi RapidMiner. Algoritma yang digunakan pada penelitian ini adalah K-NN, *Decision Tree* dan *Naïve Bayes*, menurut [49] rumus yang digunakan untuk menentukan tingkat akurasi adalah sebagai berikut :

a. *Accuracy*

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% \quad (4.1)$$

b. *Precision*

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (4.2)$$

c. *Recall*

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (4.3)$$

d. *Error*

$$Error = \frac{(FN+FP)}{(TP+TN+FN+FP)} \times 100\% \quad (4.4)$$

Dimana : TP = *True Positive*, TN = *True Negative*, FP = *False Positive*, FN = *False Negative*

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Dari pengolahan data yang sudah dilakukan dari masing-masing algoritma di dapatkan akurasi sebagai berikut :

a) **K-Nearest Neighbor (K-NN)**

accuracy: 95.78% +/- 4.85% (micro average: 95.76%)

	true Kurang Laris	true Sangat Laris	true Laris	class precision
pred. Kurang Laris	203	1	4	97.60%
pred. Sangat Laris	3	12	0	80.00%
pred. Laris	0	2	11	84.62%
class recall	98.54%	80.00%	73.33%	

Gambar 4.5 Nilai Akurasi Algoritma K-NN

Pada Gambar 4.5 di atas dapat dilihat bahwa akurasi dari algoritma K-NN pada RapidMiner adalah 95.78% dengan *class precision* untuk pred. kurang laris adalah 97.50% pred. sangat laris adalah 80.00% dan pred. laris adalah 84.62%. Selain itu juga terdapat *class recall true* kurang laris adalah 98.54% *true* sangat laris adalah 80.00% dan *true* laris adalah 73.33%. Dengan nilai *error* nya adalah 4.24%.

Berdasarkan Gambar 4.5 nilai yang di dapatkan hasil akurasi, *class precision*, *class recall* dan *error* pada rapidminer seperti yang sudah disebutkan di atas. Selain di uji dengan *software* rapidminer, dapat juga di uji dengan rumus (4.1), (4.2), (4.3) dan (4.4) untuk menghitung dengan cara manual yaitu :

a) *Accuracy*

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\%$$

$$= \frac{((203+12+11)+(23+215+214))}{((203+12+11)+(5+3+2)+(5+7+8)+(23+215+214))} \times 100\%$$

$$= 0.9578 \times 100\%$$

$$Accuracy = 95.78\%$$

b) *Class precision*

- *Class precision* pred. kurang laris = $\frac{TP}{TP+FP} \times 100\%$

$$= \frac{203}{203+4+1} \times 100\%$$

$$= 0.97596 \times 100\%$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Class precision pred. kurang laris = 97.60%

$$\begin{aligned} \bullet \text{ Class precision pred. sangat laris} &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{12}{12+3} \times 100\% \\ &= 0.8 \times 100\% \end{aligned}$$

Class precision pred. sangat laris = 80.00%

$$\begin{aligned} \bullet \text{ Class precision pred. laris} &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{11}{11+2} \times 100\% \\ &= 0.8462 \times 100\% \end{aligned}$$

Class precision pred. laris = 84.62%

c) *Class recall*

$$\begin{aligned} \bullet \text{ Class recall true kurang laris} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{203}{203+3} \times 100\% \\ &= 0.9854 \times 100\% \end{aligned}$$

Class recall true kurang laris = 98.54%

$$\begin{aligned} \bullet \text{ Class recall true sangat laris} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{12}{12+1+2} \times 100\% \\ &= 0.8 \times 100\% \end{aligned}$$

Class recall true sangat laris = 80.00%

$$\begin{aligned} \bullet \text{ Class recall true laris} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{11}{11+4} \times 100\% \\ &= 0.7333 \times 100\% \end{aligned}$$

Class recall true laris = 73.33%

d) *Error*

$$\begin{aligned} \text{Error} &= \frac{(FN+FP)}{(TP+TN+FN+FP)} \times 100\% \\ &= \frac{((5+7+8)+(5+3+2))}{((203+12+11)+(23+215+214)+(5+7+8)+(5+3+2))} \times 100\% \\ &= 0.0424 \times 100\% \end{aligned}$$

Error = 4.24%

b. Decision Tree

accuracy: 95.76% +/- 2.84% (micro average: 95.76%)

	true Kurang Laris	true Sangat Laris	true Laris	class precision
pred. Kurang Laris	206	1	3	98.10%
pred. Sangat Laris	0	9	1	90.00%
pred. Laris	0	5	11	68.75%
class recall	100.00%	60.00%	73.33%	

Gambar 4. 6 Nilai Akurasi Algoritma *Decision Tree*

Pada Gambar 4.6 di atas dapat dilihat bahwa akurasi dari algoritma *Decision Tree* pada RapidMiner adalah 95.76% dengan *class precision* untuk pred. kurang laris adalah 98.10% pred. sangat laris adalah 90.00% dan pred. laris adalah 68.75%. Selain itu juga terdapat *class recall true* kurang laris adalah 100.00% *true* sangat laris adalah 60.00% dan *true* laris adalah 73.33%. Dengan nilai *error* nya adalah 4.36%.

Berdasarkan Gambar 4.6 nilai yang di dapatkan hasil akurasi, *class precision* dan *class recall* pada rapidminer seperti yang sudah disebutkan di atas. Selain di uji dengan *software* rapidminer, dapat juga di uji dengan rumus (4.1), (4.2), (4.3) dan (4.4) untuk menghitung dengan cara manual yaitu :

a. *Accuracy*

$$\begin{aligned}
 Accuracy &= \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% \\
 &= \frac{((206+9+11)+(20+215+217))}{((206+9+11)+(4+1+5)+(6+9+5)+(20+215+217))} \times 100\% \\
 &= 0.9576 \times 100\%
 \end{aligned}$$

$$Accuracy = 95.76\%$$

b. *Class precision*

- $$\begin{aligned}
 \text{Class precision pred. kurang laris} &= \frac{TP}{TP+FP} \times 100\% \\
 &= \frac{206}{206+1+3} \times 100\% \\
 &= 0.9810 \times 100\%
 \end{aligned}$$

$$\text{Class precision pred. kurang laris} = 98.10\%$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- $$\begin{aligned} \text{Class precision pred. sangat laris} &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{9}{9+1} \times 100\% \\ &= 0.9 \times 100\% \end{aligned}$$

$$\text{Class precision pred. sangat laris} = 90.00\%$$

- $$\begin{aligned} \text{Class precision pred. laris} &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{11}{11+5} \times 100\% \\ &= 0.6875 \times 100\% \end{aligned}$$

$$\text{Class precision pred. laris} = 68.75\%$$

c. *Class recall*

- $$\begin{aligned} \text{Class recall true kurang laris} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{206}{206} \times 100\% \\ &= 1 \times 100\% \end{aligned}$$

$$\text{Class recall true kurang laris} = 100.00\%$$

- $$\begin{aligned} \text{Class recall true sangat laris} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{9}{9+1+5} \times 100\% \\ &= 0.6 \times 100\% \end{aligned}$$

$$\text{Class recall true sangat laris} = 60.00\%$$

- $$\begin{aligned} \text{Class recall true laris} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{11}{11+3+1} \times 100\% \\ &= 0.7333 \times 100\% \end{aligned}$$

$$\text{Class recall true laris} = 73.33\%$$

d. *Error*

$$\begin{aligned} \text{Error} &= \frac{(FN+FP)}{(TP+TN+FN+FP)} \times 100\% \\ &= \frac{((6+9+5)+(4+1+5))}{((206+9+11)+(20+215+217)+(6+9+5)+(4+1+5))} \times 100\% \\ &= 0.0436 \times 100\% \end{aligned}$$

$$\text{Error} = 4.36\%$$

Naïve Bayes

accuracy: 91.92% +/- 4.70% (micro average: 91.95%)

	true Kurang Laris	true Sangat Laris	true Laris	class precision
pred. Kurang Laris	200	8	5	93.90%
pred. Sangat Laris	0	7	0	100.00%
pred. Laris	6	0	10	62.50%
class recall	97.09%	46.67%	66.67%	

Gambar 4. 7 Nilai Akurasi Algoritma Naïve Bayes

Pada Gambar 4.7 di atas dapat dilihat bahwa akurasi dari algoritma *Naïve Tree* pada RapidMiner adalah 91.92% dengan *class precision* untuk pred. kurang laris adalah 93.90% pred. sangat laris adalah 100.00% dan pred. laris adalah 62.50%. Selain itu juga terdapat *class recall true* kurang laris adalah 97.09% *true* sangat laris adalah 46.67% dan *true* laris adalah 66.67%. Dengan nilai *error* nya 8.05%.

Berdasarkan Gambar 4.7 nilai yang di dapatkan hasil akurasi, *class precision* dan *class recall* pada rapidminer seperti yang sudah disebutkan di atas. Selain di uji dengan *software* rapidminer, dapat juga di uji dengan rumus (4.1), (4.2), (4.3) dan (4.4) untuk menghitung dengan cara manual yaitu :

a. *Accuracy*

$$\begin{aligned}
 Accuracy &= \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% \\
 &= \frac{((200+7+10)+(17+210+207))}{((200+7+10)+(13+0+6)+(6+19+13)+(17+210+207))} \times 100\% \\
 &= 0.9192 \times 100\%
 \end{aligned}$$

$$Accuracy = 91.92\%$$

b. *Class precision*

- $$\begin{aligned}
 \text{Class precision pred. kurang laris} &= \frac{TP}{TP+FP} \times 100\% \\
 &= \frac{200}{200+8+5} \times 100\% \\
 &= 0.9390 \times 100\%
 \end{aligned}$$

$$\text{Class precision pred. kurang laris} = 93.90\%$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- $Class\ precision\ pred.\ sangat\ laris = \frac{TP}{TP+FP} \times 100\%$
 $= \frac{7}{7} \times 100\%$
 $= 1 \times 100\%$

$Class\ precision\ pred.\ sangat\ laris = 100.00\%$

- $Class\ precision\ pred.\ laris = \frac{TP}{TP+FP} \times 100\%$
 $= \frac{10}{10+6} \times 100\%$
 $= 0.625 \times 100\%$

$Class\ precision\ pred.\ laris = 62.50\%$

c. *Class recall*

- $Class\ recall\ true\ kurang\ laris = \frac{TP}{TP+FN} \times 100\%$
 $= \frac{200}{200+6} \times 100\%$
 $= 0.9709 \times 100\%$

$Class\ recall\ true\ kurang\ laris = 97.09\%$

- $Class\ recall\ true\ sangat\ laris = \frac{TP}{TP+FN} \times 100\%$
 $= \frac{7}{7+8} \times 100\%$
 $= 0.4667 \times 100\%$

$Class\ recall\ true\ sangat\ laris = 46.67\%$

- $Class\ recall\ true\ laris = \frac{TP}{TP+FN} \times 100\%$
 $= \frac{10}{10+5} \times 100\%$
 $= 0.6667 \times 100\%$

$Class\ recall\ true\ laris = 66.67\%$

d. *Error*

$$Error = \frac{(FN+FP)}{(TP+TN+FN+FP)} \times 100\%$$

$$= \frac{((6+19+13)+(13+0+6))}{((200+7+10)+(17+210+207)+(6+19+13)+(13+0+6))} \times 100\%$$

$$= 0.0805 \times 100\%$$

$Error = 8.05\%$

4.1.6 Analisis Hasil Pengujian

Berdasarkan pengujian yang dilakukan terhadap algoritma K-NN, *Decision Tree* dan *Naïve Bayes* dengan menggunakan *tools* Rapidminer untuk menguji algoritma-algoritma tersebut sehingga menghasilkan perbandingan dari ketiga algoritma tersebut yang dapat dilihat pada tabel 4.8.

Tabel 4. 2 Hasil Pengujian Algoritma

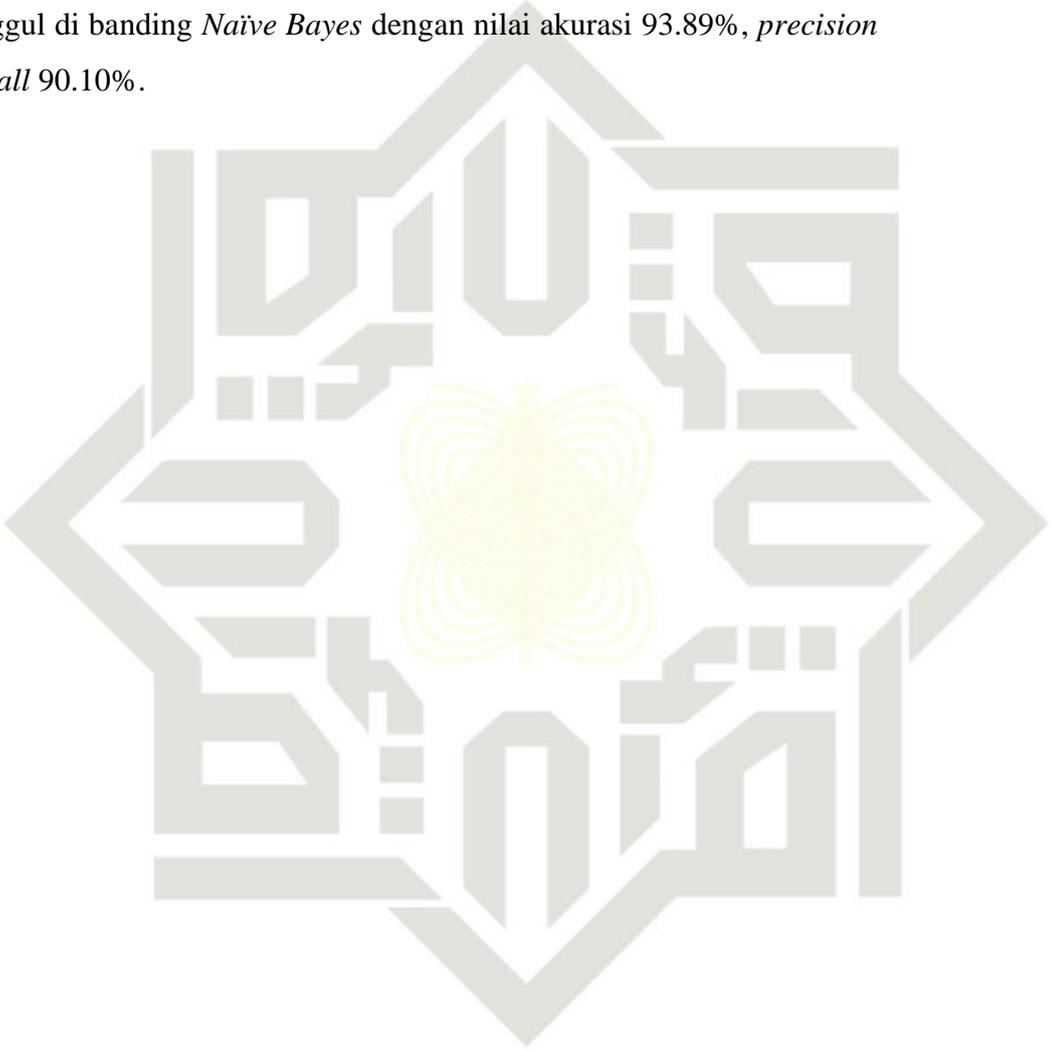
Algoritma	Tingkat Akurasi	Error
K-NN	95.78%	4.24%
<i>Decision Tree</i>	95.76%	4.36%
<i>Naïve Bayes</i>	91.92%	8.05%

Tabel 4.2 menunjukkan hasil pengujian dari metode yang digunakan. Pengujian terhadap algoritma-algoritma yang tersebut untuk analisis tingkat penjualan produk pada penelitian ini dapat dilihat bahwa tingkat akurasi untuk algoritma K-NN adalah 95.78% dengan nilai *error* sebesar 4.24% dan *class precision* untuk pred. kurang laris adalah 97.60%, pred. sangat laris adalah 80.00%, pred. laris adalah 84.62%. Selain itu juga terdapat *class recall* untuk *true* kurang laris 98.54%, *true* sangat laris 80.00% dan *true* laris 73.33%, algoritma *Decision Tree* adalah 95.76% dengan nilai *error* sebesar 4.36% dan *class precision* untuk pred. kurang laris adalah 98.10%, pred. sangat laris adalah 90.00%, pred. laris adalah 68.75%. Selain itu juga terdapat *class recall* untuk *true* kurang laris 100.00%, *true* sangat laris 60.00% dan *true* laris 73.33%, algoritma *Naïve Bayes* adalah 91.92% dengan nilai *error* sebesar 8.05% dan *class precision* untuk pred. kurang laris adalah 93.90%, pred. sangat laris adalah 100.00%, pred. laris adalah 67.50%. Selain itu juga terdapat *class recall* untuk *true* kurang laris 97.09%, *true* sangat laris 46.67% dan *true* laris 66.67%. Dari hasil pengujian tersebut dapat dilihat bahwa algoritma K-NN memiliki tingkat akurasi lebih besar dibandingkan algoritma lainnya.

4.1.7 Diskusi

Dari hasil penelitian yang sudah dilakukan algoritma K-NN lebih unggul dibandingkan algoritma yang lain seperti *Decision Tree* dan *Naïve Bayes*. Seperti penelitian yang dilakukan oleh [50] didapatkan hasil dan kesimpulan bahwa algoritma K-NN lebih akurat dibandingkan *Decision Tree* dengan hasil nilai akurasi

98.18%. Lalu juga pada penelitian yang dilakukan oleh [51] dari penelitian tersebut juga didapatkan hasil serta kesimpulan bahwa algoritma K-NN lebih akurat di bandingkan algoritma *Decision Tree* dan *Naïve Bayes* dengan nilai akurasi nya yaitu 91.26%, *precision* 92.97%, *recall* 91.26%, dan *overall performance* 91.83%. Serta penelitian yang dilakukan oleh [52] juga di dapatkan hasil bahwa algoritma K-NN lebih unggul di banding *Naïve Bayes* dengan nilai akurasi 93.89%, *precision* 86.90% dan *recall* 90.10%.



UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.