# PENERAPAN METODE *ENSEMBLE* UNTUK MENINGKATKAN KINERJA ALGORITME KLASIFIKASI PADA *IMBALANCED DATASET*

**Yoga Pristyanto**

*Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta*
*Jl. Ringroad Utara, Condong Catur, Sleman, Yogyakarta*
*Email: yoga.pristyanto@amikom.ac.id*

**Abstrak**

*Pada bidang data mining sering kali para peneliti tidak memperhatikan keseimbangan distribusi kelas pada dataset. Hal ini dapat menimbulkan kesulitan yang cukup serius pada algoritme klasifikasi. karena secara teori mayoritas classifier mengasumsikan distribusi yang relatif seimbang, sehingga menyebabkan kinerja suatu algoritme klasifikasi menjadi kurang maksimal. Oleh karena itu, pada penelitian ini diterapkan metode ensemble dengan penambahan adaptive boosting untuk menyelesaikan permasalahan tersebut. Dari hasil pengujian yang dilakukan pada penelitian ini, metode ensemble dengan penambahan adaptive boosting dapat meningkatkan nilai kinerja algoritme klasifikasi. Nilai kinerja algoritme Naive Bayes dengan Adaptive Boosting akurasi yang dihasilkan sebesar 91.98%, sensitifitas sebesar 91.98%, spesifisitas sebesar 96.49%, dan g-mean sebesar 94.21%. Nilai kinerja algoritme Support Vector Machine dengan Adaptive Boosting akurasi yang dihasilkan sebesar 91.52%, sensitifitas sebesar 91.52%, spesifisitas sebesar 96.29%, dan g-mean sebesar 93.88%. Sedangkan Nilai kinerja algoritme Decision Tree dengan Adaptive Boosting akurasi yang dihasilkan sebesar 94.37%, sensitifitas sebesar 94.37%, spesifisitas sebesar 97.73%, dan g-mean sebesar 96.03%. Hal ini menunjukkan bahwa metode ensemble dengan Adaptive Boosting dapat menjadi solusi untuk meningkatkan kinerja algoritme pada imbalanced dataset.*

**Kata Kunci:** *adaptive boosting, data mining, ensemble, ketidakseimbangan kelas, klasifikasi.*

## 1. Pendahuluan

Data mining merupakan suatu penyelesaian masalah dengan melakukan analisis terhadap data yang disajikan dalam *database*. Selain itu data mining juga digunakan untuk mengetahui pola data, dimana setiap pola memiliki karakteristik masing-masing yang dapat memberikan informasi penting dari data tersebut [1]. Data mining dapat diartikan sebagai berbagai macam cabang ilmu pengetahuan yang menjadi satu, terdiri atas sistem basis data, statistika, *machine learning*, *visualization*, dan informasi pengetahuan. Data mining telah berhasil diterapkan di berbagai bidang ilmu seperti bisnis, bioinformatika, genetika, kedokteran, pendidikan dan lain sebagainya [2].

Beberapa teknik yang sering digunakan dalam data mining ialah klasterisasi, asosiasi, estimasi dan klasifikasi. Pada bidang pembelajaran mesin, teknik klasifikasi sering dimanfaatkan untuk berbagai hal antara lain untuk prediksi kinerja siswa, klasifikasi jenis penyakit, memprediksi kecurangan pada transaksi kartu kredit dan masih banyak hal lagi yang dapat dibantu dengan menggunkanan teknik klasifikasi [3]. Klasifikasi merupakan proses untuk menemukan sebuah model atau pola yang dapat menggambarkan serta membedakan kelas pada suatu *dataset*. Tujuannya agar model tersebut dapat digunakan untuk memprediksi obyek dengan label kelas yang tidak diketahui. Model tersebut didasarkan pada analisis data latih. Model dari hasil klasifikasi dapat dimanfaatkan untuk memprediksi tren data masa depan [4].

Ada beberapa algoritme klasifikasi yang sering digunakan dalam penelitian, terkait pembelajaran mesin yaitu *Decision Tree* (DT) [5], *Neural Network* (NN) [6], *K-Nearest Neighbor* (KNN), *Naive Bayes* (NB), dan *Support Vector Machine* (SVM) [7]. Akan tetapi mayoritas algoritme klasifikasi memiliki kelemahan dalam menangani klasifikasi dengan *dataset* yang memiliki ketidakseimbangan kelas [8]. Namun dari beberapa penelitian terkait data mining khususnya pada *machine learning* yang telah dilakukan sering kali para peneliti tidak memperhatikan keseimbangan distribusi kelas pada *dataset*. Ketidakseimbangan kelas merupakan suatu keadaan dimana terdapat perbedaan yang cukup signifikan antara jumlah *instance* kelas minoritas dengan jumlah *instance* kelas mayoritas. Ketidakseimbangan kelas menjadi salah satu masalah dalam domain dunia nyata (*real world problem*) yang sering muncul dalam bidang data mining [9]. Keberadaan distribusi kelas yang tidak seimbang dapat mempengaruhi performa dari suatu algoritme klasifikasi, karena suatu algoritme klasifikasi bekerja dengan mengasumsikan distribusi kelas pada *dataset* relatif seimbang dan biaya kesalahan klasifikasi yang sama [10]. Hal tersebut tentunya dapat menimbulkan resiko terjadinya kesalahan klasifikasi (*missclassification*) terhadap *dataset*, sehingga berakibat pada kinerja suatu algoritme klasifikasi menjadi tidak maksimal [11]. Oleh karena itu diperlukan penanganan lebih lanjut terkait adanya ketidakseimbangan kelas pada suatu *dataset*.

Berdasarkan penelitian yang telah dilakukan terkait penanganan terhadap ketidakseimbangan kelas pada *dataset*, terdapat dua pendekatan yang dapat diterapkan yaitu pendekatan pada level data dan pendekatan pada level algoritmik. Pendekatan level data biasanya dilakukan pada tahap pra-pemrosesan data dengan mengubah atau memperbaiki kecondongan distribusi kelas yang terdapat pada *dataset*. Metode yang sering

dipakai dalam pendekatan pada level data ialah menerapkan teknik *resampling* maupun sintesis data. Pada pendekatan level algoritmik cara kerjanya ialah menyesuaikan operasi algoritme yang ada untuk membuat suatu *classifier* lebih kondusif terhadap klasifikasi kelas minoritas atau dengan kata lain dilakukan modifikasi maupun penggabungan (*ensemble*) dari beberapa algoritme [11]. Pada pendekatan level data terdapat kekurangan yaitu beresiko terjadinya duplikasi data dan hilangnya informasi yang penting di dalam *dataset*, hal tersebut tentunya akan berpengaruh juga terhadap kinerja algoritme klasifikasi yang digunakan [9].

Oleh karena itu pada penelitian ini dilakukan penanganan ketidakseimbangan kelas pada *dataset* menggunakan metode *ensemble* untuk meminimalisir resiko terjadinya duplikasi data dan hilangnya informasi yang penting di dalam *dataset*. Berikut ini studi yang telah dilakukan terkait penanganan terhadap ketidakseimbangan kelas, pada beberapa studi tesebut menggunakan beberapa pendekatan sebagai solusinya. Seperti penelitian yang dilakukan oleh [12], [13] dan [14]. Mereka membuktikan bahwa penerapan teknik *resampling* atau pendekatan level data untuk menangani ketidakseimbangan kelas pada *dataset* dapat meningkatkan kinerja dari algoritme klasifikasi. Sedangkan [15] dan [8] membuktikan bahwa penerapan teknik *ensemble* menggunakan *boostraping* atau *bagging* dapat meningkatkan kinerja dari algoritme klasifikasi yang digunakan. Oleh karena itu pada penelitian ini dilakukan penanganan ketidakseimbangan kelas pada *dataset* menggunakan metode *ensemble* menggunakan *adaptive boosting*, hal ini dikarenakan belum banyak penelitian yeng menerapkan metode *ensemble* menggunakan *adaptive boosting* untuk menangani ketidakseimbangan kelas pada *dataset*.

Kontribusi yang dilakukan pada penelitian ini adalah:

1. Penerepan metode *ensemble* yang kami usulkan dapat menjadi solusi untuk menangani ketidakseimbangan kelas pada *dataset*.
2. Menunjukan bahwa penanganan terhadap ketidakseimbangan kelas pada *dataset* dapat meningkatkan kinerja algoritme klasifikasi.
3. Dapat menjadi referensi bagi penelitian selanjutnya terkait penanganan terhadap ketidakseimbangan kelas pada *dataset*.

## 2. Metode

### 2.1 Alat

Pada penelitian ini menggunakan perangkat keras dan perangkat lunak yang digunakan adalah sebagai berikut.

Perangkat Keras:

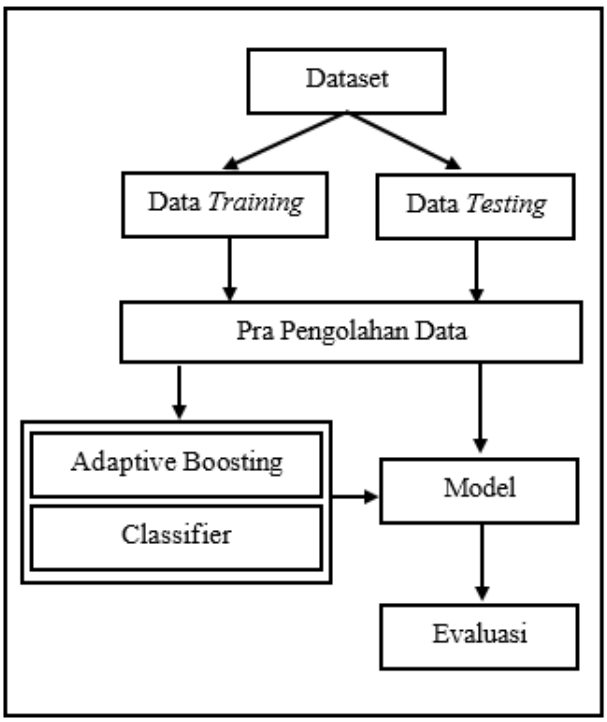
1) Processor: Intel Core i5 2410M 2.30 GHz
2) RAM: 6.00 GB

Perangkat Lunak :

1) R-Studio v.1.0.136
2) Weka versi 3.80

### 2.2 Alur Penelitian

Alur penelitian yang digunakan pada penelitian ini ditunjukkan pada Gambar 1.




**Gambar 1.** Diagram alur tahapan penelitian

Berikut ini adalah pemaparan dari langkah-langkah penelitian yang ditunjukkan pada Gambar 1.

#### 2.2.1. *Dataset*

*Dataset* yang digunakan pada penelitian ini adalah data *User Knowledge Modeling Dataset* (https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling). *Dataset* tersebut merupakan *dataset* publik yang diambil dari UCI Machine Learning. Data *User Knowledge Modeling Dataset* memiliki 403 *instance*, 5 *attributes* fitur dan 1 *attribute class*. Tabel 1 merupakan potongan data *User Knowledge Modeling Dataset* yang digunakan pada penelitian ini.

**Tabel 1.** Potongan *dataset*

| STG | SCG | STR | LPR | PEG | UNS (*Class*) |
|---|---|---|---|---|---|
| 0 | 0,1 | 0,5 | 0,26 | 0,05 | Very Low |
| 0,05 | 0,05 | 0,55 | 0,6 | 0,14 | Low |
| 0,08 | 0,18 | 0,63 | 0,6 | 0,85 | High |
| 0,2 | 0,2 | 0,68 | 0,67 | 0,85 | High |
| 0,22 | 0,22 | 0,9 | 0,3 | 0,9 | High |
| 0,14 | 0,14 | 0,7 | 0,5 | 0,3 | Low |
| 0,16 | 0,16 | 0,8 | 0,5 | 0,5 | Middle |

Keterangan:

- STG (*The degree of study time for goal object materails*).
- SCG (*The degree of repetition number of user for goal object materails*)

- STR (*The degree of study time of user for related objects with goal object*)
- LPR (*The exam performance of user for related objects with goal object*)
- PEG (*The exam performance of user for goal objects*)
- UNS (*The knowledge level of user*) merupakan *attribute class* pada *dataset* tersebut.

*Dataset User Knowledge Modeling* memiliki 5 *attributes* bertipe *numeric*, dan 1 *attribute* kelas dengan jumlah kelas sebanyak empat yaitu (*very low, low, middle,* dan *high*). Selain itu *dataset User Knowledge Modeling* memiliki distribusi kelas yang tidak seimbang. Gambar 2 merupakan ilustrasi distribusi kelas pada *dataset* tersebut.


**Gambar 2.** Ilustrasi Distribusi Kelas Pada *Dataset*

Pada Gambar 2 menunjukan adanya distribusi kelas yang tidak seimbang dimana kelas *Very Low* sebanyak 50 *instance*, *Low* sebanyak 129 *instance*, *Middle* sebanyak 122 *instance* dan *High* sebanyak 130 *instance*, secara teori hal ini dapat mempengaruhi kinerja dari suatu algoritme klasifikasi.

### 2.2.2. Pra Pengolahan Data

Data *preprocessing* merupakan tahapan dimana data akan dilakukan pengisian data yang kosong, menghilangkan duplikasi data, memeriksa inkonsistensi data, pembersihan data serta memperbaiki kesalahan pada data. Proses pembersihan meliputi pengisian data yang kosong, menghilangkan duplikasi data, memeriksa inkonsistensi data, dan memperbaiki kesalahan pada data. Biasanya data yang kosong disebabkan oleh adanya data baru yang belum ada informasinya [16].

Pada penelitian ini *dataset* yang digunakan tidak terdapat *missing value* atau data yang kosong. Sehingga bisa langsung dilanjutkan ke tahapan berikutnya.

### 2.2.3. Klasifikasi dan Model

Pada tahap ini algoritme klasifikasi yang akan digunakan ialah *Decision Tree* (DT), *Support Vector Machine*, dan *Naive Bayes*. Sedangkan metode *ensemble* yang digunakana ialah *adaptive boosting*.

**a) *Adaptive Boosting***

*Adaptive Boosting* (*adaboost*) merupakan salah satu dari beberapa varian pada algoritme *boosting* (Liu, 2015). *Adaboost* merupakan *ensemble learning* yang sering digunakan pada algoritme *boosting*. Algoritme *AdaBoost* dari Freund dan Schapire (1995) merupakan algoritme penguat praktis pertama, dan tetap menjadi salah satu yang paling banyak digunakan dan dipelajari, dengan aplikasi di berbagai bidang. *Boosting* bisa dikombinasikan dengan *classifier* algoritme yang lain untuk meningkatkan performa klasifikasi. Tentunya secara intuitif, penggabungan beberapa model akan membantu jika model tersebut berbeda satu sama lain. *Adaboost* dan variannya telah sukses diterapkan pada beberapa bidang (domain) karena dasar teorinya yang kuat, prediksi yang akurat, dan kesederhanaan yang besar. Langkah-langkah pada algoritme *adaboost* adalah sebagai berikut [17].

i. *Input*: Suatu kumpulan sampel penelitian dengan label $\{(X_i, Y_i),...,(X_N, X_N)\}$, suatu *component learn algoritme*, jumlah perputaran T.
ii. *Initialize*: Bobot suatu sampel pelatihan $W_N^1 = 1/N$,untuk semua i=1,...,N.
iii. Do for t = 1,...,T
   - Gunakan *component learn* algoritme untuk melatih suatu komponen klasifikasi, $h_t$, pada sample bobot pelatihan.
   - Hitung kesalahan pelatihannya pada $h_t$: $\varepsilon_t = \sum_{i=1}^{N} W_i^t, Y_i \neq h_t(X_i)$.
   - Tetapkan bobot untuk *component classifier* $h_t = \propto_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$.
   - *Update* bobot *sample* pelatihan $W_i^{t+1} = \frac{W_i^t \exp\{-\propto_t, Y_i, h_t(X_i)\}}{C_t}$, $i = 1,...,N$ $C_t$ adalah suatu konstanta normalisasi.
iv. *Output* : $f\ x\ = sign\ (\ \sum_{t-1}^{N} \propto_t\ h_t(x))$.

Secara teori *Boosting* berfungsi untuk mengoptimalkan kinerja algoritme klasifikasi agar kinerjanya bias maksimal. Pada kasus imbalanced dataset, penerapan boosting tidak akan merubah struktur pada dataset, yang artinya kondisi dataset tetap dalam bentuk imbalanced.

**b) Naive Bayes**

Naive Bayes merupakan sebuah pengklasifikasi probabilitas sederhana yang mengaplikasikan Teorema Bayes dengan asumsi tidak ada ketergantungan (*independent*) yang tinggi. Salah satu keuntungan algoritme Naive Bayes ialah dalam menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian hanya membutuhkan jumlah data pelatihan yang kecil. Karena diasumsikan sebagai variable independent, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians [2]. Berikut ini persamaan umum dari Naive Bayes :

$$p\ C\ F_1, \dots, F_1\ = \frac{p\ C\ \ p\ F_1,\dots,F_n\ C}{p\ F_1,\dots,F_n} \qquad (1)$$

Dimana variabel C merepresentasikan kelas, sementara variabel $F_1$...Fn merepresentasikan berbagai karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka persamaan tersebut menjelaskan bahwa peluang masuknya sampel dengan karakteristik tertentu dalam kelas C adalah peluang munculnya kelas C

sebelum masuknya sampel tersebut, dikali dengan peluang kemunculan berbagai karakteristik sampel pada kelas C dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (*evidence*).

**c) *Support Vector Machine***

*Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik. *Algoritme Support Vector Machine* (SVM) pertama kali diperkenalkan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar SVM merupakan kombinasi dari beberapa teori komputasi yang telah ada puluhan tahun sebelumnya. SVM akan bekerja sebagai berikut, menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi. Dalam dimensi baru ini, akan mencari *hyperplane* pemisah optimal linear. Dengan pemetaan nonlinear yang tepat untuk dimensi yang cukup tinggi, data dari dua kelas selalu dapat dipisahkan dengan *hyperplane*. Algoritme SVM menemukan hyperplane dengan menggunakan dukungan vektor dan *margin* [2].

Karakteristik SVM secara umum ialah sebagai berikut [18] :

i. Secara prinsip SVM adalah *linear classifier*.
ii. *Pattern recognition* dilakukan dengan mentransformasikan data pada *input space* ke ruang yang memiliki dimensi yang lebih tinggi. Hal ini membedakan SVM dari solusi *pattern recognition* pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi *input space*.
iii. Menerapkan strategi *Structural Risk Minimization* (SRM).
iv. Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi dua kelas

**d) Decision Tree**

Pada penelitian ini algoritme decision tree yang dipilih ialah C4.5, hal ini dikarenakan algoritme tersebut populer dan sering kali digunkana oleh para peneliti sebelumnya. Algoritme C4.5 merupakan pengembangan dari ID3 yang dikembangkan oleh J. R. Quintlant pada tahun 1987 [2]. Untuk membangun pohon keputusan dalam algoritme C4.5, hal pertama yang dilakukan yaitu memilih atribut sebagai akar, kemudian dibuat cabang untuk tiap-tiap nilai didalam akar tersebut. Langkah berikutnya yaitu membagi kasus dalam cabang. Kemudian ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama. Untuk memilih atribut dengan akar, didasarkan pada nilai *gain* tertinggi dari atribut-tribut yang ada. Gain (S,A) merupakan perolehan informasi dari atribut A *relative* terhadap *output* data S. Perolehan informasi didapat dari *output* data atau *variable* dependent S yang dikelompokkan berdasarkan atribut A, dinotasikan dengan gain (S,A). Berikut ini persamaan untuk menghitung nilai *gain* [19].

$$Gain\ \ S,A\ \ = Entropy\ \ S\ \ - \sum_{i=1}^{n} \frac{Si}{S} \times Entropy\ \ Si \qquad (2)$$

dimana,
*S* : Himpunan kasus
*A* : Atribut
*n* : Jumlah partisi atribut A
|*Si*| : Jumlah kasus pada partisi ke-i
|*S*| : Jumlah kasus dalam S

Sedangkan nilai *Entropy* dapat dihitung menggunakan persamaan berikut ini.

$$Entropy\ \ S\ \ = \sum_{i=1}^{n} - pi \times log_2 pi \qquad (3)$$

dimana,
*S* : Himpunan kasus
*n* : Jumlah partisi S
*pi* : Proporsi dari *Si* terhadap *S*

### 2.2.4. Evaluasi Kinerja

Evaluasi merupakan proses pengujian kinerja algoritme klasifikasi yang digunakan. Pada umumnya evaluasi kinerja algoritme klasifikasi. menggunakan *confusion matrix* [20]. Evaluasi dengan *confusion matrix* akan menghasilkan nilai *accuracy*, *sensitivity*, *specificity* dan *g-mean*. Akurasi dalam klasifikasi merupakan persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi [21]. *Specificity* proporsi kasus negatif yang diidentifikasi dengan benar. *Recall* atau *sensitivity* merupakan proporsi kasus positif yang diidentifikasi dengan benar [22]. Sedangkan *geometric mean* (*g-mean*) adalah salah satu pengukuran paling komprehensif untuk mengevaluasi kinerja algoritme klasifikasi khususnya dalam permasalahn ketidakseimbangan kelas pada *dataset*. *G-Mean* dapat menunjukkan akurasi keseluruhan dari akurasi kelas minoritas dan akurasi kelas mayoritas [9],[23]. Berikut ini persamaan untuk menghitung *akurasi*, *specificity*, *sensitivity*, dan *g-m*ean [2].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (4)$$

$$Sensitivity = \frac{TP}{TP+FN} \qquad (5)$$

$$Specificity = \frac{TN}{TN+FP} \qquad (6)$$

$$G - Mean = \sqrt{Sensitivity * Specificity} \qquad (7)$$

Keterangan:

- *True* Positive adalah jumlah *record* positif yang diklasifikasikan sebagai positif.
- *False* positif adalah jumlah *record* negatif yang diklasifikasikan sebagai positif.
- *False* negatif adalah jumlah *record* positif yang diklasifikasikan sebagai negatif.
- *True* negatif adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif.

## 3. Hasil dan Pembahasan

Tahap awal penelitian ini adalah pra pengolahan *dataset*, pada penelitian ini *dataset* yang digunakan tidak terdapat missing value atau data yang kosong maupun informasi yang rusak. Sehingga bisa langsung dilanjutkan ketahapan berikutnya yaitu tahap pengujian algoritme.

Pada proses pengujian algoritme *dataset* dibagi menjadi dua bagian yaitu sebagai data training dan data testing. Data training digunakan untuk melatih model sedangkan data testing digunakan untuk memvalidasi model. Kami menggunakan 80% untuk melatih model dan 20% digunakan untuk memvalidasi model, sehingga didapatkan hasil evaluasi algoritme sebagai berikut pada Tabel 2 dan Tabel 3.

**Tabel 2.** Evaluasi Algoritme Klasifikasi Tanpa *Adaptive Boosting* (%)

| Algoritme | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Naive Bayes | 89.91 | 89.91 | 95.64 | 92.73 |
| SVM | 81.39 | 81.39 | 91.33 | 86.21 |
| Decision Tree (C4.5) | 92.89 | 92.89 | 97.21 | 95.03 |

1=Akurasi; 2=Sensitivity; 3=Specificity; 4=G-Mean;

**Tabel 3.** Evaluasi Algoritme Klasifikasi Dengan *Adaptive Boosting* (%)

| Algoritme | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Adaptive Boosting + Naive Bayes | 91.98 | 91.98 | 96.49 | 94.21 |
| Adaptive Boosting + SVM | 91.52 | 91.52 | 96.29 | 93.88 |
| Adaptive Boosting + Decision Tree (C4.5) | 94.37 | 94.37 | 97.73 | 96.03 |

1=Akurasi; 2=Sensitivity; 3=Specificity; 4=G-Mean;

Berdasarkan Tabel 2 dan Tabel 3 menunjukkan nilai akurasi, sensitifitas, spesifisitas, serta g-mean yang dihasilkan oleh metode ensemble menggunakan adaptive boosting menghasilkan nilai evaluasi yang lebih baik dibandingkan metode tanpa menggunakan adaptive boosting. Hal ini menunjukkan bahwa penerapan metode ensemble dengan penambahan adaptive boosting dapat meningkatkan kinerja algoritme klasifikasi pada *imbalanced* dataset.

## 4. Kesimpulan

Ketidakseimbangan kelas pada *dataset* merupakan hal yang sering ditemui dalam klasifikasi pada data mining. Berdasarkan penelitian yang telah dilakukan, metode ensemble dengan penambahan adaptive boosting dapat menghasilkan nilai kinerja yang lebih baik dibandingkan metode tanpa penambahan adaptive boosting. Sehingga metode yang diusulkan dapat menjadi solusi dalam proses klasifikasi khususnya dengan kondisi data yang *imbalanced*. Untuk penelitian selanjutnya diharapkan penelitian dapat dilakukan khususnya untuk pengujian metode menggunakan metode *ensemble* lainnya seperti *bagging* maupun *stacking*.

## Daftar Pustaka


[1] Ian H. Wilten & Eibe Frank, *Data Mining Practical Machine Learning Tools and Techniques*, Second Edi. San Francisco: Morgan Kaufmann Publishers, 2005.

[2] Jiawei Han and Micheline Kamber, *Jiawei Han & Micheline Kamber*, Second Edi. San Francisco: Morgan Kaufmann Publishers, 2006.

[3] Y. Pristyanto, N. A. Setiawan, and I. Ardiyanto, "Hybrid Resampling to Handle Imbalanced Class on Classification of Student Performance in Classroom," in *The First International Conference on Informatics and Computational Sciences (ICICoS 2017)*, 2017, pp. 215–220.

[4] T. M. Christian and M. Ayub, "Exploration of classification using NBTree for predicting students' performance," in *Proceedings of 2014 International Conference on Data and Software Engineering*, 2014, pp. 1–5.

[5] G. Gray, C. McGuinness, and P. Owende, "An application of classification models to predict learner progression in tertiary education," *2014 4th IEEE Int. Adv. Comput. Conf. IACC 2014*, pp. 549–554, 2014.

[6] S. A. Kumar, M. N. Vijayalakshmi, and D. V. M. N. S.Anupama Kumar, "Appraising the Significance of Self Regulated Learning in Higher Education Using Neural Networks," *Int. J. Eng. Res. Dev.*, vol. Volume 1, no. Issue 1, pp. 9–15, 2012.

[7] M. Mayilvaganan and D. Kalpanadevi, "Comparison of classification techniques for predicting the performance of students academic environment," *Commun. Netw. Technol. (ICCNT), 2014 Int. Conf. Comput. Intell. Comput. Res.*, pp. 113–118, 2014.

[8] R. S. Wahono, N. S. Herman, and S. Ahmad, "Neural network parameter optimization based on genetic algorithm for software defect prediction," *Adv. Sci. Lett.*, vol. 20, no. 10–12, pp. 1951–1955, 2014.

[9] S. Aries and R. S. Wahono, "Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software," *J. Softw. Eng.*, vol. 1, no. 2, pp. 76–85, 2015.

[10] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit.*, vol. 40, no. 12, pp. 3358–3378, 2007.

[11] R. Longadge and S. Dongre, "Class Imbalance Problem in Data Mining Review," *Int. J. Comput. Sci. Netw.*, vol. 2, no. 1, pp. 83–87, 2013.

[12] G. Hu, T. Xi, F. Mohammed, and H. Miao, "Classification of wine quality with imbalanced data," *Proc. IEEE Int. Conf. Ind. Technol.*, pp. 1712–1717, 2016.

[13] S. T. Jishan, R. I. Rashu, N. Haque, and R. M. Rahman, "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique," *Decis. Anal.*, vol. 2, no. 1, pp. 1–25, 2015.

[14] R. I. Rashu, N. Haq, and R. M. Rahman, "Data mining approaches to predict final grade by overcoming class imbalance problem," *2014 17th Int. Conf. Comput. Inf. Technol. ICCIT 2014*, pp. 14–19, 2014.

[15] A. R. Naufal, R. Satria Wahono, and A. Syukur, "Penerapan Bootstrapping dan Weighted Information Gain untuk Optimasi Parameter pada Algoritma Support Vector Machine untuk Prediksi Loyalitas Pelanggan oleh :," *J. Intell. Syst.*, vol. 1, no. 2, pp. 98–108, 2015.

[16] O. N. Pratiwi, "Predicting student placement class using data mining," in *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2013*, 2013, no. August, pp. 618–621.

[17] H. Liu, H. Tian, Y. Li, and L. Zhang, "Comparison of four Adaboost algorithm based artificial neural networks in wind speed predictions," *Energy Convers. Manag.*, vol. 92, pp. 67–81, 2015.

[18] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine," *Proceeding Indones. Sci. Meeiting Cent. Japan*, 2003.

[19] Kusrini and E. Taufiq Luthfi, *Algoritma Data Mining*, Edisi Pert. Yogyakarta: Penerbit Andi, 2009.

[20] B. Max, *Principles of Data Mining*. London: Springer, 2007.

[21] M. Han, J., & Kamber, *Data Mining: Concepts and Techniques Second*, Second Edi., vol. 12. San Fransisco: Morgan Kauffman, 2006.

[22] D. M. W. Powers, "Evaluation: From Precision, Recall And F-Measure To ROC, Informedness, Markedness & Correlation," vol. 2, no. 1, pp. 37–63, 2011.

[23] Y. Pristyanto, I. Pratama, and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," in *2018 International Conference on Information and Communications Technology, ICOIACT 2018*, 2018, pp. 310–314.