Integrasi Density Based Feature Selection dan Adaptive Boosting dalam Mengatasi Ketidakseimbangan Kelas

Sudarto[1], Muhammad Zarlis[2], Pahala Sirait[3]
[1,2]Universitas Sumatera Utara, Jl. Universitas No. 24 A Medan 20215, telp/fax. (061) 8221379
[1,3]STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789
[1,2]Program Studi Magister Teknik Informatika, Universitas Sumatera Utara
[1]Jurusan Sistem Informasi, STMIK Mikroskil, Medan
[3]Jurusan Teknik Informatika, STMIK Mikroskil, Medan
[1]sudarto@mikroskil.ac.id, [2]m.zarlis@yahoo.com, [3]pahala@mikroskil.ac.id

Abstrak

Ketidakseimbangan kelas (Class Imbalance) dari dataset antara dua kelas yang berbeda yaitu kelas mayoritas dan kelas minoritas, berpengaruh pada algoritma C4.5 yang cenderung menghasilkan akurasi prediksi yang baik pada kelas mayoritas tetapi menjadi tidak konduktif dalam memprediksi contoh kelas minoritas, sehingga nilai hasil akurasi pengklasifikasian (classifier) C4.5 menjadi tidak optimal. Untuk mengurangi pengaruh ketidakseimbangan kelas pada pengklasifikasi C4.5, maka perlu dilakukan dengan menerapkan kombinasi dari metode seleksi fitur yaitu algoritma Adaptive Boosting (Adaboost) dan metode Density Based Feature Selection (DBFS). Penerapan algoritma adaboost dalam seleksi fitur dilakukan untuk memberi bobot pada setiap fitur yang direkomendasikan, sehingga ditemukan fitur yang merupakan classifier yang kuat, sedangkan DBFS berfokus dalam mengidentifikasi kelas minoritas dan mengevaluasi dampak dari sebuah fitur yang bermanfaat berdasarkan rangking fitur agar dapat direkomendasikan pada classifier C4.5 dalam proses pengklasifikasian. Hasil penelitian menunjukkan bahwa, kinerja akurasi pengklasifikasi C4.5 pada dataset mahasiswa lulusan dengan mengkombinasikan DBFS sebelum proses adaboost, dengan pengaturan nilai confidence level 0,50 dan 30 fold cross-validation, menunjukkan tingkat akurasi klasifikasi yang relatif lebih baik dalam penanganan ketidakseimbangan kelas.

Kata kunci— *Class-imbalance, Classifier-C4.5, Adaboost, DBFS, Fold Cross-Validation*

Abstract

The Class Imbalance of dataset between two different class are majority and minority class, which impact on the algorithm C 4.5 that tend to produce good prediction accuracy on the class majority but not be conductive in predicting instances of minority class, so the value of accuracy of classification results C4.5 not optimal. To reduce the influence of class imbalance in the classifier C4.5, is applying a combination of feature selection methods namely Adaptive Boosting (Adaboost) algorithms and Density Based Feature Selection (DBFS) method. Application of adaboost algorithm in feature selection done to give weights to each recommended feature, so will found a feature with strong classifier, While the DBFS focusing in identifying minority classes and evaluating the impact of a useful features based on rank features, then it can be recommended classifier C 4.5 in the process of classification. The results study , shows the performance accuracy classifier C 4.5 on a dataset of student graduates with combines DBFS before the process of adaboost, value setting of the confidence level 0.50 and 30 fold cross-validation, indicates the level of accuracy thefdd classification of the relatively better in handling the class imbalance.

Keywords— *Class-imbalance, Classifier-C4.5, Adaboost, DBFS, Fold Cross-Validation*

1. PENDAHULUAN

Proses klasifikasi merupakan salah satu tugas dalam *datamining* yang digunakan untuk meramalkan sebuah nilai dari sekumpulan data. Salah satu tantangan terbesar dalam penelitian klasifikasi pada *datamining* adalah masalah ketidakseimbangan kelas yang umumnya ditemukan dalam

aplikasi dunia nyata [1]. Ketidakseimbangan kelas (*class imbalance*) terjadi dalam jumlah *training data* antara dua kelas yang berbeda. Satu kelas memiliki jumlah data yang besar (mayoritas) sedangkan kelas yang lain memiliki jumlah data yang minoritas [2]. Ketidakseimbangan kelas (*class imbalance*) biasanya cenderung menyebabkan *overlapping*, kurangnya data yang representatif, *small disjuncts* atau adanya *noise* data dan *borderlin*e *instances* yang membuat proses belajar *classifier* sulit [3]. Selain itu juga bahwa ketidakseimbangan kelas (*class imbalance*) dan *noise* dapat berpengaruh pada kualitas data dalam hal kinerja klasifikasi [4]. Ini menunjukkan ketidakseimbangan kelas (*class imbalance*) menyebabkan terjadinya *misclassification* [5].

Permasalahan ketidakseimbangan kelas (class imbalance) juga dapat menyebabkan akurasi dalam klasifikasi dengan algoritma C4.5 tidak optimal [6]. Hasil klasifikasi pada algoritma C4.5 dalam predikat kelulusan mahasiswa tepat waktu dengan grade *cumlaude* bisa diperoleh dengan syarat utama adalah pernah menjadi asisten semasa kuliah , berasal dari jurusan IPA semasa SMA , rerata SKS per semester 18 dan berjenis kelamin wanita [7]. Sebagian besar kasus data yang telah dilakukan untuk klasifikasi mahasiswa tepat waktu dan tidak tepat waktu adalah tidak seimbang, yang berarti bahwa hanya sebagian kecil mahasiswa tidak tepat waktu dan sebagian besar tepat waktu.

Ada tiga pendekatan untuk menangani dataset tidak seimbang (*unbalanced*), pendekatan pada level data mencakup berbagai teknik *resampling* dan sintesis data untuk memperbaiki kecondongan distribusi kelas *training data*. Pada tingkat algoritmik, metode utamanya adalah menyesuaikan operasi algoritma yang ada untuk membuat pengklasifikasi (*classifier*) agar lebih konduktif terhadap klasifikasi kelas minoritas [8]. Sedangkan pada pendekatan menggabungkan atau memasangkan (*ensemble*) metode, ada dua algoritma *ensemble-learning* paling populer, yaitu *boosting* dan *bagging* [9]. Pada pendekatan algoritma dan *ensemble* memiliki tujuan yang sama, yaitu memperbaiki algoritma pengklasifikasi tanpa mengubah data, sehingga dapat dianggap ada 2 pendekatan saja, yaitu pendekatan level data dan pendekatan level algoritma [10]. Karena masalah ketidakseimbangan kelas biasanya disertai dengan permasalahan dari dataset berdimensi tinggi, teknik *sampling* dan metode algoritma tidaklah cukup menangani ketidakseimbangan kelas (*class imbalance*). Menerapkan seleksi fitur (*feature selection*) adalah tindakan penting yang perlu dilakukan dalam menangani ketidakseimbangan kelas (*class imbalance*) dari dataset berdimensi tinggi [11].

Untuk dataset tidakseimbang (*imbalance*), metode seleksi fitur juga harus fokus pada atribut yang membantu dalam identifikasi kelas minoritas [1]. Selain itu, kinerja metode seleksi fitur berkembang ketika rasio ketidakseimbangan meningkat. Di berbagai rasio ketidakseimbangan kelas, metode DBFS (*Density Based Feature Selection*) melebihi metode saingan seleksi fitur lainnya terutama ketika lebih dari 0,5 % dari fitur yang dipilih untuk tugas klasifikasi. Peningkatan ini lebih nyata sesuai dengan evaluasi statistik AUC (*area under curve*) terutama dengan rasio ketidakseimbangan yang tinggi [12]. Ketidakseimbangan kelas juga dapat ditangani dengan adanya pendekatan untuk menggabungkan seleksi fitur dengan proses *boosting*. Fokusnya pada dua skenario yang berbeda yaitu seleksi fitur dilakukan sebelum proses *boosting* dan seleksi fitur yang dilakukan dalam proses *boosting*. Hasilnya menunjukkan bahwa melakukan seleksi fitur dalam *boosting* umumnya lebih baik daripada menggunakan seleksi fitur sebelum proses *boosting* [13]. Salah satu contoh algoritma *boosting* adalah algoritma *adaptive boosting (adaboost)*, telah dilaporkan sebagai meta-teknik untuk mengatasi masalah ketidakseimbangan kelas (*class imbalance*) [14]. AdaBoost secara teoritis dapat secara signifikan digunakan untuk mengurangi kesalahan dari beberapa algoritma pembelajaran yang secara konsisten menghasilkan kinerja pengklasifikasi yang lebih baik. Kinerja *adaBoost* lebih baik dari *random forest* untuk prediksi performansi siswa dan dapat memperbaiki kinerja *classifier* [15].

Dalam penelitian ini, dilakukan penanganan ketidakseimbangan kelas yang berfokus pada fitur atau atribut yang membantu mengidentifikasi ketepatan akurasi kelas minoritas agar dapat mempengaruhi hasil dalam pengklasifikasian. Dengan demikian, maka diperlukan suatu model untuk penanganan ketidakseimbangan kelas (*class imbalance*) dengan menggunakan metode *Density Based Feature Selection* (DBFS) dan *Adaptive boosting* pada algoritma klasifikasi C4.5 serta pengukuran peningkatan kinerja dari sudut pandang akurasi, presisi dan sensitivitas (*recall*) melalui perbandingan algoritma klasifikasi C4.5 dengan menggunakan metode DBFS dan *Adaboost*.

2. METODE PENELITIAN

2.1. Pengumpulan Data

Pada penelitian ini, menggunakan dataset kelulusan mahasiswa STMIK Mikroskil pada program studi Sistem Informasi tahun ajaran 2004, 2005, 2006, berupa data akademik dan data non-akademik yang memiliki struktur kurikulum yang sama . Data kelulusan mahasiswa dimaksudkan untuk mencari dan membentuk pola perolehan status akademik yang akan digunakan untuk memprediksi kelulusan mahasiswa. Data kelulusan mahasiswa yang bisa digunakan sebagai dataset diperoleh dengan melakukan query data dari beberapa database SIPT (Sistem Informasi Perguruan Tinggi) STMIK Mikroskil yang dikelolah oleh Unit Pelaksana Teknis Pusat Sistem Informasi (UPTPSI) .

2.2. Pengolahan Data

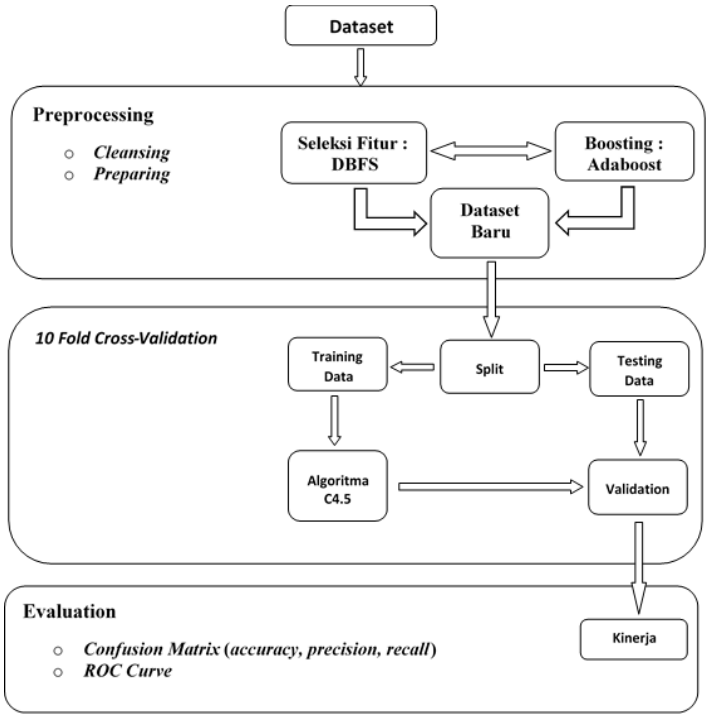
Dataset hasil dari penggabungan beberapa atribut atau fitur terdapat permasalahan missing value sebanyak 132 instances dari 1086 instances. Dengan melakukan imputasi dan menghilangkannya merupakan cara penanganan missing value pada data tersebut, sehingga instances yang akan diolah menjadi sebanyak 954. Nilai atribut atau fitur pada dataset yang digunakan berupa angka dan simbol dikonversikan kedalam bentuk kategorikal yaitu data nominal yang tidak dapat dinyatakan bahwa satu kategori tidak lebih baik dari kategori lainnya dan nilai – nilainya tidak dapat diurutkan. Rincian spesifikasi fitur atau atribut seperti pada Tabel 1.

Tabel 1. Spesifikasi fitur Dataset Kelulusan Mahasiswa Lulusan

No	Atribut	Nilai Nominal Dan Kategori	Jumlah	Persentase (%)
1	Asal Sekolah	1. Dalam Kota	596	62.50%
		2. Luar Kota	358	37.50%
2	Jenis Kelamin	1. Laki-Laki	609	63.80%
		2. Wanita	358	36.20%
3	Shift Kuliah	1. Pagi	698	73.20%
		2. Sore	256	26.80%
4	Indeks Prestasi Semester 1	1. > = 3.00	408	42.80%
		2. > = 2.50	284	29.80%
		3. < 2.50	262	27.50%
5	Indeks Prestasi Semester 2	1. > = 3.00	357	37.40%
		2. > = 2.50	334	35.00%
		3. < 2.50	263	27.60%
6	Indeks Prestasi Semester 3	1. > = 3.00	405	42.50%
		2. > = 2.50	287	30.10%
		3. < 2.50	262	27.50%
7	Indeks Prestasi Semester 4	1. > = 3.00	411	43.10%
		2. > = 2.50	302	31.70%
		3. < 2.50	241	25.30%
8	Indeks Prestasi Semester 5	1. > = 3.00	341	35.70%
		2. > = 2.50	332	34.80%
		3. < 2.50	281	29.50%
9	Indeks Prestasi Semester 6	1. > = 3.00	431	45.20%
		2. > = 2.50	331	34.70%
		3. < 2.50	192	20.10%
10	Indeks Prestasi Semester 7	1. > = 3.00	380	39.80%
		2. > = 2.50	314	32.90%
		3. < 2.50	269	27.30%
11	Indeks Prestasi Semester 8	1. > = 3.00	394	41.30%
		2. > = 2.50	378	39.60%
		3. < 2.50	182	19.10%
12	Indeks Prestasi Kumulatif	1. > = 3.50	69	62.70%
		2. > = 2.75	598	30.10%
		3. < 2.75	287	0.72%
13	Rerata SKS	1. > 20	619	64.90%
		2. < 20	335	35.10%
14	Status Akademik	1. Tepat Waktu	838	87.80%
		2. Tidak Tepat Waktu	116	22.20%

2.3. Model yang Diusulkan

Model yang diusulkan dalam menangani ketidakseimbangan kelas yaitu dengan mengintegrasikan pendekatan seleksi fitur dan pendekatan algoritma, meliputi penerapan metode DBFS dan Adaboost untuk meningkatkan kinerja pengklasifikasi C4.5. kerangka kerja model yang diusulkan ditunjukkan pada gambar 1.

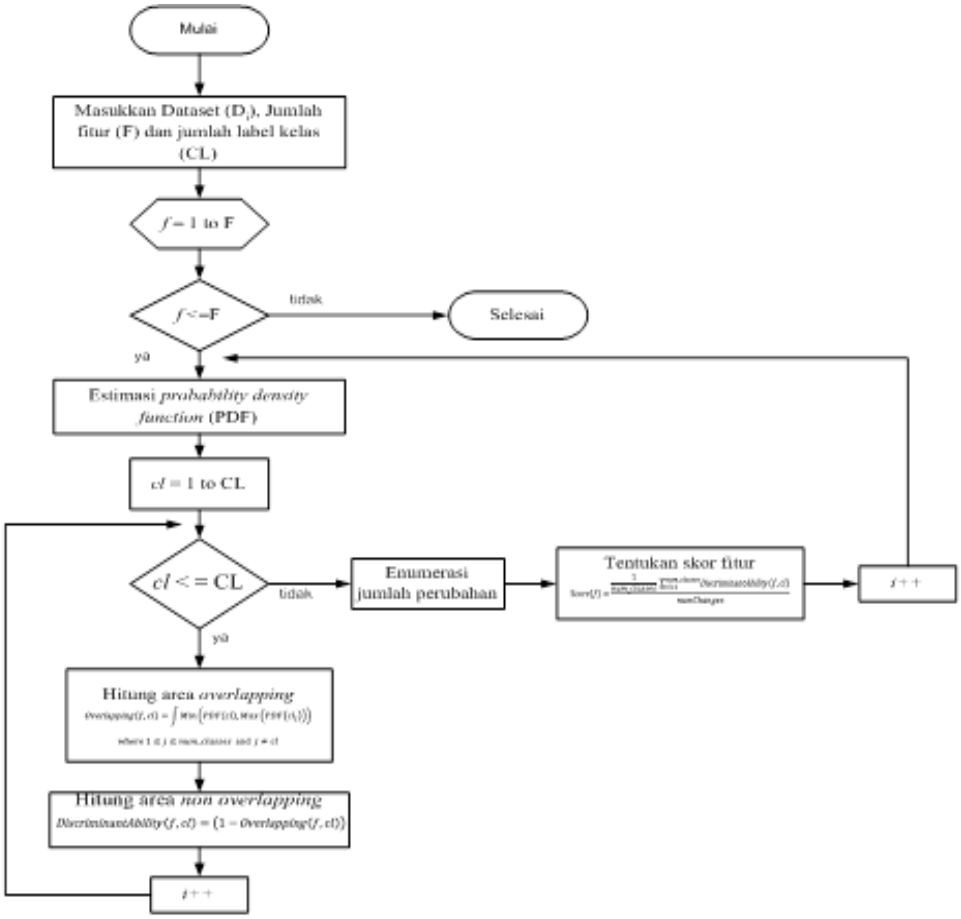


Gambar 1. Kerangka Kerja Model yang diusulkan

Pada model yang diusulkan, pengolahan awal data dibersihkan dan dipilah. Selanjutnya dalam penanganan ketidakseimbangan kelas pada dataset kelulusan mahasiswa akan dilakukan dengan menerapkan metode seleksi fitur DBFS dan proses boosting yaitu adaboost. Dataset kelulusan mahasiswa yang baru dibagi menjadi *X* sesuai nilai validasi (*X-fold cross validation*), satu bagian (*1/X*) digunakan sebagai data uji (*testing*) sisanya digunakan sebagai data latih (*training*). Selanjutnya data *training* diproses dengan metode pengklasifikasi C4.5 dan kemudian diuji dengan data uji melalui proses validasi. Hasil validasi digunakan untuk mengukur kinerja masing – masing model.

Sasaran metode DBFS adalah mengevaluasi dampak dari sebuah atribut atau fitur yang bermanfaat dan fokus pada fitur yang membantu dalam mengidentifikasi kelas minoritas [12]. Prosedur penanganan seleksi fitur dengan metode DBFS ditunjukkan pada gambar 2.

Gambar 2. menunjukkan *flowchart* metode DBFS dengan sejumlah masukan berupa jumlah fitur dan jumlah label kelas pada dataset. Iterasi dilakukan berdasarkan jumlah fitur pada masing-masing label kelas. Selama perulangan maka dihitung nilai estimasi *probability density function* (PDF) dari fitur disetiap label kelas. Selanjutnya, prosedur untuk menentukan perangkingan fitur dimulai dengan penghitungan nilai area *overlapping* setiap fitur masing – masing label kelas. Untuk penghitungan jumlah nilai area *overlapping* menggunakan estimasi PDF untuk setiap fitur dari masing – masing label kelas. Penghitungan area *non overlapping* berdasarkan nilai *discriminant ability* untuk setiap fitur dari masing – masing label kelas agar dapat ditemukan fitur yang andal dalam mengklasifikasikan instance kelas. Jika nilai *overlapping* dan *discriminant ability* setiap fitur dari masing – masing label kelas terpenuhi, langkah berikutnya mengenumerasi perubahan jumlah nilai estimasi PDF setiap fitur dari satu label kelas ke label kelas lainnya. Jumlah nilai perubahan dan rata – rata nilai *discriminant ability* setiap fitur dari masing – masing label kelas dihitung untuk menentukan skor fitur. Sehingga, semakin tinggi skor dari sebuah fitur maka peringkatnya akan semakin rendah.

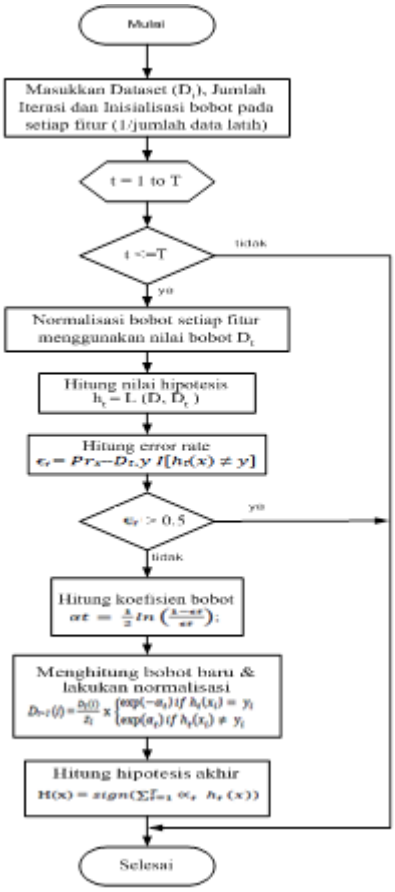


Gambar 2. Flowchart Metode DBFS

Selanjutnya, Penerapan algoritma *adaboost* dalam seleksi fitur dilakukan untuk memberi bobot pada setiap fitur yang direkomendasikan, sehingga ditemukan fitur yang merupakan *classifier* yang kuat dan untuk meningkatkan kinerja klasifikasi terhadap suatu algoritma *classifier*. Adapun *flowchart* Algoritma *Adaboost* ditunjukkan pada gambar 3.

Pada algoritma *adaboost*, dimana masukan berupa sejumlah data *training* dan data *testing* serta jumlah iterasi. Langkah awal dilakukan inisialisasi bobot untuk setiap fitur sebesar 1 dibagi dengan jumlah data *training*. Selanjutnya dilakukan perulangan sesuai masukan jumlah iterasi. Selama iterasi dilakukan, normalisasikan distribusi setiap fitur data *training* agar sama dengan 1. Hitung nilai hipotesis *weak classifier* dan nilai kesalahannya (*error rate*) dari setiap fitur data *training* , jika nilai kesalahannya lebih besar dari 0,5 maka iterasi dihentikan. Jika tidak lebih besar dari 0,5 maka hitung ulang koefisien kesalahan dan faktor normalisasi agar bobot baru bernilai antara -1 sampai 1.

Proses perhitungan dilakukan sampai jumlah iterasi tercapai atau nilai kesalahan (*error rate*) lebih dari 0,5. Setelah perulangan selesai, *strong classifier* akan didapatkan dan merupakan gabungan hasil voting dari mayoritas pembobotan dari semua *weak classifier* yang didapat dari setiap iterasi . jika hasil *strong classifier* lebih kecil dari 0 maka dikategorikan sebagai fitur yang tidak relevan, jika lebih besar dari 0 sampai dengan 1 maka dikategorikan sebagai fitur yang direkomendasikan untuk pengklasifikasian.



Gambar 3. Flowchart Algoritma Adaboost

2.4. Evaluasi dan Validasi

Validasi dilakukan menggunakan *stratified k-fold cross validation* . Proses *stratification* akan dilakukan terlebih dahulu sebelum proses *cross validation* untuk dapat mereduksi varian estimasi. Metode evaluasi standard yaitu *stratified 10-fold cross-validation* adalah pilihan terbaik untuk mendapatkan hasil validasi yang akurat dari data *testing* dan data *training*. *10-fold cross-validation* akan mengulang pengujian sebanyak 10 kali dan hasil pengukuran adalah nilai rata-rata dari 10 kali pengujian. Keuntungan metode ini, menghindari *overlapping* pada data *testing*. Dimana *test set* bersifat *mutually exclusive* dan secara efektif mencakup keseluruhan dataset [16].

Pengukuran kinerja algoritma pengklasifikasi dilakukan dengan menggunakan *confusion matrix*. Dimana *confunsion matrix* diperoleh dari proses validasi. Mengevaluasi kinerja algoritma pengklasifikasi umumnya menggunakan hasil keseluruhan pada pengujian dataset [8]. *Confusion matrix* dapat membantu menunjukkan rincian kinerja pengklasifikasi dengan memberikan informasi jumlah fitur suatu kelas yang diklasifikasikan dengan tepat dan tidak tepat [17]. Setelah dibuat *confusion matrix*, selanjutnya dihitung nilai akurasi, sensitivitas atau *recall* dan presisi. Formulasi perhitungan yang digunakan adalah sebagai berikut :

$$Accuracy = \frac{TP+TN}{(TP+FN)+(FP+TN)}$$
$$Recall = \frac{TP}{TP+FN} \quad (1)$$
$$Precision = \frac{TP}{TP+FP}$$

Hasil pengukuran kinerja model yang diperoleh, digunakan untuk membandingkan antara model dasar yaitu algoritma C4.5 dengan model yang dibentuk menggunakan kombinasi DBFS dan *adaboost*. Kualitas model yang dihasilkan dapat dilihat berdasarkan nilai *Area Under curve* (AUC) dan *Receiver Operating Character* (ROC) *curve*. Evaluasi dengan *ROC curve* secara teknis menggambarkan grafik dua dimensi atau *trade-off* antara *true positive* (TP) dengan *false positive* (FP). Hasil *ROC curve* akan digunakan untuk menemukan nilai AUC, dimana nilai AUC digunakan untuk menentukan klasifikasi keakuratan pengujian diagnostik. Ukuran AUC dihitung sebagai daerah kurva ROC dengan persamaan 2.

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \quad (2)$$

3. HASIL DAN PEMBAHASAN

3.1. Hasil

Penelitian ini menghasilkan keluaran yang dapat dianalisis untuk menghasilkan informasi dan pengetahuan yang berguna melalui proses dan hasil eksperimen yang sudah dilakukan dengan menggunakan aplikasi *rapidminer studio* 6.5 dan *Xampp* 2.4. Dimana eksperimen pertama yaitu menguji model prediksi pengklasifikasi C4.5 berdasarkan dataset kelulusan mahasiswa. Nilai *gain ratio* tertinggi bukan *gain* (*a*) digunakan dalam pemilihan atribut (seleksi fitur) *test* untuk menghindari bias terhadap atribut yang memiliki nilai unik [18]. Nilai *gain ratio* yang diperoleh dari perhitungan manual menunjukkan atribut rerata SKS akan dijadikan sebagai atribut *root node* (simpul akar) pada C.45.

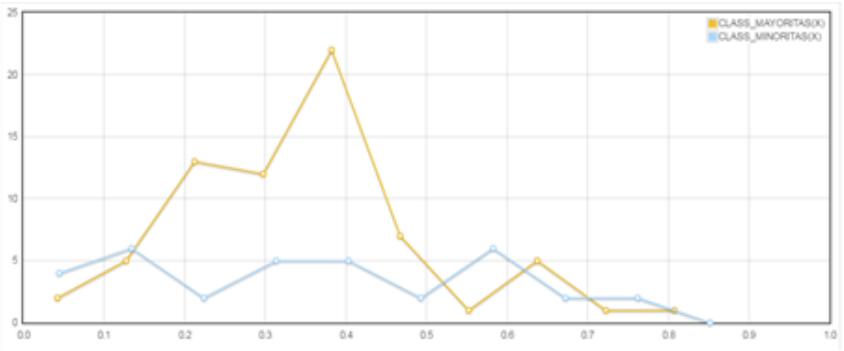
Pada pengklasifikasi C4.5, untuk mengatasi permasalahan *overfitting* menggunakan metode pemangkasan (*prunning*) yaitu *pre prunning* untuk menghasilkan model analisis yang optimal [19]. Pemangkasan (*pruning*) pada pohon (*tree*) yang dihasilkan dilakukan berdasarkan nilai *confidence level* yang mengecil [20]. Oleh karena itu akan dilakukan beberapa pengujian dengan nilai *confidence level* yang diubah – ubah mulai dari 0.95, 0.50, 0.25 dan 0.10.

Tabel 2. Hasil Pengukuran C45 tanpa Seleksi fitur

Validasi / Confidence	TP	FP	FN	TN	Akurasi	Recall		Presisi	
						TW	TTW	TW	TTW
5 Fold									
0.25	798	40	62	54	89.31 %	92.79 %	57.45 %	95.23 %	46.55 %
0.50	797	41	62	54	89.20 %	92.78 %	56.84 %	95.11 %	46.55 %
10 Fold									
0.25	790	48	63	53	88.37 %	92.97 %	55.45 %	94.63 %	48.28 %
0.50	790	48	63	53	88.37 %	92.61 %	52.48 %	94.27 %	45.69 %
20 Fold									
0.25	803	35	53	63	90.78 %	93.81 %	64.29 %	95.82 %	54.31 %
0.50	801	37	52	64	90.67 %	93.90 %	63.37 %	95.58 %	55.17 %
30 Fold									
0.25	794	44	55	61	89.61 %	93.52 %	58.10 %	94.75 %	52.59 %
0.50	792	46	54	62	89.51 %	93.62 %	57.41 %	94.51 %	53.45 %

Dari eksperimen pertama yang dilakukan secara iteratif untuk tabel 2. nilai *confidence level* 0,95 dan 0,50 maupun 0,25 dan 0,10 dalam setiap 5, 10, 20, 30 *fold cross-validation* memiliki kecenderungan hasil yang sama.

Pada eksperimen kedua dimulai dengan menerapkan seleksi fitur menggunakan DBFS dalam penanganan ketidakseimbangan kelas dan untuk meningkatkan akurasi pengklasifikasi C4.5. Dari hasil estimasi *Probability Density Function* (PDF) maka menimbulkan area *overlapping* di setiap kelas untuk beberapa ruang fitur, atau kadang-kadang bahkan di semua ruang fitur. *Overlapping* terjadi pada area dimana jumlah nilai estimasi PDF dari setiap fitur pada kelas minoritas lebih besar dari kelas mayoritas ditunjukkan pada gambar 4.



Gambar 4. Area Overlapping pada setiap kelas

Dari hasil perhitungan nilai *overlapping* maka ditentukan nilai ketentuan diskriminan (*discriminant ability*). Rata – rata nilai ketentuan diskriminan (*discriminant ability*) dari setiap fitur pada masing – masing kelas dibagi dengan jumlah perubahan dijadikan acuan dalam menghitung skors setiap fitur. Dimana fitur dengan skor terkecil merupakan peringkat tertinggi dalam perangkingan dan dapat direkomendasikan dalam proses pengklasifikasian C4.5. Berikut peringkat dari seleksi fitur dengan menggunakan DBFS ditunjukkan pada tabel 3.

Tabel 3. Peringkat Fitur

Peringkat	Atribut / Fitur	Skor
1	Rerata SKS	0.6167
2	Asal Sekolah	0.6427
3	Shift Kuliah	0.6751
4	Jenis Kelamin	0.7004
5	IP Sem 2	0.7758
6	IP Sem 5	0.7918
7	IP Sem 1	0.8034
8	IP Sem 8	0.8185
9	IP Sem 7	0.8214
10	IP Sem 4	0.8228
11	IP Sem 3	0.8228
12	IP Sem 6	0.8626
13	IPK	0.9073

Hasil perangkingan fitur dataset kelulusan mahasiswa akan direkomendasikan dengan menggunakan persentase moderat. Parameter untuk jumlah % (persentase) proporsi dari fitur yang akan diproses pada pengklasifikasi C4.5 yaitu 40 % (5 fitur), 60 % (8 fitur) dan 70 % (9 fitur) adalah *milestone* dari angka – angka moderat antara 0 % sampai dengan 100 % (Jamhari et al, 2014). Hasil pengukuran C4.5 dengan DBFS persentase 70 % ditunjukan pada tabel 4 .

Tabel 4. Hasil Pengukuran C4.5 dengan DBFS (70 %)

Validasi / Confidence	TP	FP	FN	TN	Akurasi	Recall		Presisi	
						TW	TTW	TW	TTW
5 Fold									
0.25	799	39	65	51	89.10 %	92.48 %	56.67 %	93.35 %	43.97 %
0.50	796	42	63	53	88.99 %	92.67 %	55.79 %	94.99 %	45.69 %
10 Fold									
0.25	804	34	61	55	90.04 %	92.95 %	61.80 %	95.94 %	47.41 %
0.50	804	34	59	57	90.25 %	93.16 %	62.64 %	95.94 %	49.14 %
20 Fold									
0.25	804	34	62	54	89.94 %	92.84 %	61.36 %	95.94 %	46.55 %
0.50	803	35	59	57	90.15 %	93.16 %	61.96 %	95.82 %	49.14 %
30 Fold									
0.25	811	27	58	58	91.09 %	93.33 %	68.24 %	96.78 %	50.00 %
0.50	809	29	56	60	91.09 %	93.53 %	67.42 %	96.54 %	51.72 %

Diketahui bahwa pengujian dengan menerapkan seleksi fitur DBFS persentase moderat 70 % yang direkomendasikan dari keseluruhan fitur serta nilai *confidence level* 0.25 dan 30 *fold - cross validation* menghasilkan kinerja akurasi, *recall* dan presisi tertinggi.

Pada eksperimen ketiga, dimulai dengan menerapkan seleksi fitur menggunakan DBFS. Hasil dari perangkingan fitur pada DBFS, maka persentase moderat sebesar 70 % hasil dari eksperimen kedua memiliki nilai akurasi tertinggi akan direkomendasikan pada algoritma *adaboost*. Dari hasil seleksi fitur dengan *adaboost* ditunjukkan pada tabel 4, bahwa nilai H_x sama dengan +1 adalah fitur yang layak direkomendasikan dan nilai H_x sama dengan -1 adalah fitur yang tidak direkomendasikan pada pengklasifikasian. Berikut hasil pengukuran dengan DBFS sebelum *adaboost* ditunjukkan Tabel 5.

Tabel 5. Hasil Seleksi Fitur dengan DBFS sebelum Adaboost

No.	Atribut / Fitur	H_x
1	Rerata SKS	+1
2	Asal Sekolah	+1
3	Shift Kuliah	+1
4	Jenis Kelamin	+1
5	IP Sem 2	+1
6	IP Sem 5	+1
7	IP Sem 1	+1
8	IP Sem 8	-1

Tabel 6. Hasil Pengukuran C4.5 dengan DBFS sebelum Adaboost

Validasi / Confidence	TP	FP	FN	TN	Akurasi	Recall		Presisi	
						TW	TTW	TW	TTW
5 Fold									
0.25	811	27	72	44	89.62 %	91.85%	61.97 %	96.78 %	37.93 %
0.50	807	31	62	54	90.25 %	92.87 %	63.53 %	96.30 %	46.55 %
10 Fold									
0.25	817	21	71	45	90.35 %	92.00 %	68.18 %	97.49 %	38.79 %
0.50	816	22	67	49	90.67 %	92.41 %	69.01 %	97.37 %	42.24 %
20 Fold									
0.25	817	21	68	48	90.67 %	92.32 %	69.57 %	97.49 %	41.38 %
0.50	815	23	63	53	90.98 %	92.82 %	69.74 %	97.26 %	45.69 %
30 Fold									
0.25	818	20	68	48	90.77 %	92.33 %	70.59 %	97.61 %	41.38 %
0.50	818	20	62	54	91.39 %	92.95 %	72.97 %	97.61 %	46.55 %

Tabel 6. menunjukkan, dengan penerapan dua seleksi fitur pada pengklasifikasi C4.5 Akurasi semakin lebih membaik dari eksperimen sebelumnya menjadi 91,39 % pada saat pengujian dilakukan pada 30 *fold-cross validation* dan nilai *confidence level* sebesar 0,50 .

Pada eksperimen keempat dimulai dengan menerapkan seleksi fitur menggunakan *adaboost* untuk melakukan pembobotan pada delapan fitur yang direkomendasikan sehingga yang merupakan *classifier* yang kuat dan selanjutnya dilakukan evaluasi dampak dari sebuah fitur yang bermanfaat berdasarkan rangking fitur menggunakan DBFS. Hasil pengukuran dari perangkingan seleksi fitur pada eksperimen keempat ditunjukkan pada tabel 7.

Tabel 7. Hasil Pengukuran C4.5 dengan DBFS setelah proses Adaboost

Validasi / Confidence	TP	FP	FN	TN	Akurasi	Recall		Presisi	
						TW	TTW	TW	TTW
5 Fold									
0.25	817	21	84	32	88.99 %	90.68 %	60.38 %	97.49 %	27.59 %
0.50	815	23	80	36	89.20 %	91.06 %	61.02 %	97.26 %	31.03 %
10 Fold									
0.25	815	23	72	44	90.05 %	91.88 %	65.67 %	97.26 %	37.93 %
0.50	813	25	70	46	90.05 %	92.07 %	64.79 %	97.02 %	39.66 %
20 Fold									
0.25	813	25	73	43	89.73 %	91.76 %	63.24 %	97.02 %	37.07 %
0.50	811	27	72	44	89.62 %	91.85 %	61.97 %	96.78 %	37.93 %
30 Fold									
0.25	808	30	76	40	88.90 %	91.40 %	57.14 %	96.42 %	34.48 %
0.50	806	32	74	42	88.90 %	91.59 %	56.76 %	96.18 %	36.21 %

Tabel 7 menunjukkan bahwa nilai akurasi tertinggi dengan menerapkan seleksi fitur DBFS setelah *adaboost* pada nilai *confidence level* 0,25 pada 10 *fold cross-validation* yaitu sebesar 90,05 %.

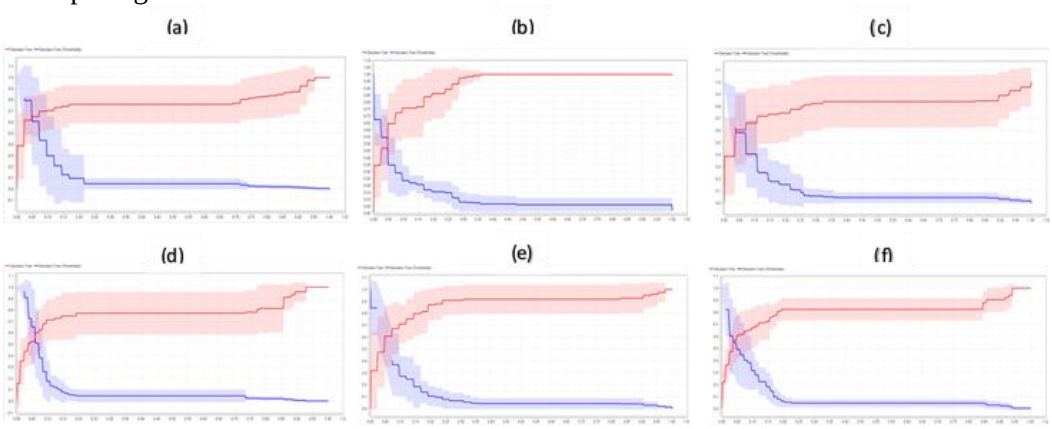
Dari hasil pengukuran beberapa eksperimen yang dilakukan dapat disimpulkan bahwa peningkatan kinerja akurasi, presisi dan sensitivitas (*recall*) model C4.5, dipengaruhi dengan adanya penerapan seleksi fitur pada pengklasifikasi C4.5 dan penentuan jumlah *fold-cross validation* serta nilai *confidence level*. Artinya, jika nilai *fold-cross validation* dan nilai *confidence level* semakin besar maka nilai akurasi, presisi dan sensitivitas (*recall*) cenderung semakin meningkat dan akan semakin bagus kehandalan model C4.5 dalam penanganan ketidakseimbangan kelas pada dataset kelulusan mahasiswa.

3.2. Analisis Kinerja Pengklasifikasian C45

Berdasarkan pengukuran kinerja, diperoleh informasi tingkat kinerja meliputi kemampuan model *classifier* dalam mengklasifikasikan data secara umum (akurasi). Dengan terlebih dahulu menentukan fitur yang bermanfaat menggunakan DBFS dan memberikan bobot pada fitur yang direkomendasikan agar ditemukan fitur yang merupakan *classifier* yang kuat menggunakan *adaboost* pada pengklasifikasi C4.5 mampu meningkatkan kinerja nilai akurasi dan presisi. Pada kinerja nilai sensitivitas (*recall*) meningkat dengan hanya menggunakan DBFS pada pengklasifikasi C4.5 tanpa harus menggunakan adaboost .

Dari hasil pengukuran kinerja pada model kelulusan mahasiswa menunjukkan bahwa kecenderungan jika jumlah *X-fold cross-validation* semakin besar maka kinerja nilai akurasi dan nilai presisi juga meningkat, sementara nilai sensitivitas (*recall*) juga akan meningkat, apabila pengklasifikasi C4.5 tidak dikombinasikan dengan DBFS maupun *adaboost*. Sehingga terdapat model prediksi kelulusan mahasiswa yang memiliki kinerja lebih baik dari pengklasifikasi C4.5. Dengan demikian, kinerja pengklasifikasi C4.5 masih bisa ditingkatkan untuk memperbaiki model prediksi kelulusan mahasiswa.

Penilaian hasil pengukuran kinerja model, dilakukan untuk menentukan model mana yang memiliki kinerja terbaik. Untuk dataset tidak seimbang, akurasi lebih didominasi oleh ketepatan pada data kelas minoritas, maka metrik yang tepat adalah *Area Under the ROC* (*Receiver Operating Characteristic*) *Curve* (Kurva AUROC atau AUC) [21]. Berikut kurva ROC dari beberapa eksperimen ditunjukkan pada gambar 5.



Gambar 5. Kurva ROC (a) C4.5 tanpa seleksi fitur (b) C4.5+DBFS 40% (c) C4.5+DBFS 60% (d) C4.5+DBFS 70% (e) C4.5+DBFS+Adaboost (f) C4.5+Adaboost+DBFS

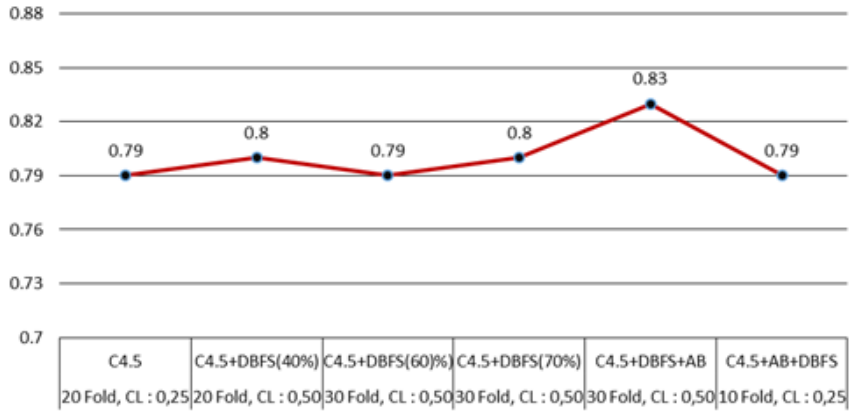
ROC menggambarkan *tradeoff* antara *True Positif* (TP) dan *False Positif* (FP). Pencatatan dalam *ROC* dinyatakan dalam sebuah klausa yaitu semakin rendah titik kekiri (0.0), maka dinyatakan sebagai klasifikasi prediksi mendekati/menjadi negatif, sedangkan semakin keatas titik kekanan (1.1), maka dinyatakan sebagai klasifikasi prediksi mendekati/menjadi positif. Titik dengan nilai 1 dinyatakan sebagai tingkat *True Positif* (*TP*), sedangkan titik dengan nilai 0 dinyatakan sebagai tingkat *False Positive* (*FP*). Pada titik (0.1) merupakan klasifikasi prediksi adalah sempurna karena semua kasus baik positif maupun negatif dinyatakan dengan benar (*True*). Sedangkan untuk (1.0) klasifikasi prediksi semuanya dinyatakan sebagai tidak benar (*False*). Ketika tidak ada kurva ROC yang mendominasi maka AUC akan mempermudah perbandingan kinerja model pengklasifikasi kedalam satu angka [2].

Pada gambar 5.a. menunjukkan kurva ROC kinerja model C4.5 tanpa seleksi fitur pada pengujian 20-*fold* dan *confidence level* 0,25 maka nilai AUC (*Area Under Curve*) sebesar 0,79. Maka tingkat akurasi didiagnosa sebagai klasifikasi sedang (*Fair classification*). Pada gambar 5.b. menunjukkan kurva ROC kinerja model C4.5 dengan seleksi fitur DBFS (40 %) pada pengujian 20-*fold* dan *confidence level* 0,5 maka nilai AUC (*Area Under Curve*) sebesar 0,80. Maka tingkat akurasi didiagnosa sebagai klasifikasi baik (*Good classification*). Gambar 5.c. menunjukkan kurva ROC kinerja model C4.5 dengan seleksi fitur DBFS (60 %) pada pengujian 30-*fold* dan *confidence level* 0,5 maka nilai AUC (*Area Under Curve*) sebesar 0,79. Maka tingkat akurasi didiagnosa sebagai klasifikasi sedang (*Fair classification*). Gambar 5.d. menunjukkan kurva ROC kinerja model C4.5 dengan seleksi fitur DBFS (70 %) pada pengujian 30-*Fold* dan *Confidence Level* 0,5 maka nilai AUC (*Area Under Curve*) sebesar 0,80. Maka tingkat akurasi didiagnosa sebagai klasifikasi baik (*Good classification*). Gambar 5.e. menunjukkan kurva ROC kinerja model C4.5 dengan seleksi fitur DBFS sebelum *adaboost* pada pengujian 30-*Fold* dan *Confidence Level* 0,5 dimana nilai A4UC (*Area Under Curve*) sebesar 0,81. Maka tingkat akurasi didiagnosa sebagai klasifikasi baik (*Good Classification*). Dan gambar 5.f. menunjukkan kurva ROC kinerja model C4.5 dengan seleksi fitur DBFS setelah *adaboost* pada pengujian 10-*fold* dan *confidence level* 0,25 dimana nilai AUC (Area Under Curve) sebesar 0,79. Maka tingkat akurasi didiagnosa sebagai klasifikasi sedang (*Fair Classification*).

AUC merupakan ukuran kinerja yang popular dalam ketidakseimbangan kelas, nilai AUC yang tinggi menunjukkan kinerja yang lebih baik [22]. Rekapitulasi penilaian hasil pengukuran yang lebih baik dari setiap eksperimen untuk nilai dari kinerja model C4.5 tanpa seleksi fitur, kinerja model C4.5 dengan menerapkan seleksi fitur DBFS persentase 40 %, 60%, 70%, kinerja model C4.5 dengan menerapkan seleksi fitur DBFS sebelum *adaboost* dan kinerja model C4.5 dengan seleksi fitur DBFS setelah *adaboost* ditunjukkan pada tabel 8 serta grafik rekapitulasi nilai AUC ditunjukkan Gambar 6.

Tabel 8. Rekapitulasi Nilai AUC

Eksperimen	Model	Nilai AUC	Keterangan
20 Fold, CL : 0,25	C4.5	0,79	*Fair Classification*
20 Fold, CL : 0,50	C4.5+DBFS(40%)	0,8	*Good Classification*
30 Fold, CL : 0,50	C4.5+DBFS(60)%)	0,79	*Fair Classification*
30 Fold, CL : 0,50	C4.5+DBFS(70%)	0,8	*Good Classification*
30 Fold, CL : 0,50	C4.5+DBFS+AB	0,83	*Good Classification*
10 Fold, CL : 0,25	C4.5+AB+DBFS	0,79	*Fair Classification*



Gambar 6. Grafik Rekapitulasi Nilai AUC

Pada gambar 6 menunjukkan bahwa model C4.5+DBFS+AB lebih baik dalam penanganan ketidakseimbangan kelas pada dataset kelulusan mahasiswa dengan tingkat diagnosa adalah klasifikasi baik (*Good Classification*). Sedangkan pengklasifikasi C4.5 tanpa seleksi fitur memiliki tingkat diagnosa klasifikasi sedang (*Fair Classification*). Hal ini menunjukkan bahwa dengan menerapkan

seleksi fitur dan proses *boosting* pada pengklasifikasi C4.5 dapat menangani permasalahan ketidakseimbangan kelas pada dataset kelulusan mahasiswa.

4. KESIMPULAN

Beberapa kesimpulan yang dihasilkan dalam penelitian ini untuk mengurangi pengaruh ketidakseimbangan kelas adalah sebagai berikut :

1. Terdapat model prediksi kelulusan mahasiswa yang memiliki kinerja lebih baik dari pengklasifikasi C4.5, sehingga kinerja pengklasifikasi C4.5 masih bisa ditingkatkan untuk memperbaiki model prediksi kelulusan mahasiswa.
2. Menerapkan seleksi fitur DBFS dapat meningkatkan kinerja pengklasifikasi C4.5 dengan diagnosa klasifikasi baik (*Good Classification*) dalam penanganan ketidakseimbangan kelas, selama proses validasi data *training* dan data *testing* dilakukan dengan metode 30-*fold Cross-Validation* dan nilai *confidence level* dalam proses *prunning* 0,5.
3. Menerapkan seleksi fitur DBFS sebelum proses *Adaboost* dapat meningkatkan kinerja pengklasifikasi C4.5 dengan diagnosa klasifikasi baik (*Good Classification*), selama proses validasi data *training* dan data *testing* dilakukan dengan metode 30-*fold Cross-Validation* dan nilai *confidence level* dalam proses *prunning* 0,5.
4. Integrasi seleksi fitur DBFS setelah proses *Adaboost* cenderung tidak memberikan dampak positif pada peningkatan kinerja pengklasifikasi C4.5 kecuali dalam proses validasi data *training* dan data *testing* dilakukan dengan metode 10-*fold Cross-Validation* dan nilai *confidence level* dalam proses *prunning* 0,5.

5. SARAN

Penelitian ini telah memberikan hasil pengukuran pada masing – masing model prediksi kelulusan mahasiswa untuk mengurangi pengaruh ketidakseimbangan kelas. Beberapa hasil positif dan negatif dari penelitian yang telah dilakukan mungkin akan mendorong adanya penelitian lanjutan dimasa yang akan datang. Beberapa saran terkait penelitian dimasa yang akan datang antara lain :

1. Keadaan *overlapping* terjadi pada area dimana jumlah nilai estimasi *probability density function* (PDF) dari setiap fitur dikelas minoritas lebih besar daripada kelas mayoritas. Penelitian lebih lanjut mungkin dapat dilakukan agar adanya penanganan hubungan *overlapping* minimum setiap kelas secara optimal.
2. Dibutuhkan penelitian lebih lanjut untuk mencari rasio nilai prediksi dan nilai aktual yang optimal pada kelas mayoritas dan kelas minoritas untuk menghasilkan nilai *Root Mean Square Error* (RMSE) yang baik

DAFTAR PUSTAKA

[1] Pant, H. & Srivastava, R. 2015. A Survey on Feature Selection Methods for Imbalanced Datasets. *International Journal of Computer Engineering & Application, vol IX, Issue II* . pp 197 – 204.

[2] Weiss, G. M. (2013). *Foundations of Imbalanced Learning. In H. He, & Y. Ma,* Imbalanced Learning: Foundations, Algorithms, and Applications *(pp. 13-41). New Jersey: John Wiley & Sons.*

[3] Japkowicz, N. (2013). Assessment Metrics for Imbalanced Learning. In H. He, & Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications* (pp. 187-206). New Jersey: John Wiley & Sons.

[4] *Khoshgoftaar, T. M., Gao, K., & Seliya, N. (2010). Attribute Selection and Imbalanced Data: Problems in Software Defect Prediction.* International Conference on Tools with Artificial Intelligence *(pp. 137-144). IEEE Computer Society.*

[5] Zhou, Z. –H., & Liu, X.-Y. (2006). "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng., vol. 18, no. 1*, pp. 63–77

[6] Rahayu, E. S., Wahono, S. R. & Supriyanto, C. (2015). Penerapan Model Average Gain, Threshold Pruning dan Cost Complexity Pruning untuk Split Atribut pada Algoritma C4.5. *Journal of Intelligent System, Vol.1, No.2* . pp 91 – 97.

[7] Yusuf, S. N., (2014). Penerapan Algoritma C4.5 untuk Klasifikasi Predikat Kelulusan Mahasiswa Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) . pp A1 – A6*

[8] Zhang, H., & Wang. Z. (2011). "A Normal Distributions-Based Over-Sampling Approach to Imbalanced Data Classification. *Advanced Data Mining and Applications – 7th International Conference* (pp. 83-96). Beijing, Springer.

[9] *Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets.* Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). 285, *pp. 13-22. Singapore: Springer. doi:10.1007/978-981-4585-18-7_2.*

[10] *Peng, Y., & Yao, J. (2010). AdaOUBoost: Adaptive Over-sampling and Under-sampling to Boost the Concept Learning in Large Scale Imbalanced Data Sets.* Proceedings of the international conference on Multimedia information retrieval *(pp. 111-118). Philadelphia, Pennsylvania, USA: ACM.*

[11] Tiwari, D. (2014). Handling Class Imbalanced Problem using Feature Selection, *International Journal of Advanced Research in Computer Science & Technology, Vol.2, Issue. 2, Ver.3* : pp. 516 – 520

[12] Alibeigi, M., Hashemi, S. & Hamzeh, A. (2012). DBFS : An effective Density Based Feature Selection scheme for small sample size and high dimensional imbalanced dataset, *Data & Knowledge Engineering 81 - 82* (pp. 67-103).

[13] Gao, K., Khoshgoftar, T. & Wald, R. 2014. Combining Feature Selection and Ensemble Learning for Software Quality Estimation. *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference.*

[14] Sun, Y., Mohamed, K. S., Wong, A. K., & Wang, Y. (2007). Cost-sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognition Society*, 3358-3378.

[15]Pandey, M., Taruna, S., 2014. A Comparative Study of Ensemble Method for Students Performance Model. *International Journal of Computer Application, Vol 103. No 8* : pp.26-31

[16] Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques.* Berlin: Springer-Verlag.

[17] Bramer, M. (2007). *Principles of Data Mining*. London: Springer.

[18] Moertini, V. S., 2007, "Pengembangan Skalabilitas Algoritma Klasifikasi C4.5 Dengan Pendekatan Konsep Operator Relasi, studi kasus: pra-pengolahan dan klasifikasi citra batik", Bandung .

[19] Witten, I. H, Frank, E and Hall, M. A., (2011), *Data Mining Practical Machine Learning Tools and Techniques*, 3rd ed. USA: Morgan Kaufmann Publishers.

[20] Ayub, M., Kristanti, T., & Caroline, M (2014). Model Analisis Classification dengan J48 untuk data mahasiswa dan dosen diperguruan tinggi. *SNASTIA.*

[21] Wang, S., & Yao, X. (2013). Using Class Imbalance Learning for Software Defect Prediction. *IEEE Transactions on Reliability*, 434-443

[22] Liu, X.-Y., & Zhou, Z.-H. (2013). *Ensemble Methods for Class Imbalance Learning. In H. He, & Y. Ma,* Imbalanced Learning: Foundations, Algorithms, and Applications *(pp. 61-82). NewJersey: John Wiley & Sons.*