Penerapan Metode *Average Gain*, *Threshold Pruning* dan *Cost Complexity Pruning* untuk *Split* Atribut pada Algoritma C4.5

Erna Sri Rahayu, Romi Satria Wahono dan Catur Supriyanto
Fakultas Ilmu Komputer Universitas Dian Nuswantoro
ernaesr81@gmail.com, romi@romisatriawahono.net, caturs@research.dinus.ac.id

Abstrak: C4.5 adalah algoritma klasifikasi *supervised learning* untuk membentuk pohon keputusan (*Decision Tree*) dari data. *Split* atribut merupakan proses utama dalam pembentukan pohon keputusan (*Decision Tree*) di C4.5. Proses pemilihan *split* atribut di C4.5 belum dapat mengatasi *misclassification cost* di setiap *split* sehingga berpengaruh pada kinerja pengklasifikasi. Setelah dilakukan pemilihan *split* atribut, proses selanjutnya adalah *pruning*. *Pruning* adalah proses yang dilakukan untuk memotong atau menghilangkan beberapa cabang (*branches*) yang tidak diperlukan. Cabang (*branches*) atau *node* yang tidak diperlukan dapat menyebabkan ukuran *Decision Tree* menjadi sangat besar dan hal ini disebut *over-fitting*. Untuk saat ini *over-fitting* merupakan trend riset di kalangan peneliti. Metode-metode untuk pemilihan *split* atribut diantaranya *Gini Index*, *Information Gain*, *Gain Ratio* dan *Average Gain* yang diusulkan oleh Mitchell. *Average Gain* tidak hanya mengatasi kelemahan pada *Information Gain* tetapi juga membantu untuk memecahkan permasalahan dari *Gain Ratio*. Metode *split* atribut yang diusulkan pada penelitian ini adalah menggunakan nilai *average gain* yang dikalikan dengan selisih misklasifikasi. Sedangkan teknik *pruning* dilakukan dengan mengkombinasikan *threshold pruning* dan *cost complexity pruning*. Pada penelitian ini, pengujian metode yang diusulkan akan diterapkan pada dataset kemudian hasil kinerjanya akan dibandingkan dengan hasil kinerja metode *split* atribut yang menggunakan *Gini Index*, *Information Gain* dan *Gain Ratio*. Metode pemilihan *split* atribut yang menggunakan *average gain* yang dikalikan dengan selisih misklasifikasi dapat meningkatkan kinerja pengklasifikasi C4.5. Hal ini ditunjukkan melalui uji *Friedman* bahwa metode *split* atribut yang diusulkan, ditambah dengan *threshold pruning* dan *cost complexity pruning* mempunyai hasil kinerja berada di peringkat 1. Pohon keputusan *(Decision Tree)* yang terbentuk melalui metode yang diusulkan berukuran lebih kecil.

Kata kunci: *Decision Tree*, C4.5, *split* atribut, *pruning*, *over-fitting*, *average gain*.

1 PENDAHULUAN

Decision Tree merupakan algoritma pengklasifikasian yang sering digunakan dan mempunyai struktur yang sederhana dan mudah untuk diinterpretasikan (Mantas & Abellán, 2014). Pohon yang terbentuk menyerupai pohon terbalik, dimana akar (*root*) berada di bagian paling atas dan daun (*leaf*) berada di bagian paling bawah. *Decision Tree* merupakan model klasifikasi yang berbentuk seperti pohon, dimana *Decision Tree* mudah untuk dimengerti meskipun oleh pengguna yang belum ahli sekalipun dan lebih efisien dalam menginduksi data (C. Sammut, 2011). Induksi di *Decision Tree* adalah salah satu teknik tertua dan yang paling tertua untuk model *learning discriminatory*, yang mana model tersebut telah dikembangkan secara mandiri di statistik dan di komunitas *machine learning*. Proses pembentukan *Decision Tree* dibagi menjadi 3 (T Warren Liao, 2007) yaitu, (1) pembentukan pohon (*tree*), (2) *pruning*, (3) mengekstrak aturan (*rule*) dari pohon keputusan yang terbentuk. *Decision Tree* baik digunakan untuk klasifikasi atau prediksi.

Decision Tree telah diaplikasikan di berbagai bidang contohnya di bidang pengobatan (Setsirichok et al., 2012). Salah satu contohnya adalah penerapan C4.5 *Decision Tree* yang digunakan untuk mengklasifikasikan karakteristik darah sehingga dapat mengklasifikasikan 80 *class* kelainan thalassemia yang menyebar di Thailand. Contoh lain penerapan *Decision Tree* untuk memprediksi pasien kanker payudara (Ture, Tokatli, & Kurt, 2009). Selain di bidang pengobatan, *Decision Tree* juga diterapkan di bidang bisnis (Duchessi & Lauría, 2013)(Duchessi & Lauría, 2013) dan deteksi kegagalan (Sahin, Bulkan, & Duman, 2013). Tantangan di *Decision Tree* saat ini adalah sehubungan dengan performa tingkat akurasi, skalabilitas, perkembangan *dataset* dan aplikasi-aplikasi baru yang belum dikembangkan.

Beberapa algoritma yang telah dikembangkan berdasar *Decision Tree* adalah (1) CHAID (*Chi-squared Automatic Interaction Detection*) yang mana *split* tiap *node* berdasar pada *Chi-square test* pada masing-masing atribut, (2) CART (*Classification And Regression Tree*) membentuk *Decision Tree* dengan penghitungan *Gini Index* untuk kriteria *split*, (3) C4.5 yang merupakan variasi pengembangan dari ID3 (*Iterative Dichotomiser 3)* (Gorunescu, 2011). Jika ID3 (*Iterative Dichotomiser 3)* menggunakan *Entropy* untuk kriteria *split*, sedangkan di C4.5 menggunakan *Gain Ratio* untuk kriteria *split*nya. Atribut yang memiliki *Gain Ratio* tertinggi yang akan dipilih. Lim et al (2000) telah membandingkan tingkat akurasi, kompleksitas dan waktu training dari ketiga algoritma klasifikasi tersebut, dan hasilnya menunjukkan bahwa C4.5 mempunyai tingkat akurasi yang bagus dan mudah untuk diinterpretasikan.

C4.5 adalah algoritma klasifikasi *supervised learning* untuk membentuk pohon keputusan (*Decision Tree*) dari data (Mantas & Abellán, 2014)(Mantas & Abellán, 2014)(Quinlan, 1993). C4.5 *Decision Tree* menggunakan kriteria *split* yang telah dimodifikasi yang dinamakan *Gain Ratio* oleh Mitchael (1997) dalam proses pemilihan *split* atribut. *Split* atribut merupakan proses utama dalam pembentukan pohon keputusan (*Decision Tree*) di C4.5 (Quinlan, 1986). Tahapan dari algoritma C4.5 adalah (1) menghitung nilai *Entropy*, (2) menghitung nilai *Gain Ratio* untuk masing-masing atribut, (3) atribut yang memiliki *Gain Ratio* tertinggi dipilih menjadi akar (*root*) dan atribut yang memiliki nilai *Gain Ratio* lebih rendah dari akar (*root*) dipilih menjadi cabang (*branches*), (4) menghitung lagi nilai *Gain Ratio* tiap-tiap atribut dengan tidak mengikutsertakan atribut yang terpilih menjadi akar (*root*) di tahap sebelumnya, (5) atribut yang memiliki *Gain Ratio*

tertinggi dipilih menjadi cabang (*branches*), (6) mengulangi langkah ke-4 dan ke-5 sampai dengan dihasilkan nilai *Gain* = 0 untuk semua atribut yang tersisa.

Setelah dilakukan pemilihan *split attribute*, proses selanjutnya adalah *pruning*. *Pruning* adalah proses yang dilakukan untuk memotong atau menghilangkan beberapa cabang (*branches*) yang tidak diperlukan (C. Sammut, 2011). *Pruning* dilakukan untuk mengembangkan kehandalan generalisasi *Decision Tree* dan akurasi prediksi *Decision Tree* dengan memindahkan *node* yang tidak diperlukan di *Decision Tree* (Otero, Freitas, & Johnson, 2012). Cabang (*branches*) atau *node* yang tidak diperlukan dapat menyebabkan ukuran *Decision Tree* menjadi sangat besar dan hal ini disebut *over-fitting* (Larose, 2006) (Larose, 2005). Untuk saat ini *over-fitting* merupakan trend riset di kalangan peneliti.

Over-fitting dapat menghasilkan model yang baik di training data tetapi secara normal tidak dapat menghasilkan model *tree* yang baik ketika diterapkan di *unseen data* (Wang, Qin, Jin, & Zhang, 2010). *Over-fitting* disebabkan oleh *noisy* data, *irrelevant feature* (Wang et al., 2010). *Noisy* data akan menyebabkan terjadinya misklasifikasi, sehingga *over-fitting* akan menyebabkan tingkat akurasi yang buruk dalam pengklasifikasian. Permasalahan lain di C4.5 adalah ketidakseimbangan data yang juga menyebabkan akurasi C4.5 buruk dalam pengklasifikasian data.

Permasalahan *over-fitting* dapat diatasi dengan melakukan teknik *pruning* (Zhang, 2012). Macam-macam teknik *pruning* untuk mengatasi *over-fitting* adalah *Laplace pruning* yang diperkenalkan oleh Bradford, yang kemudian disempurnakan oleh Provost dan Domingos. Model yang dikembangkan yaitu *Decision Tree* yang melewatkan proses *pruning* dengan melakukan *smoothing* menggunakan *Laplace correction method* (Wang, Qin, Zhang, & Zhang, 2012). Tetapi metode ini mempunyai kelemahan pada *dataset* dengan distribusi data yang tidak seimbang sehingga Zadrozny dan Elkan mengusulkan *Decision Tree* yang tidak *di-pruning* dan menempatkan skor *smoothing* dari daun (*leaf*). Metode *smoothing* yang diusulkan dinamakan *m-estimation* (Wang et al., 2010). Metode ini dilakukan untuk mendapatkan perkiraan probabilitas yang lebih baik.

Banyak strategi untuk pemilihan *split* atribut, diantaranya *Information Gain* (Quinlan, 1986) dan *GINI Index* (Gorunescu, 2011). Kedua strategi diatas digunakan untuk mengukur *impurity*, dimana atribut yang mempunyai nilai pengurangan *impurity* maksimal (*most impurity reduce*) akan terpilih untuk membangun *Decision Tree*. Metode yang lain adalah *average gain* yang diusulkan oleh Mitchell. *Average gain* tidak hanya mengatasi kelemahan pada *informasi gain* tetapi juga membantu untuk memecahkan permasalahan dari *gain ratio*.

Metode yang diusulkan pada penelitian ini untuk proses pemilihan *split* atribut adalah menggunakan nilai *average gain* yang dikalikan dengan selisih antara misklasifikasi setelah *di-split* dan sebelum *di-split*. Sedangkan permasalahan *over-fitting* akan diatasi dengan menerapkan metode *threshold pruning* sebagai proses *pre-pruning*. *Threshold pruning* dilakukan dengan menghitung *misclassification cost* untuk masing-masing potensial *split* atribut. Sedangkan untuk *post pruning* dipilih metode *cost complexity pruning* yang merupakan salah satu jenis *pessimistic error pruning*.

Paper ini disusun sebagai berikut: pada bagian 2 paper terkait dijelaskan. Pada bagian 3, metode yang diusulkan dijelaskan. Hasil percobaan perbandingan antara metode yang diusulkan dengan metode lainnya disajikan pada bagian 4. Akhirnya, kesimpulan dari penelitian kami disajikan pada bagian terakhir.

2 PENELITIAN TERKAIT

2.1 Metode Info Gain (Quinlan, 1993)

Metode penelitian ini diperkenalkan oleh Quinlan dengan berdasar model ID3 (*Iterative Dichotomiser 3)*. Metode yang diperkenalkan Quinlan cocok untuk *dataset* dengan variabel diskret akan tetapi metode yang diperkenalkan tidak cocok untuk *dataset* dengan *missing value*. Metode penelitian Quinlan menggunakan pemilihan *split* atribut yang disebut *Gain*. Informasi yang disampaikan tergantung pada probabilitas dan dapat diukur dalam *bits* sebagai minus algoritma berbasis 2. Sebagai contoh -log2(1/8) = 3 *bits*. Untuk mendapatkan nilai yang diharapkan (*expected information*) yang berkaitan dengan *class-class* yang ada, maka Quinlan menjumlahkan seluruh *class* secara proporsional dengan frekuensi mereka di *S*, seperti di bawah ini.

$$info(S) = -\sum_{j=1}^{k} \frac{freq(Cj,S)}{|S|} \; x \; log_2(\frac{freq(C_j,S)}{|S|}) \; bits \quad (2.1)$$

Ketika diterapkan di training kasus, info (*T*) diukur dari rata-rata informasi yang dibutuhkan untuk mengidentifikasi *class* yang terdapat di kasus *T*. Hal ini disebut dengan *Entropy* (S). Sekarang bandingkan perhitungan yang mirip setelah *T* selesai dipartisi sesuai dengan *n* hasil dari tes *X*. Nilai yang diharapkan (*expected information*) dapat ditentukan melalui pembobotan jumlah dari semua *subset*.

$$info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \; x \; info(T_i) \quad (2.2)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

n = jumlah *subset*
T = atribut
T_i = *subset* dari sebuah atribut

Entropy didefinisikan sebagai nilai informasi yang diharapkan. Dan nilai *Entropy* dapat dihitung melalui rumus persamaan dibawah ini:

$$Gain\ (X) = info(S) - info_x(T) \quad (2.3)$$

Perhitungan informasi di atas didapat dari partisi *T* sesuai dengan tes *X*. Kemudian dipilih atribut yang mempunyai nilai *information gain* yang maksimal.
Pada metode yang diperkenalkan Quinlan tidak melakukan proses *pruning*.

2.2 Metode Info Gain Ratio (Quinlan, 1993)

Metode penelitian ini merupakan pengembangan dari metode *Iterative Dichotomiser 3* (ID3). Quinlan memperkenalkan metode ini dengan nama C4.5, dimana untuk pemilihan *split* atribut menggunakan metode *Info Gain Ratio (IGR)* menggantikan *Info Gain (IG).* C4.5 yang diperkenalkan dapat bekerja pada variabel kontinyu dan *missing value*.
Rumus persamaan *Info Gain Ratio (IGR)* seperti berikut:

$$gain\ ratio\ (X) = \frac{gain\ (X)}{split\ info\ (X)} \quad (2.4)$$

Dimana split info (X) mempunyai persamaan rumus sebagai berikut:

$$split\ info\ (X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \; x \; log_2(\frac{|T_i|}{|T|}) \quad (2.5)$$

Pada metode penelitian ini proses *pruning* menggunakan *posterior complex pruning*.

2.3 Metode Credal Decision Tree (Abellán, 2013)

Joaquin Abellan menggunakan *Imprecise Info Gain* (IIG) untuk pembentukan *Decision Tree*. Pohon (*tree*) yang dibentuk hanya untuk variabel diskret. Metode yang diusulkan oleh Joaquin Abellan & Andres R. Masegosa disebut *Credal Decision Tree. Credal Decision Tree* tidak dapat bekerja pada *dataset* yang mempunyai *missing values*.

Pada proses pemilihan *split* atribut, *Credal Decision Tree* menggunakan *imprecise probability* dan *uncertainty measure* di *credal set*. Interval probabilitas didapat dari *dataset* untuk masing-masing kasus dalam sebuah variabel *class* menggunakan *Walley's Imprecise Dirichlet Model* (IDM). Didefinisikan metode yang diusulkan Abellan dan Moral sebagai *Imprecise Info Gain* (IIG) dengan persamaan rumus seperti berikut:

$$IIG\ (X,C) = S * \big(K(C)\big) - \sum_i p(x_t)\ S * (K(C|X = x_t)) \quad (2.6)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

C	=	*class* variabel
X	=	atribut
S	=	maksimum *Entropy*
$K(C)\ dan\ (K(C\|X = x_t))$	=	*credal set* yang diperoleh melalui *Imprecise Dirichlet Model* (IDM)
$C\ dan\ C\|X = x_t$	=	variabel
$P(X = x_t)$	=	probabilitas distribusi

Pada metode Credal *Decision Tree*, atribut yang terpilih adalah atribut yang mempunyai nilai *Imprecise Info Gain (IIG)* maksimal. Metode *Credal Decision Tree* melewatkan proses *post pruning*.

Dataset yang digunakan diunduh dari *UCI repository of machine learning data sets* dengan alamat *ftp://ftp.ics.uci.edu/machine-learning-databases*. Dataset tersebut antara lain; *Anneal, Audiology, Autos, Breast-cancer, Colic, Cmc, Credit-german, Diabetes-pima, Glass 2, Hepatitis, Hypothyroid, Ionosphere, Kr-vs-kp, Labor, Lymph, Mushroom, Segment, Sick, Solar-flare1, Sonar, Soybean, Sponge, Vote, Vowel, Zoo*.

2.4 Metode Credal C4.5 (Mantas & Abellán, 2014)

Metode penelitian ini diusulkan oleh Carlos J. Mantas & Joaquin Abellan. Metode penelitian yang diusulkan cocok untuk pengklasifikasian *dataset* dengan *noise*. Pemilihan *split* atribut pada metode ini menggunakan *Imprecise Info Gain Ratio* (IIGR) menggantikan *Info Gain Ratio* (IGR). *Imprecise Info Gain Ratio* (IIGR) menggunakan *imprecise probability* untuk menghitung nilai atribut dan variabel *class*. Metode penelitian yang diusulkan oleh Carlos J. Mantas & Joaquin Abellan disebut dengan *Credal C4.5*. Perhitungan *Imprecise Info Gain Ratio* (IIGR) pada *Credal C4.5* menggunakan persamaan rumus berikut ini:

$$IIGR^D\ (Class, X) = \frac{IIG^D(Class,X)}{H(X)} \quad (2.7)$$

$$IIG^D(Class, X) = H * \big(K^D(Class)\big) - \sum_i P^D(X = x_i)\ H * (K^D(Class|X = x_i)) \quad (2.8)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

$Class$	=	*class* variabel
X	=	atribut
H	=	maksimum *Entropy*
$K^D(C)\ dan\ (K^D(C\|X = x_i))$	=	*credal set* yang diperoleh melalui IDM
$C\ dan\ C\|X = x_i$	=	variabel
$P^D(X = x_i)$	=	probabilitas distribusi

Jika C4.5 klasik menggunakan maksimal *Entropy* tapi di *Credal C4.5* menggunakan prinsip maksimal *uncertainty*.

Prosedur pembentukan pohon (*tree*) *Credal C4.5*

1. *If L=Ø, then Exit*
2. *Let D be the partition associated with node* No
3. *If |D| < minimum number of instances, then Exit*
4. *Calculate* $P^D(X=x_i)$ $(i=1,...,n)$ *on the convex set* $K^D(X)$
5. *Compute the value*
   $\alpha = max\ x_j \epsilon M\{IIGR^D(C, X_j)\}$
   With $M = \{X_j \in L / IIG^D(C, X_j) > avg x_j \in L\ \{IIG^D(C, X_j)\}\}$
6. *If* $\alpha \leq 0$ *then Exit*
7. *Else*
8. *Let* X_l *be the variable for which the maximum* α *is attained*
9. *Remove* X_l *from L*
10. *Assign* X_l *to node No*
11. *For each possible value* x_i *of* X_l
12. *Add a node* No_l
13. *Make* No_l *a child of No*
14. *Call BuildCredalC4.5Tree* (No_l, L)

Untuk proses *pruning*, metode yang diusulkan *Carlos J. Mantas & Joaquin Abellan* seperti C4.5 klasik yaitu menggunakan *post pruning* yakni menggunakan *Pessimistic Error Pruning*.

Dataset yang digunakan pada model penelitian *Credal C4.5* didapat dari *UCI repository of machine learning datasets* dengan alamat http://archive.ics.uci.edu/ml. *Dataset* yang digunakan 50 *dataset* diantaranya: *Anneal, Arrhythmia,Audiology, Autos, Balance-scale, Breast-cancer, Wisconsin-breast-cancer, Car, CMC, Horse-colic, Credit-rating, German-credit, Dermatology, Pima-diabetes, Ecoli, Glass, Haberman, Cleveland-14-heart-disease, Hungarian-14-heart-disease, Heart-statlog, Hepatitis, Hypothyroid, Ionosphere, Iris, kr-vs-kp, Letter, Liver-disorder, lymphography, mfeat-pixel, Nursery, Optdigits, Page-blocks, Pendigits, Primary-tumor, Segment, Sick, Solar-flare2, Sonar, Soybean, Spambase, Spectrometer, Splice, Sponge, Tae, Vehicle, Vote, Vowel, Waveform, Wine, Zoo*.

Pada penelitian ini akan menerapkan 1) metode baru *split* atribut yaitu dengan menghitung nilai *average gain* yang dikalikan dengan nilai selisih dari misklasifikasi sebelum *di-split* dan sesudah *di-split*. 2) menerapkan *pruning* yang yang terdiri dari *threshold pruning* dan *cost complexity pruning* guna mengatasi *over-fitting*.

3 METODE YANG DIUSULKAN

Dataset yang digunakan dalam penelitian ini berasal dari yaitu (1) *Breast Cancer Wisconsin*, (2) *Vote*, (3) *Flare1, (4) Hepatitis, (5) Pima Indian Diabetes*. Kelima *dataset* ini dipilih karena kelima *dataset* tersebut populer digunakan, mempunyai angka yang yang proporsional dan mempunyai *missing value*. Dalam *dataset* ini dibagi dengan 90% sebagai data *training* dan 10% sebagai data *testing*.

Tabel 1. *Dataset* yang Digunakan dalam Eksperimen

Dataset	Jumlah *Record*	Jumlah Atribut	Jumlah Atribut Nominal	Jumlah Atribut Numerik	Missing Value	Jumlah *Class*
Breast Cancer Wisconsin	286	9	9	0	16	2
Vote	435	6	6	0	288	2
Flare1	323	12	12	0	5	2
Pima Indian Diabetes	768	8	0	8	752	2
Hepatitis	155	19	15	4	122	2

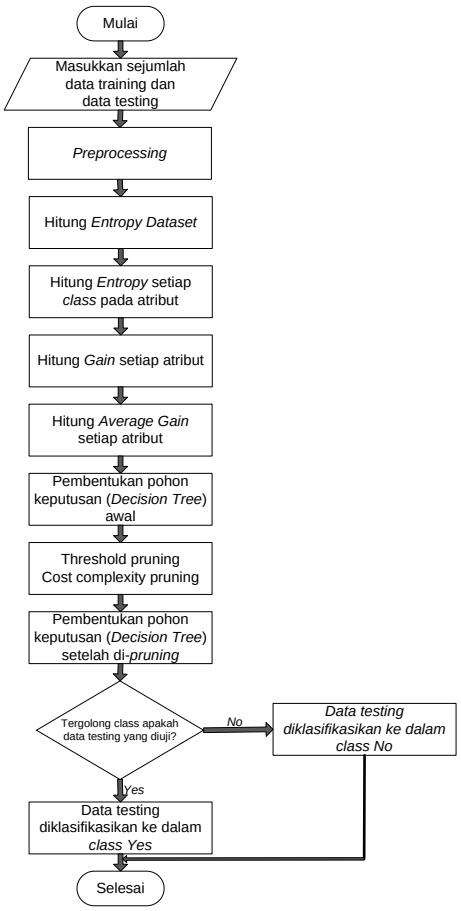
Dataset yang digunakan dalam penelitian ini mempunyai *missing value* yang harus diperlakukan secara khusus. Adapun penanganan *missing value* menurut Han dan Kamber (Han, Jiawei; Kamber, Micheline; Pei, 2012) adalah:

1. Mengabaikan *tuple* yang berisi *missing value*.

2. Mengganti *missing value* secara manual.
3. Mengganti *missing value* dengan konstanta global (misal "*Unknown"* atau ∞).
4. Mengganti *missing value* dengan nilai *mean* atau *median* dari atribut.
5. Mengganti *missing value* dengan nilai *mean* atau *median* dari semua sampel.
6. Mengganti *missing value* dengan nilai kemungkinan terbanyak dari *dataset*.

Pada penelitian ini, *missing value* pada *dataset nominal* akan digantikan dengan nilai yang mempunyai frekuensi terbanyak pada *dataset*. Sedangkan pada *dataset numerik* maka *missing value* digantikan dengan nilai *median* dari atribut.

Selanjutnya kami mengusulkan metode AG, dimana AG adalah metode *split* atribut menggunakan *average gain* yang dikalikan dengan selisih misklasifikasi. Setelah proses *split* atribut dilanjutkan dengan teknik pruning. Teknik *pruning* yang digunakan yaitu *threshold pruning* dan *cost complexity pruning*. Metode AG yang diintegrasikan dengan *threshold pruning* dan *cost complexity pruning* selanjutnya dalam penelitian ini disebut AG_*Pruning*. Metode split atribut yang kami usulkan ditunjukkan pada Gambar 1.



Gambar 1. Metode Split Atribut yang Diusulkan

Berikut alur *pseudocode* dari metode yang diusulkan:

1. Masukkan *dataset*
2. Normalisasi *dataset*
3. Hitung *Entropy dataset*

$$info(S) = -\sum_{j=1}^{k} \frac{freq(Cj,S)}{|S|} \; x \; log_2(\frac{freq(C_j,S)}{|S|}) \; bits$$

4. Hitung *Entropy* setiap *class* pada atribut

$$info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \; x \; info(T_i)$$

5. Hitung *Gain* setiap atribut

$$Gain\ (X) = info(S) - \; info_x(T)$$

6. Hitung *Average Gain* setiap atribut

$$Split\ Atribut = \frac{\left(2^{Averagegain\left(A_i,T\right)}-1\right) x\ Redu\_Mc(A_i)}{TC(A_i)+1}$$

7. Pembentukan pohon keputusan *(Decision Tree)* awal
8. Melakukan *pruning* daun keputusan *(leaf decision)* menggunakan rumus *threshold pruning*

$$Threshold = \frac{(\propto^2+1)x\ Nm\ x\ (FP+FN)}{\propto^2\ x\ Nm + (FP+FN)}$$

9. Melakukan *pruning subtree* menggunakan rumus *cost complexity pruning*

$$\alpha = \frac{\varepsilon(pruned(T,t),S)-\varepsilon(T,S)}{|leaves(T)|-|leaves(pruned(T,t)}$$

10. Pembentukan pohon keputusan *(Decision Tree)* setelah di-*pruning*
11. Pengklasifikasi akhir, dimana data *testing* diklasifikasikan menjadi *class Yes* atau *No*.

Metode *split* atribut yang diusulkan didesain untuk mencegah bias yang muncul dari atribut. Persamaan *split* atribut yang digunakan ditunjukkan melalui Persamaan 3.1.

$$Split\ Atribut = \frac{\left(2^{Averagegain\left(A_i,T\right)}-1\right) x\ Redu\_Mc(A_i)}{TC(A_i)+1} \quad (3.1)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

$TC(A_i)$ = *test cost* atribut A_i
Redu_Mc(A_i) = selisih dari *misclassification cost*

Redu_Mc(A_i) didapat dari rumus dibawah ini:

$$Redu_{Mc(A_i)} = Mc - \sum_{i=0}^{n} Mc(A_i) \quad (3.2)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

Mc = *misclassification cost* atribut A_i sebelum tes
$\sum_{i=0}^{n} Mc(A_i)$ = jumlah total *misclassification cost* atribut A_i setelah *di-split*

Nilai *test cost* dipertimbangkan karena tujuan dari metode yang diusulkan adalah untuk mengurangi atau meminimalkan *misclassification cost.*

Node yang terpilih adalah yang memenuhi syarat dibawah ini:

1. Atribut yang mempunyai nilai *split* atribut *average gain* tertinggi.
2. *Threshold*

Jika satu atau dua kondisi tersebut di atas tidak sesuai maka algoritma yang diusulkan adalah dengan memilih *node* yang mempunyai nilai *average gain* urutan ke-2 kemudian dilanjutkan dengan menguji *node* tersebut dengan 2 kondisi tersebut di atas. Jika ditemukan nilai dari sebuah atribut mempunyai nilai yang sama maka dipilih atribut yang mempunyai nilai *Redu_Mc* yang lebih besar.

Metode *pruning* yang diusulkan dalam penelitian ini dengan mengkombinasikan *threshold pruning* dan *cost complexity pruning.*

1. *Threshold pruning*

Threshold pruning memperhitungkan *misclassification costs* untuk membuat *cost reduction* pada masing-masing *split* lebih signifikan (Zhang, 2012).

Persamaan *threshold pruning* ditunjukkan pada Persamaan 3.3.

$$Threshold = \frac{(\propto^2+1)x\ Nm\ x\ (FP+FN)}{\propto^2 x\ Nm+(FP+FN)} \quad (3.3)$$

Berdasarkan persamaan di atas, berikut merupakan keterangannya:

α = parameter
Nm = jumlah minimal sampel
FP = *False Positive*
FN = *False Negative*

Parameter diatas didapat dari rentang antara jumlah minimal pada sampel yang didapat dari *10-fold cross validation* sampai dengan jumlah *misclassification reduction* (*FP* + *FN*). Dalam penelitian ini, peneliti menentukan nilai parameter α =1 dengan melalui *trial and error*.

Nilai *threshold* yang dihasilkan digunakan untuk menentukan apakah sebuah atribut perlu dipangkas atau tidak. Jika dipangkas maka atribut tersebut akan digantikan dengan daun keputusan *(decision leaf).*

2. *Cost complexity pruning*

Teknik *pruning* ini mempertimbangkan *cost complexity* dari pohon (*tree*) yaitu jumlah daun-daun *(leaves)* dalam pohon *(tree)* dan *error rate* dalam pohon *(tree)* (Rokach & Maimon, 2005). *Cost complexity pruning* terbagi menjadi 2 proses yaitu urutan dari pohon (*tree*) $T_0, T_1,\ldots,T_k$ dimana T_0 merupakan pohon (*tree*) asli sebelum *di-pruning* dan T_k adalah akar pohon (*root tree*). Tahap selanjutnya, salah satu dari pohon (*tree*) tersebut *di-pruning* berdasar perhitungan Persamaan 2.17.

$$\alpha = \frac{\varepsilon(pruned(T,t),S) - \varepsilon(T,S)}{|leaves(T)| - |leaves(pruned(T,t)} \quad (3.4)$$

Jika *subtree* menghasilkan *cost complexity* lebih rendah maka *subtree* akan di-*pruning*.

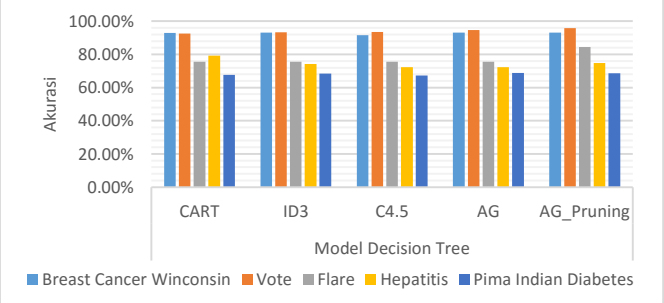
4 HASIL EKSPERIMEN

Eksperimen dilakukan menggunakan komputer personal Intel Core i3, 4 GB RAM, sistem operasi Windows 7 dan Rapid Miner 5.2.003.

Pengukuran model dilakukan dengan mengujinya menggunakan 5 *dataset UCI Repository* (*Breast Cancer Wisconsin, Vote, Flare1, Hepatitis dan Pima Indian Diabetes*). Model yang diuji adalah model *Decision Tree Classification And Regression Tree* (CART), *Iterative Dichotomiser 3* (ID3), C4.5 dan metode yang diusulkan, yaitu *Average Gain* (AG) dan *Average Gain* yang *di-pruning (*AG_*Pruning).*

Tabel 2. Rekap Pengukuran Akurasi Model Decision Tree

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	92,85%	93,13%	91,56%	93,16%	93,21%
Vote	92,64%	93,33%	93,56%	94,71%	95,86%
Flare1	75,54%	75,54%	75,54%	75,54%	84,52%
Hepatitis	79,25%	79,25%	79,25%	79,25%	79,25%
Pima Indian Diabetes	67,71%	67,71%	67,71%	67,71%	67,71%

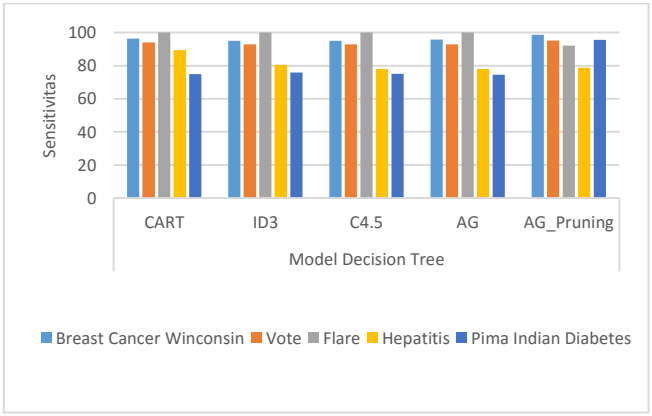


Gambar 2. Diagram Perbandingan Akurasi

Hasil pengukuran model *Decision Tree* untuk pengukuran akurasi ditunjukkan pada Tabel 2. Gambar 2 menunjukkan bahwa akurasi dari AG_*Pruning* meningkat pada 3 *dataset* yaitu *Breast Cancer Wisconsin, Vote* dan *Flare1*.

Tabel 3. Rekap Pengukuran Sensitivitas Model *Decision Tree*

Dataset	Model Decision Tree				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	96,51%	94,98%	94,98%	95,85%	98,69%
Vote	94,05%	92,86%	92,86%	92,86%	95,24%
Flare1	100%	100%	100%	100%	92,21%
Hepatitis	89,43%	89,43%	89,43%	89,43%	89,43%
Pima Indian Diabetes	75,00%	75,00%	75,00%	75,00%	75,00%

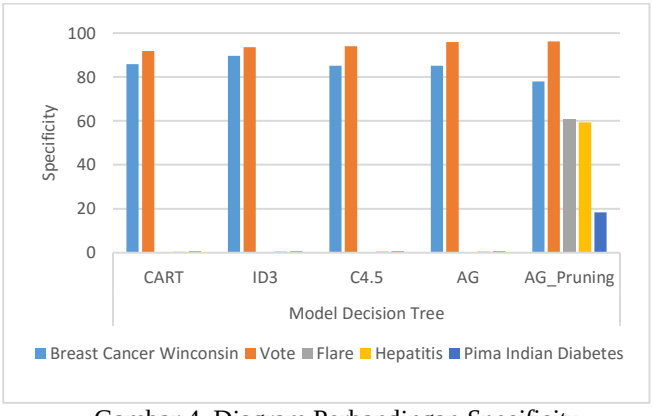


Gambar 3. Diagram Perbandingan Sensititivitas

Hasil pengukuran model *Decision Tree* untuk pengukuran sensitivitas ditunjukkan pada Tabel 3. Gambar 3 menunjukkan bahwa sensitivitas dari AG_*Pruning* meningkat hanya pada 2 *dataset* yaitu *Breast Cancer Wisconsin* dan *Vote*. Hanya pada *dataset Flare1* mengalami penurunan sensitivitas.

Tabel 4. Rekap Pengukuran *Specificity* Model *Decision Tree*

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	85,89%	89,63%	85,06%	85,06%	78,01%
Vote	91,76%	93,63%	94,01%	95,88%	96,25%
Flare1	0	0	0	0	60,76%
Hepatitis	0,41%	0,41%	0,41%	0,41%	0,41%
Pima Indian Diabetes	0,54%	0,54%	0,54%	0,54%	0,54%



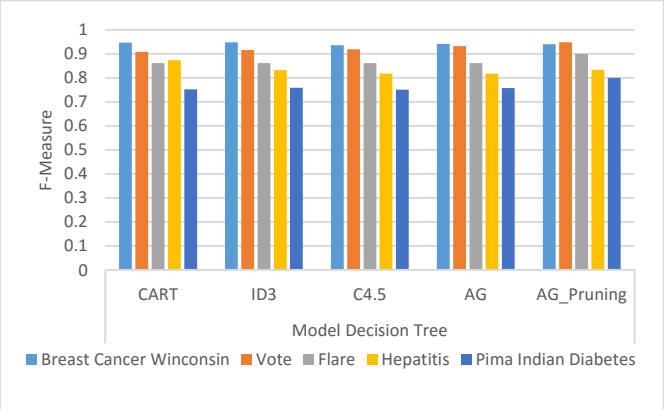
Gambar 4. Diagram Perbandingan *Specificity*

Hasil pengukuran model *Decision Tree* untuk pengukuran *specificity* ditunjukkan pada Tabel 4. Gambar 4 menunjukkan bahwa *specificity* dari AG_*Pruning* meningkat hanya pada

dataset Vote dan *Flare1*. Sedangkan pada *dataset Breast Cancer Wisconsin* mengalami penurunan *specificity*.

Tabel 5. Rekap Pengukuran *F-Measure* Model *Decision Tree*

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	0,946	0,948	0,936	0,941	0,939
Vote	0,908	0,915	0,918	0,931	0,947
Flare1	0,861	0,861	0,861	0,861	0,900
Hepatitis	0,873	0,873	0,873	0,873	0,873
Pima Indian Diabetes	0,752	0,752	0,752	0,752	0,752

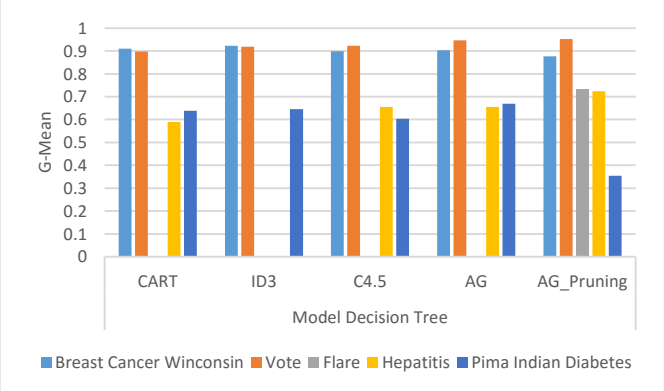


Gambar 5. Diagram Perbandingan *F-Measure*

Hasil pengukuran model *Decision Tree* untuk pengukuran *F-Measure* ditunjukkan pada Tabel 5. Gambar 5 menunjukkan bahwa *F-Measure* dari AG_*Pruning* meningkat pada dataset *Vote* dan *Flare1*. Sedangkan pada *dataset Breast Cancer Wisconsin* mengalami penurunan *F-Measure*.

Tabel 6. Rekap Pengukuran *G-Mean* Model *Decision Tree*

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	0,910	0,923	0,899	0,903	0,877
Vote	0,897	0,919	0,923	0,946	0,952
Flare1	0	0	0	0	0,731
Hepatitis	0,589	0,589	0,589	0,589	0,589
Pima Indian Diabetes	0,638	0,638	0,638	0,638	0,638



Gambar 6. Diagram Perbandingan *G-Mean*

Hasil pengukuran model *Decision Tree* untuk pengukuran *G-Mean* ditunjukkan pada Tabel 6. Gambar 6 menunjukkan bahwa *G-Mean* dari AG_*Pruning* meningkat pada *dataset Vote* dan *Flare1*. Sedangkan penurunan *G-Mean* terjadi pada *dataset Breast Cancer Wisconsin*.

Untuk mengetahui rangking peningkatan kinerja maka dilakukan uji statistik. Uji statistik yang digunakan adalah uji *Friedman*. Tabel 7 menunjukkan peringkat pengukuran kinerja *Average Gain* (AG) jika dibandingkan dengan *Classification And Regression Tree* (CART), *Iterative Dichotomiser 3* (ID3), C4.5.

Tabel 7. Peringkat Pengukuran Kinerja Pada CART, ID3, C4.5 dan AG

Kinerja	Mean Rank			
	CART	ID3	C4.5	AG
Akurasi	2,30	2,70	1,80	3,20
Sensitivitas	3,30	2,60	2,10	2,00
Specificity	1,90	2,90	2,20	3,00
F-Measure	2,50	3,10	1,80	2,60
G-Mean	1,90	3,10	2,00	3,00

Berdasarkan hasil uji *Friedman*, akurasi dan *specificity* AG berada pada peringkat 1.

Tabel 8 menunjukkan peringkat pengukuran kinerja *Average Gain* yang di-*pruning* (AG_*Pruning*) jika dibandingkan dengan *Classification And Regression Tree* (CART), *Iterative Dichotomiser 3* (ID3), C4.5 dan *Average Gain* (AG).

Tabel 8. Peringkat Pengukuran Kinerja Pada CART, ID3, C4.5, AG dan AG_*Pruning*

Kinerja	Mean Rank				
	CART	ID3	C4.5	AG	AG_*Pruning*
Akurasi	2,50	2,70	1,80	3,40	4,60
Sensitivitas	3,70	3,00	2,30	2,20	3,80
Specificity	2,10	3,10	2,40	3,20	4,20
F-Measure	2,90	3,30	1,80	2,80	4,20
G-Mean	2,30	3,50	2,40	3,40	3,40

Berdasarkan hasil uji *Friedman*, kinerja AG_*Pruning* yang meliputi akurasi, sensitivitas, *specificity*, *F-Measure* dan *G-Mean* berada pada peringkat 1.

5 KESIMPULAN

Pada pengukuran akurasi dan sensitivitas AG dapat meningkatkan kinerja algoritma C4.5 dan melalui uji *Friedman* AG berada di peringkat 1. Pada penelitian ini, pengukuran akurasi, sensitivitas, *specificity*, *F-Measure* dari model AG_*Pruning* menunjukkan bahwa AG_*Pruning* dapat meningkatkan kinerja algoritma C4.5. Berdasarkan hasil uji *Friedman* model AG_*Pruning* menunjukkan peningkatan kinerja dan berada di peringkat 1 dibanding CART, ID3, C4.5 dan AG. Model AG_*Pruning* juga menghasilkan pohon keputusan (*Decision Tree*) yang lebih kecil dibanding CART, ID3, C4.5 dan AG. Hasil penelitian ini menunjukkan bahwa *threshold pruning* dan *cost complexity pruning* dapat mengatasi permasalahan *over-fitting*.

REFERENSI

Abellán, J. (2013). Ensembles of decision trees based on imprecise probabilities and uncertainty measures, *14*, 423–430.

C. Sammut, G. W. (2011). *Encyclopedia of Machine Learning*. (C. Sammut & G. I. Webb, Eds.). Boston, MA: Springer US. doi:10.1007/978-0-387-30164-8

Duchessi, P., & Lauría, E. J. M. (2013). Decision tree models for profiling ski resorts' promotional and advertising strategies and

the impact on sales. *Expert Systems with Applications*, *40*(15), 5822–5829. doi:10.1016/j.eswa.2013.05.017

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques*. (Springer, Ed.) (12th ed., Vol. 12). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-19721-5

Han, Jiawei; Kamber, Micheline; Pei, J. (2012). *Data Mining Concepts and Techniques*. *Morgan Kaufmann* (Third Edit., Vol. 40, p. 9823). Morgan Kaufmann Publishers. doi:10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

Larose, D. T. (2005). *Discovering Knowledge in Data*. United States of America: John Wiley & Sons, Inc.

Larose, D. T. (2006). *Data Mining Methods And Models*. New Jersey: A John Wiley & Sons, Inc Publication.

Mantas, C. J., & Abellán, J. (2014). Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, *41*(10), 4625–4637. doi:10.1016/j.eswa.2014.01.017

Otero, F. E. B., Freitas, A. A., & Johnson, C. G. (2012). Inducing decision trees with an ant colony optimization algorithm. *Applied Soft Computing*, *12*(11), 3615–3626. doi:10.1016/j.asoc.2012.05.028

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. The Morgan Kaufmann Publishers.

Rokach, L., & Maimon, O. (2005). Decision Tree. *Data Mining and Knowledge Discovery Handbook*, pp 165–192. doi:10.1007/978-0-387-09823-4_9

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, *40*(15), 5916–5923. doi:10.1016/j.eswa.2013.05.021

Setsirichok, D., Piroonratana, T., Wongseree, W., Usavanarong, T., Paulkhaolarn, N., Kanjanakorn, C., … Chaiyaratana, N. (2012). Classification of complete blood count and haemoglobin typing data by a C4.5 decision tree, a naïve Bayes classifier and a multilayer perceptron for thalassaemia screening. *Biomedical Signal Processing and Control*, *7*(2), 202–212. doi:10.1016/j.bspc.2011.03.007

T Warren Liao, E. T. (2007). *Recent Advances in Data Mining of Enterprise Data : Algorithms and Applications* (Vol.6 ed.). World Scientific Publishing Co.

Ture, M., Tokatli, F., & Kurt, I. (2009). Using Kaplan–Meier analysis together with decision tree methods (C&RT, CHAID, QUEST, C4.5 and ID3) in determining recurrence-free survival of breast cancer patients. *Expert Systems with Applications*, *36*(2), 2017–2026. doi:10.1016/j.eswa.2007.12.002

Wang, T., Qin, Z., Jin, Z., & Zhang, S. (2010). Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. *Journal of Systems and Software*, *83*(7), 1137–1147. doi:10.1016/j.jss.2010.01.002

Wang, T., Qin, Z., Zhang, S., & Zhang, C. (2012). Cost-sensitive classification with inadequate labeled data, *37*, 508–516. doi:10.1016/j.is.2011.10.009

Zhang, S. (2012). Decision tree classifiers sensitive to heterogeneous costs. *Journal of Systems and Software*, *85*(4), 771–779. doi:10.1016/j.jss.2011.10.007

BIOGRAFI PENULIS

Erna Sri Rahayu. Memperoleh gelar M.Kom dari Universitas Dian Nuswantoro, Semarang. Menjadi pendidik di SMP Negeri 1 Pabelan dengan mata pelajaran yang diampu Teknologi Informasi dan Komunikasi (TIK). Minat penelitian pada saat ini di bidang data mining.

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Catur Supriyanto. Memperoleh gelar Master dari University Teknikal Malaysia Melaka (UTEM), Malaysia. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Minat penelitiannya pada bidang information retrieval, machine learning, soft computing dan intelligent system.