Cost Sensitive Tree dan *Naïve Bayes* Pada Klasifikasi *Multiclass*

M. Aldiki Febriantono[1], Ridho Herasmara[2], Gusti Pangestu[3]

[1,3] Computer Science, Fakultas Computer Science, Universitas Bina Nusantara
[2]Teknik Elektro, Fakultas Saintek, Universitas Islam Raden Rahmat
[1]m.aldiki@binus.ac.id, [2] ridho.herasmara@uniramalang.ac.id, [3] Gusti.pangestu@binus.ac.id

Abstrak

Data mining merupakan proses pengolahan data untuk mengambil keputusan secara cepat, tepat dan akurat. Data mining pada bidang kesehatan dan manufacturing menjadi hal yang sangat penting dikarenakan suatu kesalahan klasifikasi (*misclassification*) akan memiliki dampak serius. Masalah utama pada data mining ketika data yang digunakan bersifat *imbalanced multiclass* karena *classifier* kesulitan untuk mengklasifikasikan data sehingga menyebabkan terjadinya *misclassification*. Solusi untuk meminimalkan *missclasification* dengan menggunakan metode *cost sensitive* pada *classifier decision tree* C5.0 dan *naïve bayes*. Penelitian ini menggunakan dataset *glass*, *lympografi*, *vehicle*, *thyroid* dan *wine* yang diperoleh dari UCI *Respository*. Kelima dataset dilakukan proses seleksi atribut menggunakan *particle swarm optimazation*. Kemudian dataset diuji menggunakan metode *cost sensitive decision tree* C5.0 dan *cost sensitive naïve bayes*. Hasil pengujian menggunakan metode *cost sensitive decision tree* C5.0 diperoleh nilai *accuracy* pada dataset *glass*, *lympografi*, *vehicle*, *thyroid* dan *wine* berturut-turut sebesar 76.17%, 83.33%, 75.27%, 95.81% dan 95.83%. Sedangkan metode *cost sensitive naïve bayes* memiliki performa *accuracy* pada dataset berturut-turut sebesar 32.24%, 82.61%, 25.53%, 97.67% dan 94.94%.

Kata kunci : c*ost sensitive, decision tree, multiclass classification, naïve bayes.*

1. Pendahuluan

Larose (2005), menyatakan *Data mining* merupakan proses penggalian data atau pencarian pola dengan tujuan mendapatkan informasi sebagai pengetahuan untuk mengambil keputusan secara cepat, tepat dan akurat di waktu yang akan datang. Kemampuan mengambil keputusan secara cepat, tepat dan akurat pada bidang kesehatan dan *manufacturing* menjadi hal yang sangat penting. Kesalahan mengambil keputusan pada bidang kesehatan dapat berakibat fatal pada kehidupan pasien sedangkan kesalahan pada bidang *manufacturing* berdampak pada hasil produksi maka dari itu perlu adanya *data mining*. Pola data dapat dipelajari sebagai dasar pengambilan keputusan. Patel, et. Al. (2012), menyatakan salah satu cara untuk mencari pola data adalah dengan menggunakan teknik klasifikasi. Teknik tersebut menggunakan nilai fitur dari data masukan (*atribut*) dengan tujuan mengenali pola berupa prediksi *class*. Menurut Ali, et. al. (2019), berdasarkan hasil prediksinya, klasifikasi dibagi menjadi dua, yaitu klasifikasi *binary class*, dan klasifikasi *multiclass*.

Machine learning merupakan bagian dari *data mining* untuk membantu mencari pola data secara otomatis. Bernard & Adam (2015) menyatakan pendistribusian data yang tidak sama (*imbalanced data*) menjadi masalah karena *machine learning* lebih fokus pada *class* yang dominan (*mayority*) dibandingkan dengan *class* yang sedikit (*minority*) dikarenakan *minority class* memiliki jumlah data pelatihan dan pengujian yang lebih sedikit dari pada *mayority class*. Padahal *minority class* dapat memiliki pengaruh yang jauh lebih besar jika terjadi salah klasifikasi (*misclassification*), berdasarkan Wang & Yao (2012). Pada kasus terjadinya salah klasifikasi sebuah kanker ganas sebagai kanker jinak, dapat berpengaruh sangat buruk bagi Kesehatan pasien, dibandingkan ketika terjadi kesalahan klasifikasi kanker jinak sebagai kanker ganas. Beberapa metode *data mining* yang digunakan untuk menyelesaikan permasalahan pada data *imbalanced* adalah *decision tree* dan *naïve bayes.*

Konsep *decision tree* adalah merubah data tabel menjadi model pohon kemudian menghasilkan aturan keputusan (*rule*), menurut Jauhari & Supianto (2019). *Decision tree* memerlukan algoritma untuk membuat model pohon keputusan diantaranya ID3, C4.5 dan C5.0. Algoritma C5.0 memiliki berbagai kelebihan dibandingkan dengan ID3 dan C4.5 diantaranya memiliki waktu komputasi yang cepat, mampu menangani data *continue*, *categorical* dan *missing value*, berdasarkan Patel & Rana (2014). Mirip dengan *decision tree*, *naïve bayes* juga memiliki beberapa kelebihan seperti mudah digunakan, waktu komputasi yang cepat, mampu menanggani *missing value* dan data *continue* (Faisal & Mofizur, 2011). Pendekatan-pendekatan ini lebih baik ketika dibandingkan pendekatan *neural network* yang menurut Herasmara, et. al (2019),

membutuhkan waktu dan sumberdaya komputasi yang tinggi karena desain yang cenderung tidak efisien.

Chai, et. Al (2014) berargumen bahwa *Cost sensitive learning* merupakan metode yang digunakan untuk meningkatkan performa *classifier* dengan meminimalkan *misclassification cost*. Pemilihan atribut data juga memiliki peranan yang penting untuk mendapatkan performa yang maksimal. Menurut Wei, et. al. (2008), *Particle swarm optimization* merupakan algoritma yang digunakan untuk menyeleksi atribut berdasarkan nilai bobot dari setiap atribut.

Haldankar (2016) menuturkan beberapa metode yang telah dikembangkan pada beberapa penelitian dengan menggunakan *cost sensitive classifier*, diantaranya adalah *cost sensitive using random forest* dan *selection boruta* pada data *imbalanced* nasabah bank. Hasil accuracy dengan menggunakan metode tersebut sebesar 76%.

Penelitian lainnya oleh Daraei (2017), menggunakan metode *cost sensitive decision tree C4.5* dan *selection genetic algorithm* pada data pasien jantung. Hasil accuracy dari metode tersebut sebesar 82.67%. Penelitian lainnya oleh Xiangju, et. al. (2015) menggunakan metode *cost sensitive decision tree C4.5 using probabilistic mechanism* pada delapan *dataset imbalanced*. Metode tersebut mampu meminimalkan *cost* sebesar 10% pada semua *dataset*.

Penelitian ini bertujuan untuk membandingan performa metode *cost sensitive decision tree* C5.0 dan *cost sensitive naïve bayes* sebagai solusi untuk mengambil keputusan secara cepat, tepat dan akurat pada klasifikasi data *imbalanced multiclass.* Performa *classifer* diukur menggunakan parameter *accuracy, recall, precision, f-measure* dan total *cost.*

2. Pendekatan Dalam *Data Mining*

A. Data Mining

Larose (2005) menyatakan bahwa *data mining* merupakan proses penggalian data atau pencarian pola dengan tujuan mendapatkan informasi sebagai pengetahuan untuk mengambil keputusan secara cepat, tepat dan akurat di waktu yang akan datang *Data mining* memiliki beberapa teknik untuk mengenali pola antara lain deskripsi, estimasi, prediksi, klasifikasi, c*lustering* dan asosiasi. Patel, et. al. (2012) menyatakan klasifikasi merupakan proses untuk memprediksi *class* berdasarkan atribut data ke dalam label *class* yang telah didefinisikan sebelumnya. Zhang, et. al. (2010) menjabarkan tahapan yang dilakukan pada proses data mining adalah pembersihan data *(data cleaning)*, seleksi data *(data selection)*, transformasi data *(data transformation)*, proses *data mining*, seleksi atribut *(atribut selection)* dan evaluasi pola *(pattern evaluation)*

B. Particle Swarm Optimazation

Pemilihan atribut menjadi faktor penting untuk mendapatkan data yang berkualitas pada proses *data mining*, berdasarkan Xue, et. al. (2013). Atribut yang tidak relevan dapat dihilangkan sehingga dapat menghasilkan sebuah informasi yang tepat dan akurat. Salah satu algoritma yang digunakan untuk menyeleksi atribut adalah *particle swarm optimization* (PSO).

PSO merupakan sebuah teknik optimasi untuk menyeleksi atribut berdasarkan nilai bobot atribut (w). Atribut yang relevan memiliki nilai bobot 1, atribut yang tidak relevan bernilai 0, sementara atribut yang memiliki relevansi parsial akan memiliki nilai bobot atribut pada rentang 0 < w < 1. Kelebihan PSO adalah mempunyai konsep sederhana, mudah diimplementasikan, dan efisien dalam perhitungan jika dibandingkan dengan teknik seleksi atribut lainnya. Wei, et. al. (2008) memaparkan dua parameter yang digunakan untuk menentukan bobot atribut yaitu kecepatan dan posisi. Persamaan untuk menetukan kecepatan atribut *i* pada *d* dimensi ini ditunjukkan dalam Persamaan 1.

$$V_{id}^{k+1} = w \times V_{id}^{k} + c1 \times r1 \times \left(P_{id} - X_{id}\right) + c2 \times r2 \times \left(G_{id} - X_{id}\right) \quad (1)$$

Persamaan untuk menentukan posisi atribut *i* pada *d* dimensi ditunjukkan dalam Persamaan 2.

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \quad (2)$$

Keterangan:

V_{id} = Kecepatan individu ke *i* pada *d* dimensi
X_{id} = Posisi individu *i* pada *d* dimensi
ω = Parameter inertia *weight*
$c_1 c_.$ = *Learning rate*, nilainya antara 0 dan 1
$r_{1,2}$ = Paraeter random antara 0 dan 1
P_{id} = *Pbest* (*local best*) individu *i* pada *d* dimensi
G_{id} = *Gbest* (*global best*) pada *d* dimensi

C. Algoritma C5.0

Algoritma C5.0 digunakan untuk membuat pola pohon keputusan (*decision tree*) berdasarkan nilai *entropy* dan *information gain.* Thomas & Joy (2006) memaparkan tentang *Entropy* (S) sebagai jumlah bit yang dibutuhkan untuk dapat mengekstrak suatu *class* dari jumlah data acak pada ruang sample S. *Entropy* biasa digunakan sebagai suatu parameter untuk mengukur heterogenitas (keberagaman) dari suatu kumpulan sampel data. Jika kumpulan sampel data semakin heterogen, maka semakin besar nilai *entropy* dan nilai *gain* semakin kecil. Secara matematis *entropy* dirumuskan dalam Persamaan 3.

$$H(S) = \sum_{i=1}^{n} -p(s_i)\log_2(p(s_i)) \quad (3)$$

Setelah mendapatkan nilai *entropy* selanjutnya mencari nilai *information gain* berdasarkan Pandya & Pandya (2015). *Information gain* digunakan untuk mengukur efektivitas karakteristik atribut dalam mengklasifikasikan *class*. Persamaan 4 digunakan untuk menghitung *information gain* sebagai berikut.

$$IG(S, A_i) = H(S) - \sum_{\alpha \in A_i}^{n} \frac{|S_\alpha|}{|S|} H(S_\alpha) \quad (4)$$

Keterangan:
H = *Entropy*
S = Himpunan kasus
A = Atribut
N = Jumlah partisi atribut A
$|S_\alpha|$ = Jumlah kasus pada partisi ke-i
$|S|$ = Jumlah kasus dalam S
p_i = Proporsi dari S_i terhadap S

D. Metacost

Metacost merupakan algoritma dari metode *cost sensitive learning* dengan menggunakan teknik *thresholding meta learning* untuk meminimalkan *cost.* Prisip kerja dari *metacost* adalah menghitung probabilitas setiap class *j* pada label *class* (S_j) dari model *decision tree* (M_i). Persamaan untuk menghitung probabilitas ditunjukkan pada Persamaan 5.

$$P(j \mid x) = \frac{1}{\sum_i 1} \sum_i P(j \mid x, M_i) \quad (5)$$

Cost dinotasikan sebagai $C_{(i,j)}$, dimana *i* adalah aktual *class* tetapi diprediksi menjadi *class j* sehingga menyebabkan *misclassification*. Ketika probabilitas pada label *class* $P(j \mid x, M_i) > 1$, maka akan dilakukan evaluasi dengan cara melakukan teknik *pruning* dan *relabeling* sampai mendapatkan minimum *cost.* Persamaan untuk mencari nilai minimum *cost* berdasarkan Domingos (1999) ditunjukkan pada Persamaan 6.

$$S_i = \arg\min_i \sum_j P(j \mid x) C(i, j) \quad (6)$$

E. Naïve Bayes

Janasthar & Hanskunatai (2014) memaparkan *Naive Bayes* sebagai metode klasifikasi yang mengunakan teorema *Bayes* dengan mengansumsikan atribut secara *independen* atau tidak saling ketergantungan. Teorema *bayes* bekerja dengan memperbaiki atau mengevaluasi probabilitas awal (*prior probability)* menjadi probabilitas baru (*posterior probability*) sesuai pemaparan Friedman, Geiger, & Goldezmidt (1997). Faisal & Mofizur (2011) sendiri menyatakan bahwa keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*training data*) yang kecil untuk menentukan *estimasi* paremeter yang diperlukan dalam proses pengklasifikasian. Persamaan umum dari *Naïve Bayes* ditunjukkan pada Persamaan 7.

$$\mathrm{P}(C_i \mid A_{ij}) = \frac{\mathrm{P}(C_i)\mathrm{P}(A_{ij} \mid C_i)}{\mathrm{P}(A_{ij})} \quad (7)$$

Keterangan:
C_i = Data dengan *class* yang belum diketahui
A_{ij} = Hipotesis data merupakan suatu *class* tertentu
$P(C_i|A_{ij})$ = Probabilitas hipotesis C_i berdasar kondisi A_{ij} (*posteriori probabilitas)*
$P(C_i)$ = Probabilitas hipotesis C_i (*prior probabilitas)*
$P(A_{ij}|C_i)$ = Probabilitas A_{ij} berdasarkan kondisi pada hipotesis C_i
$P(A_{ij})$ = Probabilitas A_{ij}

F. Cost sensitine Naïve bayes

Chai, et. al. (2004) memaparkan *Cost sensitive learning* sebagai metode yang digunakan untuk meminimalkan *misclassification cost* atau kesalahan klasifikasi pada model *classifier*. *Cost function* merupakan nilai kesalahan dalam klasifikasi *class*. *Cost function* dinotasikan $C_{(i,j)}$ dimana aktual *class i* diklasifikasikan menjadi *class j*. Pada *naïve bayes, misclassification cost* disebut juga *conditional risk* $R(c_j \mid x)$ yang dinyatakan dalam Persamaan 8.

$$R(c_j \mid x) = \sum_i C_{ij} x P(c_i \mid x) \quad (8)$$

Dimana $P(c_i \mid x)$ adalah probabilitas *posterior* pada *classifier naïve bayes.* Untuk mendapatkan minimal *conditional risk,* digunakan Persamaan 9.

$$R(c_j \mid x) = \min_j R(c_j \mid x) \quad (9)$$

3. Klasifikasi Dengan Pendekatan *Cost Sensitive*

Penelitian ini memerlukan alat pengujian antara lain microsoft excel 2013 digunakan untuk mengolah data klasifikasi, rapid miner versi 5.3.0 digunakan untuk merancang sistem klasifikasi sedangkan untuk validasi hasil klasifikasi menggunakan MATLAB 2017.

Dataset pengujian berasal dari *University of California Irvine* (UCI) *machine learning repository* dengan url http://archive.ics.uci.edu/ml. *Dataset* yang digunakan dalam penelitian ini terdiri atas data klasifikasi *multiclass*. Deskripsi *dataset* yang digunakan, ditunjukkan pada Tabel 1.

Tabel 1. Deskripsi *Dataset* Penelitian

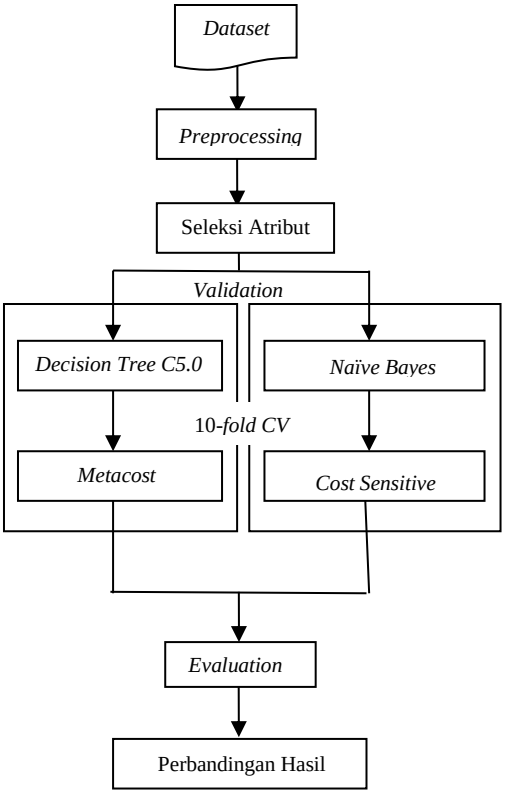
No	*Dataset*	Jumlah *Instances*	Jumlah *Atribut*	Jumlah *Class*
1	*Glass*	214	10	6
2	*Lympografi*	148	18	4
3	*Vehicle*	946	18	4
4	*Thyroid*	215	5	3
5	*Wine*	178	13	3

Dataset Glass adalah kumpulan data terkait klasifikasi *glass* (kaca) memilki data jenis real. *Dataset Lympografi* adalah Data yang memanfaatkan teknologi x-ray untuk melihat sirkulasi limfatik dan kelenjar getah bening untuk tujuan diagnosis memiliki jenis data interger. *Dataset Vehicle* adalah data mengenai klasifikasi jenis kendaraan berdasarkan sudut pandang gambar yang berbeda memiliki jenis data interger. *Dataset Thyroid* adalah kanker kelenjar pada leher. Data ini digunakan untuk memprediksi pasien *tiroid* berdasarkan *class eutiroidisme, hipotiroidisme* atau *hipertiroidisme* memiliki jenis data numeric. *Dataset Wine* adalah hasil analisis kimia dari anggur yang ditanam pada wilayah yang sama di Italia tetapi berasal dari tiga kultivar yang berbeda memiliki jenis data real.

Pemprosesan awal *dataset* dengan menghapus data *outlier* dan menggisi data *missing value*. Penghapusan data *outlier* dilakukan dengan menghapus data-data yang memiliki simpangan cukup jauh, namun ketika dilakukan pengujian, tidak terjadi perubahan accuracy yang signifikan. Pengisian *missing value* dilakukan dengan mengisikan nilai rata-rata dari atribut tersebut.

Setelah dilakukan pemprosesan awal, kemudian dilakukan seleksi atribut. Seleksi atribut ini dilakukan menggunakan algoritma *particle swarm optimization* (PSO). *Dataset* yang telah diseleksi menggunakan PSO kemudian disiapkan untuk proses validasi data, dengan menggunakan metode 10-*fold cross validation* dimana *dataset* hasil seleksi ini dibagi menjadi 90% data *training* dan 10% data *testing*.

Selanjutnya data *training* dilakukan pembelajaran untuk memperoleh pola model kemudian pola tersebut dilakukan pengujian menggunakan data *testing* dengan menggunakan metode *cost sensitive decision tree* C5.0 dan *cost sensitive naïve bayes*. Hasil pengujian kemudian dievaluasi menggunakan parameter *accuracy*, *recall*, *precision*, *f-measure* dan *total cost*. Kerangka proses klasifikasi ditampilkan pada Gambar 1.



Gambar 1. *Flowchart* Proses Klasifikasi

A. Konsep particle swarm optimazation

Particle swarm optimization (PSO) merupakan algoritma yang digunakan pada penelitian ini untuk proses seleksi atribut. Konsep dari algoritma PSO adalah memilih atribut yang relevan berdasarkan nilai bobot (w). Tahapan dalam proses *particle swarm optimization* (PSO) adalah sebagai berikut:

Step 1. Memasukkan *dataset.*
Step 2. Asumsikan bahwa kecepatan dan posisi awal ditentukan secara *random* (acak).
Step 3. Hitung kecepatan atribut *i* pada *d* dimensi menggunakan persamaan (1).
Step 4. Menghitung posisi atribut *i* pada *d* dimensi menggunakan persamaan (2).
Step 5. Melakukan evaluasi *Pbest* dan *Gbest* untuk mendapatkan posisi dan kecepatan paling maksimal pada setiap atribut.
Step 6. Evaluasi atribut berdasarkan nilai bobot yang mendekati 0 atau bernilai 0.
Step 7. Atribut yang tidak relevan dapat dihilangkan.

B. Konsep *cost sensitive decision C5.0*

Tahap awal algoritma C5.0 digunakan untuk membuat pola model pohon keputusan berdasarkan nilai *entropy* dan *information gain*. Selanjutnya hasil dari pola tersebut dilakukan evaluasi untuk mencari minimum *cost* dengan menggunakan algoritma

metacost. Konsep dari metode *cost sensitive decision tree* C5.0 sebagai berikut:

Step 1. Memasukkan *dataset.*

Step 2. Menghitung nilai *entropy* total *dataset* menggunakan persamaan (3).

Step 3. Menghitung nilai *entropy* dan *gain* pada setiap atribut menggunakan persamaan (3) dan (4).

Step 4. Menentukan node akar berdasarkan nilai *gain* terbesar.

Step 5. Menentukan *internal node* hingga menghasilkan *leaf node* berdasarkan nilai *entropy* dan *gain*.

Step 6. Proses berhenti jika atribut telah digunakan semua.

Step 7. Menghitung nilai probabilitas setiap *leaf node* menggunakan persamaan (5). Jika probabilitas > 1 maka akan dilakukan evaluasi menggunakan teknik *pruning* dan *relabel.*

Step 8. Mencari pola model *decision tree* dengan *minimum cost* menggunakan persamaan (6).

C. Konsep *Cost sensitive Naïve Bayes*

Tahap awal *naïve bayes* menemukan probabilitas *prior* selanjutnya mencari *class* probabilitas kondisional sehingga mendapatkan probabilitas *posterior* kemudian dilakukan evaluasi menggunakan *misclassification costs* untuk memperoleh minimum *cost* menggunakan *conditional risk*. Konsep dari metode *cost sensitive naïve bayes* sebagai berikut:

Step 1. Memasukkan *dataset.*

Step 2. Menghitung jumlah *class/label.*

Step 3. Menghitung jumlah kasus yang sama dengan *class* yang sama menggunakan persamaan (7).

Step 4. Kalikan semua hasil kriteria pada data.

Step5. Bandingkan hasil *class* sebagai prediksi *class.*

Step 6. Menghitung *conditional risk* menggunakan persamaan (8).

Step 7. Mencari pola model *naïve bayes* dengan *minimum cost* menggunakan persamaan (9).

D. Metode Pengujian

Mengukur performa sistem klasifikasi merupakan hal yang penting dalam *data mining*. Hasil dari performa sistem klasifikasi membuktikan seberapa akurat sistem tersebut dalam mengklasifikasi data. *Confusion matrix* merupakan salah satu cara yang digunakan untuk mengukur performa metode klasifikasi. Konsep dari *confusion matrix* yaitu mengandung sebuah informasi mengenai hasil dari pengklasifikasian. Terdapat empat informasi yaitu *true positif* (TP), *true negative* (TN), *false positif* (FP) dan *false negative* (FN). Sesuai Ramaswati (2014), *Confusion matrix* ditunjukkan pada Tabel 2 sebagai berikut:

Tabel 2. *Confusion Matrix*

		Predicted Class		
		1	***2***	***3***
Actual Class	***1***	$N_{(1,1)}$	$N_{(1,2)}$	$N_{(1,3)}$
	2	$N_{(2,1)}$	$N_{(2,2)}$	$N_{(2,3)}$
	3	$N_{(3,1)}$	$N_{(3,2)}$	$N_{(3,3)}$

Cost matrix digunakan untuk mengukur nilai kesalahan pada saat klasifikasi. *Cost matrix* didefinisikan sebagai C (*i*|*j*) dimana *i* adalah *class* aktual dan j adalah *class* prediksi. *Cost* kesalahan prediksi pada *multiclass* ditampilkan sebagai berikut $C_{(2,1)}$, $C_{(3,1)}$, $C_{(1,2)}$, $C_{(3,2)}$,$C_{(1,3)}$, $C_{(2,3)}$ ketika *cost* klasifikasi benar ditampilkan sebagai berikut $C_{(1,1)}$, $C_{(2,2)}$, $C_{(3,3)}$ nilai tersebut dapat dianggap bernilai 0. Sesuai dengan Xue, et. al. (2013), Table 3 merupakan tampilan dari *cost matrix*.

Tabel 3. *Cost Matrix*

	Predicted Class			
		1	***2***	***3***
Actual Class	***1***	$C_{(1,1)}$	$C_{(1,2)}$	$C_{(1,3)}$
	2	$C_{(2,1)}$	$C_{(2,2)}$	$C_{(2,3)}$
	3	$C_{(3,1)}$	$C_{(3,2)}$	$C_{(3,3)}$

Accuracy = (TP + TN) / N_{total}
Precision = TN/(TN + FP)
Recall = TP / (TP + FN)
F-Measure = 2 x ((Recall x Precision)/ ((Recall + Precision))
Total Cost = N_{total} [(C_{FN} +FN)*(C_{FP} + FP)]

4. Hasil dan Pembahasan

A. Persiapan Data

Pada tahap persiapan data, dilakukan eliminasi data *outlier* dan pengisian data atribut yang kosong menggunakan data rata-rata atribut dari *dataset* tersebut. Dari persiapan ini didapati *dataset glass* sebanyak 214 data dan 9 atribut, *lympografi* sebanyak 148 data dan 18 atribut, *dataset vehicle* sebanyak 376 data dan 18 atribut, *dataset thyroid* 215 data dan 5 atribut, serta *dataset wine* sebanyak 178 data dan 13 atribut. Hasil pengujian ditampilkan seperti Tabel 4.

Tabel 4. Hasil pengujian PSO

Dataset training	***Decision Tree***		***Naïve Bayes***	
	DT (%)	**DT + PSO (%)**	**NB (%)**	**NB+ PSO (%)**
Glass	67.29	70.09	33.18	29.44
Lympografi	75.00	78.99	74.32	80.43
Vehicle	71.01	71.54	42.02	40.96
Thyroid	94.42	94.42	96.74	97.67
Wine	92.86	94.38	98.31	94.94

Berdasarkan hasil pengujian nilai bobot *dataset glass, lympografi, vehicle, thyroid* dan *wine*

menggunakan algoritma PSO pada *decision tree* dan *naïve bayes*. Hasil pengujian, diperoleh nilai accuracy maksimal pada *decision tree* jika nilai atribut berturut-turut ≥ 0.2, > 0, ≥ 0.3,≥ 0.1, > 0. Sedangkan hasil pengujian seleksi atribut menggunakan algoritma PSO dengan *naïve bayes* pada *dataset lympografi, thyroid* dan *wine* memiliki nilai accuracy maksimal jika atribut berturut-turut > 0, > 0.1, > 0 namun pada *dataset glass* dan *vehicle*, PSO tidak bisa meningkatkan nilai accuracy.

B. Pengujian dan Analisis

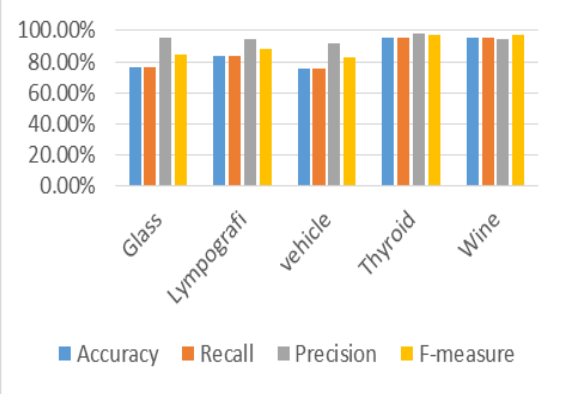
Penelitian ini dilakukan dengan dua metode pengujian, yaitu menggunakan metode *Cost Sensitive Decision Tree C5.0* (csDT) dan metode *Cost Sensitive Naive Bayes* (csNB).

Percobaan pertama, pengujian pada *dataset glass, lympografi, vehicle, thyroid*, dan *wine* menggunakan metode csDT C5.0. Hasil dari pengujian ditampilkan sebagai berikut:

Tabel 5. Hasil Pengujian *csDT C5.0*

Dataset	***Accuracy (%)***	***Recall (%)***	***Precision (%)***	***F-measure (%)***
Glass	76.17	76.17	95.23	84.64
Lympografi	83.33	83.33	94.44	88.54
Vehicle	75.27	75.27	91.76	82.70
Thyroid	95.81	95.81	97.91	96.85
Wine	95.83	95.83	94.44	96.86

Berdasarkan Tabel 5 menunjukan hasil proses pengujian pada *dataset glass, lympografi, vehicle, thyroid, wine* memiliki performa *accuracy* berturut-turut, 76.17%, 83.33%, 75.27%, 95.81%, dan 95.83%. Dimana nilai *accuracy* menunjukan ratio prediksi benar terbesar didapatkan pada pengujian *csDT C5.0* dengan menggunakan dataset *wine*. Sedangkan untuk hasil pengujian *precision* yang menunjukan hasil ratio prediksi terbesar didapatkan dengan menggunakan dataset *Thyroid*. Hasil lain menunjukan bahwa *recall* dan *f-measure* memiliki nilai terbesar pada pengujian menggunakan dataset *wine* masing-masing sebesar 95.83% dan 96.86%.



Gambar 2. Diagram Hasil Pengujian csDT

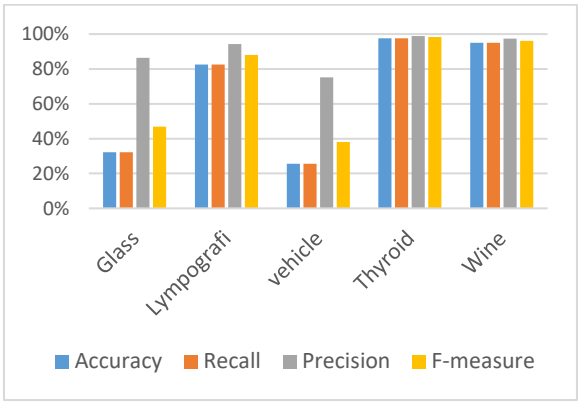
Gambar 2 merupakan visualisasi hasil pengujian Tabel 5 dalam bentuk diagram. Diagram tersebut menunjukan bahwa dengan menggunakan metode *csDT C5.0* pada kelima jenis dataset tersebut memiliki prosentase *accuracy* diatas 75%. Hasil tersebut menunjukan bahwa metode *csDT C5.0* memiliki performa klasifikasi yang cukup baik pada pengujian ini.

Pada percobaan kedua pengujian dilakukan dengan menggunakan metode csNB pada lima *dataset*. Hasil ditunjukkan pada Tabel 6.

Tabel 6. Hasil Pengujian csNB

Dataset	***Accuracy (%)***	***Recall (%)***	***Precision (%)***	***Fmeasure (%)***
Glass	32.24	32.24	86.45	46.97
Lympografi	82.61	82.61	94.20	88.03
Vehicle	25.53	25.53	75.18	38.12
Thyroid	97.67	97.67	98.84	98.25
Wine	94.94	94.94	97.47	96.19

Tabel 6 merupakan hasil pengujian *dataset glass, lympografi, vehicle, thyroid* dan *wine* memiliki nilai *accuracy* berturut-turut sebagai berikut 32.24%, 82.61%, 25.53%, 97.67%, 94.94%. Dimana nilai *accuracy* menunjukan ratio prediksi benar terbesar didapatkan pada pengujian *csNB* dengan menggunakan dataset *Thyroid*. Sedangkan untuk hasil pengujian *precision*, *recall* dan *f-measure* menunjukan nilai prosentase terbesar dengan menggunakan dataset *Thyroid* masing-masing sebesar 94.94%, 97.47% dan 96.19%.



Gambar 3. Diagram Hasil Pengujian csNB

Gambar 3 merupakan visualisasi hasil pengujian Tabel 6 dalam bentuk diagram. Diagram tersebut menunjukan bahwa dengan menggunakan metode *csNB* pada kelima jenis dataset tersebut terdapat tiga dataset yang memiliki nilai prosentase diatas 75% yaitu *lympografi, thyroid* dan *wine*. Sedangkan dataset *glass* dan *vehicle* memiliki prosentase *accuracy* dibawah 75%. Hasil tersebut menunjukan bahwa metode *csNB* memiliki performa klasifikasi yang kurang baik pada karakteristik dataset jenis *glass* dan *vehicle*.

Tabel 7. Hasil Perhitungan *Cost*

	Decision Tree			***Naïve Bayes***		
Dataset	***DT***	***DT + PSO***	***csDT***	***NB***	***NB + PSO***	***csNB***
Glass	9800	8192	5202	40898	40898	42050
Lympografi	2738	1682	1058	2888	1458	1152
vehicle	22898	23762	17298	95048	95048	156800
Thyroid	288	288	162	98	50	50
Wine	288	200	98	18	18	162

Tabel 7 merupakan hasil pengujian klasifikasi menggunakan metode *decision tree* dan *naïve bayes*, dimana kesalahan klasifikasi dapat dihitung menggunakan cost. Jadi, semakin besar nilai *cost* maka *classifier* memiliki performa yang buruk. Hasil pengujian menunjukan bahwa metode *cost sensitive* mampu meningkatkan performa *classifier* dengan meminimalkan kesalahan klasifikasi pada metode *decision tree* dengan baik pada semua dataset yang diuji. Namun, metode *cost sensitive* tidak mampu meminimalkan kesalahan klasifikasi pada semua data uji jika menggunakan metode *naïve bayes*. Hal tersebut ditunjukan pada hasil pengujian menggunakan dataset *glass* dan *vehicle* yang adanya peningkatan nilai costnya masing-masing sebesar 40898 dan 95048 menjadi 42050 dan 156800.

Hasil tersebut menunjukan bahwa metode *cost sensitive decision tree* C5.0 memiliki performa klasifikasi yang baik dan stabil pada kelima data uji. Sedangkan *cost sensitive naïve bayes* dinilai memiliki performa kurang baik jika melakukan klasifikasi data *multiclass*.

5. Kesimpulan

Berdasarkan hasil pengujian dan evaluasi dapat disimpulkan bahwa pengujian dengan menggunakan metode *cost sensitive decision tree* C5.0 memiliki nilai accuracy yang lebih baik dari pada menggunakan metode *cost sensitive naïve bayes* pada *dataset glass*, *lympografi*, *vehicle* dan *wine* berturut-turut 76.17%, 83.33%, 75.27% dan 95.83%. Sedangkan dengan menggunakan metode *cost sensitive naïve bayes* memiliki nilai accuracy yang lebih baik dari pada *cost sensitive decision tree C5.0* pada *dataset thyroid* sebesar 97.67%. Performa klasifikasi tidak hanya ditentukan pada pemilihan metode namun karakteristik dari data uji meliputi jenis data dan jumlah class juga akan mempengaruhi hasil klasifikasi. Pada penelitian ini menunjukan cost sentive decision tree C5.0 memilki performa yang lebih baik pada jenis data real dan interger sedangkan metode cost naive bayes memiliki performa yang baik jika menggunakan jenis data numerik.

Daftar Pustaka:

Ali H, M. N. M. Salleh, Saedudin, R. and Hussain, K. (2019): *Imbalance class problems in data mining: a review.",* Indones. J. Electr. Eng. Comput. Sci., vol. 14, no. 3., 2019, pp. 1560–1571.

Bernard, S., Chatelain, C., and Adam, S., (2015): *The Multiclass ROC Front method for cost-sensitive classification*. Pattern Recognition, vol. 52., 2015: pp. 46–60.

Chai, X., Deng, L., Yang., Q., et al., (2004): *Test Cost Sensitive Naïve Bayes Classification.*, Proceedings of the 4th IEEE International Conference on Data Mining pp.51-58.

Daraei A., (2017): *An Efficient Predictive Model for Myocardial Infarction Using Cost-sensitive J48 Model.* Iran J Public Health, Vol. 46, No.5. 2017: pp.682-692.

Domingos P., (1999): *MetaCost: A general method for making classifiers cost-sensitive.* Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining. ACM Press., 1999, pp. 155-164.

Faisal KM, Mofizur RC. (2011): *Enhanced classification accuracy on naïve bayes data mining models.* International journal of computer applications, 2011, 28(3): 9-16

Friedman N.,Geiger., D and Goldezmidt M., (1997): *Bayesian Network Classifier*. Machine Learning, 1997, pp:131-163.

Haldankar A.N. (2016): *A Cost Sensitive classifier for Big Data.* IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT).

Jauhari F,, Supianto, A.A., (2019): *Building student's performance decision tree classifier using boosting algorithm.* Indones. J. Electr. Eng. Comput. Sci., vol. 14, no. 3., 2019, pp. 1298–1304.

Janasthar, S. and A. Hanskunatai, (2014): *The ensemble of Naïve Bayes Classifiers for Hotel Searching*, International Computer Science and Engineering Conference (ICSEC), 2014.

Larose D. T., (2005): *Discovering knowledge in data: an introduction to data mining.* Jhon Wiley & Sons Inc.

Pandya R., dan Pandya, J., (2015): *C5.0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning.* International Journal of Computer Applications, vol. 117., 2015: pp. 0975 – 8887.

Patel B.R, dan Rana, K.K., (2014): *A Survey on Decision Tree Algorithm For Classification.* International Journal of Engineering Development and Research, Vol. 2, No. 1., 2014.

Patel B.N, Prajapati, S.G., and Lakhtaria, K.I. (2012): *Efficient Classification of Data Using*

Decision Tree. Bonfring international journal of data mining, Vol. 2, No. 1., 2012.

Ramaswati M., (2014): *Validating Predictive Performance of Classifier Models for Multiclass Problem in Educational Data Mining*, International Journal of Computer Science Issue,Vol. 11, Issue.5., 2014.

Herasmara, R., Muslim, M.A., Mudjirahardjo, P. (2019): *Optimasi Struktur Convolutional Neural Network LeNet5m dengan Pendekatan MorphNet*, Jurnal EECCIS, Malang, Teknik Elektro Universitas Brawijaya.

Thomas M.C. and Joy A. T. (2006): *Elements of imformation Theory*, A John Wiley & Sons, INC., Publication, 2006, pp. 13-14.

Wang, S. and Yao, X., (2012): *Multiclass Imbalance Problems : Analysis and Potential Solutions.*" IEEE Trans. Syst. Man. Cybern., vol. 42, no. 4., 2012, pp. 1119–1130.

Wei S, Ching, Y.K., Chieh, C.S., and Jung, L.Z. (2008): *Particle Swarm Optimization for Parameter Determination and Feature Selection of Support Vector Machines.* ScienceDirect: Expert System With Aplications., 2008, pp.1817- 1824.

Xiangju L, Hong Z and William Z., (2015): *A Cost Sensitive Decision Tree Algorithm with Two Adaptive Mechanisms*, Knowledge-Based System, vol. 88, 2015, pp. 24-23.

Xue, B., Zhang, M., & Browne, W. N. (2013): *Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach.* IEEE Transactions on Cybernetics, 43(6), 2013, pp. 1656–1671.

Zhang, S., Zhang, C., and Yang, Q., (2010): *Data preparation for data mining*. Applied Artificial Intelligence an International Journal, Vol. 17, 2010, pp. 5-6.