

Meta-Algoritme *Adaptive Boosting* untuk Meningkatkan Kinerja Metode Klasifikasi pada Prestasi Belajar Mahasiswa

Yuni Yamasari^{1,2}, Supeno M. S. Nugroho³, Dwi F. Suyatno⁴, Mauridhi H. Purnomo⁵

Abstract— Determining the right class on student achievement is important in an evaluation process, because placing students in the right class helps lecturer in reflecting the successfulness of learning process. This problem relates to the performance of classification method which is measured by the classifier metrics. High performance is indicated by the optimality of these classifier's metrics. Besides, meta-algorithm adaptive boosting has been proven to be able to improve the performance of classifier in various fields. Therefore, this paper employs adaptive boosting to reduce the number of incorrect student placement in a class. The experimental results of implementing adaptive boosting in classifying student achievement shows that there is an increase of performance of all classification metrics, i.e., Kappa, Precision, Recall, F-Measure, ROC, and MAE. In terms of accuracy, J-48 is able to rise about 3.09%, which means this method reduces three misclassified students. Additionally, decisionStump increases 12.37% of accuracy. This also means this method is able to decrease 12 misclassified students. Finally, Simple Cart reaches the highest accuracy of about 23.71%, while the number of misclassified students is reduced to 24 students. However, there is no improvement in Random Forest method by using this adaptive boosting.

Intisari— Penetapan kelas yang tepat pada prestasi belajar mahasiswa adalah suatu hal yang penting dalam proses evaluasi. Hal ini disebabkan keberadaan mahasiswa pada kelas yang tepat dapat membantu dosen untuk merefleksikan berhasil tidaknya proses belajar mengajar yang telah dilakukan. Masalah ketepatan kelas ini berkaitan dengan kinerja metode klasifikasi yang diukur dengan ukuran-ukuran *classifier*. Kinerja yang tinggi diindikasikan dengan optimalnya ukuran-ukuran *classifier* tersebut. Di sisi lain, meta-algoritme *adaptive boosting* telah terbukti mampu meningkatkan kinerja *classifier* pada berbagai bidang. Oleh karena itu, makalah ini menerapkan *adaptive boosting* untuk meningkatkan kinerja *classifier* agar keberadaan mahasiswa pada kelas yang tidak sesuai bisa direduksi. Hasil uji coba penerapan *adaptive boosting* pada klasifikasi prestasi belajar mahasiswa ini menunjukkan peningkatan kinerja pada semua ukuran *classifier*, yaitu *Kappa*, *Precision*, *Recall*, *F-Measure*, *ROC* dan *MAE*. Berkaitan dengan akurasi, metode J-

48 mencapai kenaikan sekitar 3,09%, sehingga metode ini dapat mereduksi kesalahan kelas sebanyak tiga mahasiswa. Untuk *DecisionStump*, kenaikan akurasi adalah sekitar 12,37%, sehingga metode ini mampu menurunkan ketidaktepatan kelas sebanyak 12 mahasiswa. Untuk akurasi *Simple Cart*, metode ini meningkat sekitar 23,71%, sehingga metode ini paling optimal dalam pengurangan kesalahan kelas, yaitu sebanyak 24 mahasiswa. Sedangkan untuk metode *Random Forest* tidak ada peningkatan kinerja dengan penerapan *adaptive boosting* ini.

Kata Kunci— klasifikasi, prestasi belajar, kinerja, *adaptive boosting*.

I. PENDAHULUAN

Kemajuan teknologi informasi dan komunikasi membawa pengaruh positif terhadap dunia pendidikan. Hal ini ditunjukkan dengan banyaknya instansi pendidikan yang membangun perangkat lunak sebagai alat bantu dalam proses belajar mengajar dan proses evaluasi. Perangkat lunak untuk tujuan ini dapat dibangun dengan perangkat lunak *open source* maupun dengan pemrograman sendiri. Sistem *e-Learning* untuk SMKN Surabaya yang dibangun dengan pemrograman PHP dan MySQL merupakan salah satu contoh sistem yang dibangun sendiri [1]. Sedangkan sistem *e-learning* yang dibangun dengan menggunakan *moodle* dan *eFront* merupakan contoh proses belajar mengajar yang dibangun secara *open source* [2], [3]. Untuk sistem evaluasi, pembangunan sistem dengan *R package-exam* dan *UTME* merupakan beberapa contoh dari sistem evaluasi yang dibangun secara *open source* juga [4], [5]. Secara keseluruhan, pembangunan perangkat lunak ini membantu tercapainya keoptimalan dalam proses belajar mengajar dan proses evaluasi.

Penggunaan sistem *e-learning* dan sistem evaluasi ini akan menghasilkan tumpukan data pada tempat penyimpanan. Tumpukan data ini dapat digunakan untuk pengoptimalan pada proses-proses tersebut dengan penerapan *data mining*. Hal ini dapat dilakukan karena penerapan ini menghasilkan pengetahuan yang sangat berguna. Pengetahuan ini diperoleh dari beberapa aktivitas dari *data mining*, antara lain klasifikasi, klaster, dan analisis asosiasi. Beberapa penelitian tentang klasifikasi antara lain prediksi mahasiswa nonaktif dengan menggunakan beberapa metode klasifikasi [6], klasifikasi untuk memprediksi prestasi belajar siswa dengan menggunakan *Naïve Bayesian* [7], dan klasifikasi dengan *decision tree* untuk mengetahui hubungan kinerja akademik dan bidang minat [8]. Untuk penelitian pengklasteran, penelitian yang telah dilakukan antara lain pengklasteran untuk mengevaluasi kualitas pengajaran [9] dan penggunaan

¹ Mahasiswa S3, Program Studi Pasca Sarjana Teknik Elektro, Gedung B,C & AJ, Kampus ITS Sukolilo, Jalan Arif Rahman Hakim, Keputih, Surabaya, 60111, (telp: 031-5994251; email: yamasari16@mhs.ee.its.ac.id)

^{2,4} Dosen, Jurusan Teknik Informatika, Gedung A, Kampus Unesa Ketintang, Jalan Ketintang, Surabaya, 60231 (telp: 031-8299563 ;e-mail: yuniyamasari@unesa.ac.id, dwifatrianto@unesa.ac.id)

^{3,5} Dosen, Departemen Teknik Komputer & Program Studi Pasca Sarjana Teknik Elektro, Gedung B, C & AJ, Kampus ITS Sukolilo, Jalan Arif Rahman Hakim, Keputih, Surabaya, 60111, (telp: 031-5994251;e-mail:mardi@ee.its.ac.id, hery@ee.its.ac.id)

pengklasteran untuk mengevaluasi kinerja mahasiswa [10]. Sementara itu, penelitian yang berkaitan dengan analisis asosiasi adalah penerapan algoritme genetika untuk mengetahui pola-pola siswa dalam mengerjakan proses evaluasi dalam sistem *e-learning* [11].

Berkaitan dengan klasifikasi pada data siswa dari sistem evaluasi, pengetahuan yang dihasilkan adalah kelas-kelas siswa atau mahasiswa yang didasarkan pada kemampuan siswa atau mahasiswa. Kelas ini menunjukkan prestasi belajar siswa atau mahasiswa untuk mata pelajaran/mata kuliah tertentu. Pengetahuan ini sangat penting bagi guru atau dosen untuk pengoptimalan proses belajar mengajar karena pemetaan kemampuan anak didik dapat secara akurat dilihat. Selain itu, pengetahuan ini juga merefleksikan berhasil tidaknya pengajaran yang telah dilakukan. Namun, kinerja dari beberapa metode klasifikasi perlu ditingkatkan, karena akan berimbas pada level akurasi yang rendah dan akhirnya jumlah mahasiswa yang berada pada kelas yang tidak tepat akan semakin banyak.

Beberapa peneliti berusaha meningkatkan kinerja metode klasifikasi pada domain permasalahannya. Penyelesaian dicoba dilakukan dengan beberapa cara. Beberapa peneliti melakukan pemilihan fitur [12] dan pengestrakan fitur [13] untuk peningkatan kinerja klasifikasi ini. Selain itu, beberapa penelitian melakukan penerapan meta-algoritme untuk peningkatan kinerja ini. Salah satu meta-algoritme yang telah terbukti dapat meningkatkan kinerja adalah *adaptive boosting*. Beberapa penelitian yang menerapkan meta-algoritme ini antara lain adalah penelitian untuk perencanaan dan perubahan lingkungan [14], penelitian untuk pengembangan sistem lacak visual [15], peningkatan kinerja yang berkaitan dengan penelitian dalam bidang farmasi [16], peningkatan kinerja dalam bidang pemrosesan suara [17], peningkatan pemrosesan suara pada bidang khusus kesehatan [18], dan pengoptimalan pemrosesan gambar [19].

Oleh karena itu, dalam makalah ini diusulkan meta-algoritme *adaptive boosting* untuk meningkatkan kinerja klasifikasi prestasi belajar mahasiswa. Penerapan ini dilakukan untuk mengatasi ketidaktepatan kelas dari prestasi belajar mahasiswa. Ketepatan ini diindikasikan dengan kinerja metode klasifikasi yang meningkat. Semakin tinggi level akurasi dan ukuran-ukuran kinerja *classifier*, maka semakin sedikit mahasiswa berada pada kelas yang salah.

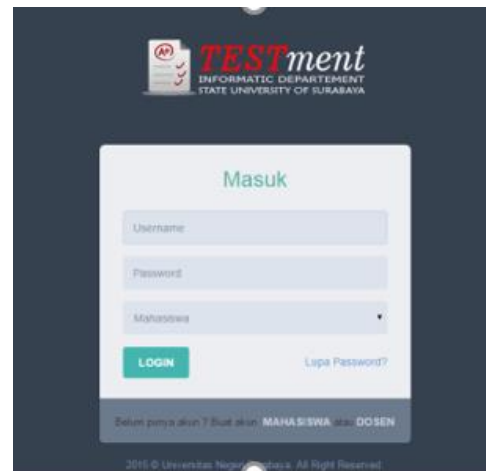
II. SISTEM EVALUASI

Pada bagian ini dibahas sistem TESTment yang digunakan sebagai alat untuk mengumpulkan data mahasiswa. Kemudian pembahasan dilanjutkan dengan proses pengumpulan data mahasiswa dengan sistem ini.

A. Sistem TESTment

Sistem ini merupakan sistem evaluasi yang dibangun sebagai alat untuk mengukur kemampuan mahasiswa dalam mata kuliah pemrograman dasar. Sistem ini telah divalidasi oleh ahli media sebelum digunakan untuk uji coba. Pada sistem ini, semua soal telah divalidasi oleh tim dosen mata kuliah pemrograman dasar. *Item* soal pada sistem ini terdiri

atas soal utama dan soal bimbingan. Rancangan soal ini ditujukan agar mahasiswa tidak langsung dinilai dari benar dan salahnya jawaban, tetapi juga dari usaha yang dilakukan. Selain itu, rancangan soal ini dimaksudkan agar semakin banyak perilaku mahasiswa yang tersimpan di dalam basis data.



Gbr. 1 Tampilan login sistem TESTment.

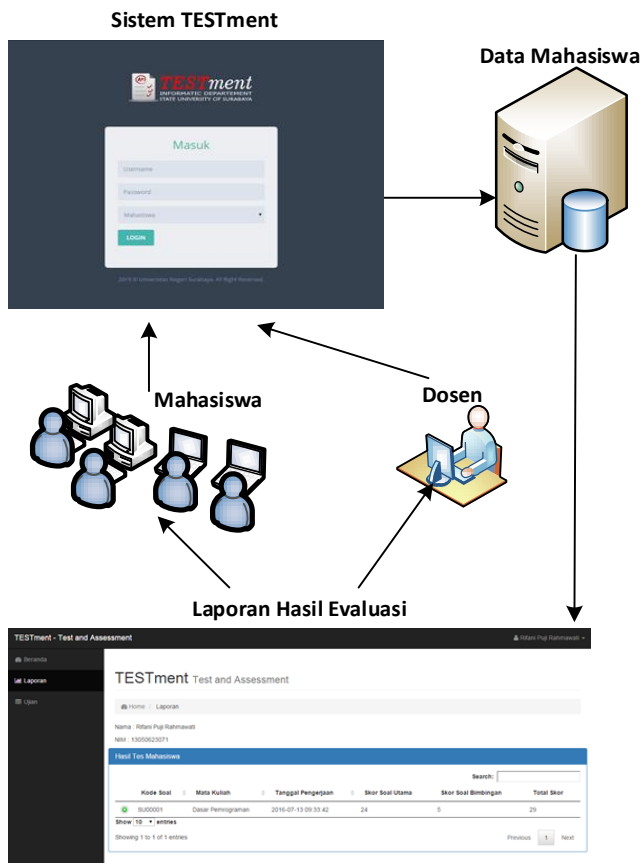
Mahasiswa harus melakukan *login* terlebih dahulu agar dapat mengakses sistem ini. Tampilan pertama sistem ini berupa *form login* seperti diperlihatkan pada Gbr. 1. Soal utama disusun berupa jawab singkat, sedangkan soal bimbingan dirancang dengan tipe soal pilihan ganda dan digunakan untuk membantu mahasiswa dalam memecahkan soal utama. Karena soal bimbingan ditujukan untuk memandu mahasiswa dalam memecahkan soal utama, maka pada setiap soal bimbingan tersedia petunjuk pengerjaan. Waktu pengerjaan untuk semua soal disimpan juga di basis data.

B. Pengumpulan Data

Makalah ini mempergunakan data yang proses pengumpulannya diperlihatkan pada Gbr. 2. Tampak bahwa data mahasiswa yang digunakan diperoleh ketika mahasiswa mengikuti evaluasi mata kuliah pemrograman dasar. Evaluasi ini dilakukan oleh pengguna yang terlebih dulu melakukan *login*.

Sebenarnya, sistem TESTment ini memiliki level pengguna, yaitu mahasiswa dan dosen. Namun, di sini hanya difokuskan pada pengumpulan data mahasiswa saja. Data mahasiswa ini merupakan perilaku mahasiswa yang tersimpan di basis data ketika mengakses sistem TESTment. Pada umumnya, data mahasiswa diolah untuk menghasilkan laporan hasil evaluasi yang dapat diakses oleh mahasiswa dan dosen. Pada makalah ini, diterapkan disiplin ilmu *data mining* untuk menggali data mahasiswa sehingga menghasilkan suatu pengetahuan.

Mahasiswa yang menjadi partisipan dalam penelitian ini berjumlah 98 mahasiswa. Partisipan mengikuti evaluasi mata kuliah pemrograman dasar dengan menggunakan sistem TESTment ini. Partisipan berasal dari tiga program studi dari jurusan Teknik Informatika Unesa angkatan 2015, yaitu satu kelas Sistem Informasi, satu kelas Teknik Informatika, dan satu kelas Pendidikan Teknologi.



Gbr. 2 Proses pengumpulan data.

Setiap mahasiswa mengerjakan sepuluh soal utama dan setiap soal utama mempunyai tiga soal bimbingan. Pada setiap soal bimbingan disediakan bantuan petunjuk agar mahasiswa dapat mengerjakan dengan benar. Ketika mahasiswa mengerjakan semua soal, waktu pengerjaannya juga disimpan di basis data.

Dengan demikian, data mahasiswa mengandung 153 fitur atau variable. Kemudian data diekstrak berdasarkan kategori sehingga fiturnya berubah menjadi sepuluh, seperti diperlihatkan pada Tabel I [20].

III. METODOLOGI

Bagian ini membahas tentang arsitektur yang diusulkan dan meta-algoritme *adaptive boosting*.

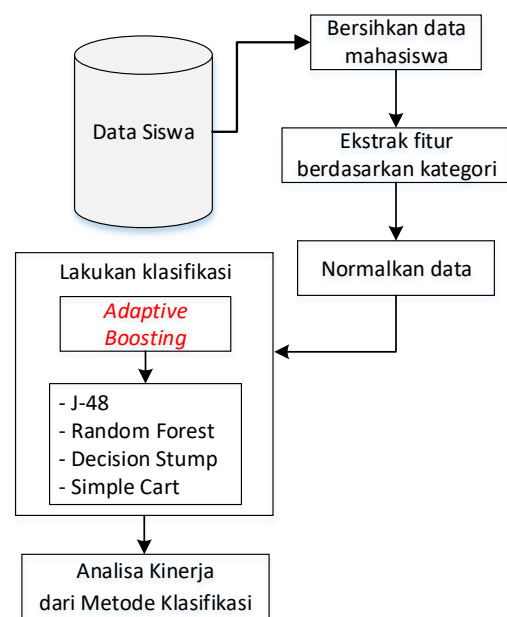
A. Arsitektur Sistem yang Diusulkan

Pada subbagian ini, dibahas tahapan-tahapan yang dilalui untuk melihat pengaruh *adaptive boosting* terhadap kinerja klasifikasi pada data mahasiswa. Tahapan-tahapan ini membentuk arsitektur sistem yang ditunjukkan pada Gbr. 3.

Arsitektur sistem ini terdiri atas enam tahapan, yaitu membersihkan data mahasiswa, mengekstraksi fitur berdasarkan kategori, menormalkan data, menerapkan meta-algoritme *adaptive boosting*, melakukan klasifikasi dengan menggunakan empat metode: *J-48*, *Random Forest*, *Decision Stump*, dan *Simple Cart*, serta menganalisis hasil klasifikasi yang merupakan tahapan terakhir dari usulan arsitektur sistem.

TABEL I
FITUR DAN TIPE DATA DARI DATA MAHASISWA

Fitur	Tipe data	Deskripsi
SU	Numerik	banyaknya soal utama yang dikerjakan
SU%	Numerik	prosentasi soal utama yang dikerjakan dengan benar
WSU	Numerik	banyaknya waktu yang dibutuhkan untuk mengerjakan soal utama
SSU	Numerik	skor soal utama
SC	Numerik	banyaknya soal cabang yang dikerjakan
SC%	Numerik	prosentasi soal cabang yang dikerjakan secara benar
WSC	Numerik	banyaknya waktu yang dibutuhkan untuk mengerjakan soal cabang
BP	Numerik	banyak petunjuk yang dibutuhkan
SSC	Numerik	skor soal cabang
ST	Numerik	skor total



Gbr. 3 Arsitektur sistem yang diusulkan.

Pada tahap pembersihan data mahasiswa, dilakukan pengecekan *data set*, mengandung data yang kosong, ganda (*redundancy*), dan seterusnya atau tidak. Pada tahap ini, data satu mahasiswa harus dihapus karena terdapat satu *field* yang nilainya rusak. Pada tahap kedua, data mahasiswa diekstrak fiturnya berdasarkan kategorinya. Tahap ini dilakukan dengan mengacu pada hasil penelitian yang telah dilakukan sebelumnya, sehingga fiturnya menjadi sepuluh, seperti diperlihatkan pada Tabel I [20]. Tahap selanjutnya adalah penormalan data yang dilakukan dengan tujuan agar *data set* mempunyai rata-rata 0 dan simpangan baku 1.

Untuk tahap keempat, diterapkan *adaptive boosting* dengan harapan kinerja *classifier* meningkat, karena meta-algoritme *adaptive boosting* ini dapat menanggulangi kesalahan (*error*) pada proses klasifikasi yang didasarkan pada penelitian lain yang telah dilakukan [21]. Pada tahap kelima dilakukan proses klasifikasi dengan empat metode klasifikasi, yaitu *J-48*,

Random Forest, Decision Stump, dan Simple Cart. Untuk tahap terakhir pada usulan arsitektur sistem ini, dilakukan analisis hasil dan pembahasan untuk melihat meta-algoritme *adaptive boosting* ini meningkatkan atau justru menurunkan kinerja klasifikasi prestasi belajar mahasiswa.

B. Meta-Algoritme Adaptive Boosting

Penelitian ini menggunakan langkah-langkah dari meta-algoritme *Adaptive Boosting*, yaitu sebagai berikut [21].

Terdapat $(x_1, y_1), \dots, (x_m, y_m)$; $x_i \in X, y_i \in \{-1, +1\}$.

Inisialisasi bobot $D_1(i) = 1/m$.

Untuk $t = 1, \dots, T$:

- Latih pembelajar lemah menggunakan distribusi D_t .
- Dapatkan hipotesis lemah $h_t: X \rightarrow \{-1, +1\}$ dengan error: $\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$.
- Pilih $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$.
- Perbarui: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$
 $= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$.

dengan Z_t adalah faktor normalisasi yang dipilih sehingga

D_{t+1} adalah sebuah distribusi.

Keluaran hipotesis akhirnya adalah

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right).$$

IV. HASIL PENGUJIAN DAN ANALISIS

Pada bagian ini dieksplorasi lebih jauh tentang implementasi dari usulan arsitektur sistem. Dalam pengujian, digunakan teknik evaluasi model *cross validation* untuk mengamati dan menganalisis hasil pengukuran kinerja. Sistem yang diusulkan diuji menggunakan *fold* 3-12. Apabila data mahasiswa terdiri atas 97 mahasiswa, maka data uji adalah 8-32 mahasiswa dan data latih adalah 65-89 mahasiswa. Ini berarti bahwa *classifier* dibangun dengan minimum mahasiswa sebanyak 65 mahasiswa dan maksimum mahasiswa sebanyak 89 mahasiswa.

A. Ukuran Kinerja Klasifikasi

Pengujian dilakukan pada empat metode klasifikasi. Pada masing-masing metode dilakukan perbandingan antara kinerja klasifikasi dengan penerapan meta-algoritme *adaptive boosting* dan tanpa penerapan meta-algoritme ini. Untuk mengukur kinerja ini, digunakan tujuh ukuran *classifier*, yaitu akurasi, *Kappa*, *MAE*, *Precision*, *Recall*, *F-Measure*, dan *ROC*.

Untuk metode klasifikasi J-48 dan *Adaptive Boosting* pada J-48 (AB_J-48), hasil pengujian disajikan pada Tabel II. Ukuran kinerja yang paling optimal adalah pada *fold* 4, 5, 7, 10, dan 11 untuk J-48 dan pada *fold* 10 dan 11 untuk AB_J-48. Pada *fold* ini, J-48 mampu mencapai nilai maksimal *Kappa* = 0,856, *Precision* = 0,908, *Recall* = 0,907, *F-Measure* = 0,905, *ROC* = 0,956, dan nilai minimal *MAE* = 0,086.

Sementara itu, metode AB_J-48 mampu mencapai nilai maksimal *Kappa* = 0,905, *Precision* = 0,94, *Recall* = 0,938, *F-Measure* = 0,938, *ROC* = 0,974, dan nilai minimal *MAE* = 0,047. Sebaliknya, nilai terendah dari kinerja terjadi pada *fold* 7, 8, dan 12 untuk J-48 dan *fold* 4, 8, dan 9 untuk AB_J-48.

TABEL II
UKURAN-UKURAN KINERJA PADA METODE J-48

J-48						
Fold	Kappa	MAE	Precision	Recall	F-measure	ROC
3	0,841	0,0932	0,896	0,897	0,896	0,94
4	0,8558	0,0859	0,908	0,907	0,905	0,96
5	0,8558	0,0859	0,908	0,907	0,905	0,96
6	0,824	0,0966	0,885	0,887	0,883	0,95
7	0,8558	0,0859	0,908	0,907	0,905	0,96
8	0,824	0,0966	0,885	0,887	0,883	0,95
9	0,8403	0,0949	0,896	0,897	0,895	0,94
10	0,8558	0,0859	0,908	0,907	0,905	0,96
11	0,8558	0,0859	0,908	0,907	0,905	0,96
12	0,824	0,0966	0,885	0,887	0,883	0,95
AB_J-48						
3	0,888	0,0548	0,932	0,928	0,927	0,99
4	0,8569	0,0664	0,908	0,907	0,907	0,97
5	0,8719	0,0544	0,92	0,918	0,916	0,97
6	0,8719	0,0544	0,92	0,918	0,916	0,97
7	0,888	0,0548	0,932	0,928	0,927	0,99
8	0,8569	0,0664	0,908	0,907	0,907	0,97
9	0,8569	0,0664	0,908	0,907	0,907	0,97
10	0,9045	0,047	0,94	0,938	0,938	0,97
11	0,9045	0,047	0,94	0,938	0,938	0,97
12	0,8719	0,0544	0,92	0,918	0,916	0,97

Adapun hasil klasifikasi dari metode ini diperlihatkan pada Tabel III. Hasil dari metode J-48 menunjukkan bahwa banyaknya mahasiswa pada kelas yang salah adalah 9 mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-5 dan ke-85 seharusnya berada di kelas 3, tetapi berada di kelas 1;
- mahasiswa ke-7, ke-17, ke-46, dan ke-93 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-39 seharusnya berada di kelas 1, tetapi berada di kelas 3;
- mahasiswa ke-90 seharusnya berada di kelas 2, tetapi berada di kelas 3; dan
- mahasiswa ke-96 seharusnya berada di kelas 1, tetapi berada di kelas 2.

Sementara itu, hasil klasifikasi dari metode AB_J-48 memperlihatkan bahwa mahasiswa pada kelas yang salah adalah sebanyak enam mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-7, ke-17, dan ke-93 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-39 seharusnya berada di kelas 1, tetapi berada di kelas 3;
- mahasiswa ke-90 seharusnya berada di kelas 2, tetapi berada di kelas 3; dan
- mahasiswa ke-96 seharusnya berada di kelas 1, tetapi berada di kelas 2.

TABEL III
HASIL KLASIFIKASI DARI METODE J-48

J-48		
Kelas	Mahasiswa ke-	Jumlah
1=tinggi	5,19,20,26,28,30,34,36,38,39,40,42,43,44,64,67,68,72,77,78,79,80,81,82,83,84,85,86,87,88,91,92,97	43
2=sedang	1,2,3,6,7,8,9,10,11,12,13,14,15,16,18,21,25,27,31,32,33,35,46,47,48,49,51,53,59,60,61,62,63,66,69,74,93,95,96	39
3=rendah	4,17,22,23,24,29,41,45,50,52,54,55,56,57,58,65,70,71,73,75,76,90,94	23
AB_J-48		
1=tinggi	5,19,20,26,28,30,34,36,38,39,40,42,43,44,64,67,68,72,77,78,79,80,81,82,83,84,85,86,87,88,91,92,97	43
2=sedang	1,2,3,6,7,8,9,10,11,12,13,14,15,16,18,21,25,27,31,32,33,35,46,47,48,49,51,53,59,60,61,62,63,66,69,74,93,95,96	39
3=rendah	4,17,22,23,24,29,41,45,50,52,54,55,56,57,58,65,70,71,73,75,76,90,94	23

Dari hasil klasifikasi pada metode J-48, dapat dilihat bahwa penerapan *adaptive boosting* ini dapat mereduksi jumlah mahasiswa yang salah kelas dari 9 mahasiswa menjadi 6 mahasiswa.

Berkaitan dengan metode klasifikasi *Random Forest* dan penerapan *Adaptive Boosting* pada *Random Forest* (*AB_Random Forest*), hasil pengujian disajikan pada Tabel IV. Ukuran kinerja yang paling optimal adalah pada *fold* 5, 6, dan 10 untuk *Random Forest* dan pada *fold* 3, 7, 8, 9, 10, 11, dan 12 untuk *AB_Random Forest*. Pada *fold* ini, *Random Forest* mampu memaksimalkan nilai *Kappa* = 0,888, *Precision* = 0,936, *Recall* = 0,928, *F-Measure* = 927, dan *ROC* = 0,981, serta meminimalkan nilai *MAE* = 0,248. Metode *AB_Random Forest* mencapai nilai maksimal yang sama dengan *Random Forest* untuk lima ukuran kinerja klasifikasi, yaitu *Kappa*, *Precision*, *Recall*, *F-Measure*, dan *MAE*. Sedangkan nilai *ROC*-nya mempunyai selisih = 0,008. Sebaliknya, nilai terendah kinerja terjadi pada *fold* 4 untuk *Random Forest* dan pada *fold* 4 dan 6 untuk *AB_Random Forest*.

Tabel V menyajikan data hasil klasifikasi. Hasil dari metode *Random Forest* dan *AB_Random Forest* menunjukkan banyaknya mahasiswa berada pada kelas yang salah adalah sama yaitu tujuh mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-5, ke-7, ke-17, dan ke-93 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-39 seharusnya berada di kelas 1, tetapi berada di kelas 3;
- mahasiswa ke-53 seharusnya berada di kelas 2, tetapi berada di kelas 3; dan
- mahasiswa ke-96 seharusnya berada di kelas 1, tetapi berada di kelas 2.

Dari hasil klasifikasi metode *Random Forest* ini, dapat dilihat bahwa penerapan *adaptive boosting* tidak meningkatkan kinerja, yang ditunjukkan dengan jumlah mahasiswa yang salah tidak berubah.

TABEL IV
UKURAN-UKURAN KINERJA PADA METODE *RANDOM FOREST*

Random Forest						
Fold	Kappa	MAE	Precision	Recall	F-Measure	ROC
3	0,8723	0,2562	0,921	0,918	0,918	0,983
4	0,8056	0,2594	0,886	0,876	0,87	0,975
5	0,8875	0,2483	0,936	0,928	0,927	0,981
6	0,8875	0,2483	0,936	0,928	0,927	0,981
7	0,8554	0,2521	0,911	0,907	0,905	0,973
8	0,8723	0,2562	0,921	0,918	0,918	0,983
9	0,8723	0,2562	0,921	0,918	0,918	0,983
10	0,8875	0,2483	0,936	0,928	0,927	0,981
11	0,8554	0,2521	0,911	0,907	0,905	0,973
12	0,8554	0,2521	0,911	0,907	0,905	0,973
AB_Random Forest						
3	0,8875	0,2498	0,936	0,928	0,927	0,989
4	0,8231	0,2568	0,891	0,887	0,884	0,971
5	0,8709	0,2549	0,931	0,918	0,916	0,983
6	0,8231	0,2568	0,891	0,887	0,884	0,971
7	0,8875	0,2498	0,936	0,928	0,927	0,989
8	0,8875	0,2498	0,936	0,928	0,927	0,989
9	0,8875	0,2498	0,936	0,928	0,927	0,989
10	0,8875	0,2498	0,936	0,928	0,927	0,989
11	0,8875	0,2498	0,936	0,928	0,927	0,989
12	0,8875	0,2498	0,936	0,928	0,927	0,989

TABEL V
HASIL KLASIFIKASI DARI METODE *RANDOM FOREST*

Random Forest dan AB_Random Forest		
Kelas	Mahasiswa ke-	Jumlah
1=tinggi	8,9,10,18,19,20,28,29,30,37,38,40,47,48,49,50,57,58,59,60,67,68,69,70,77,78,79,86,87,88,95,97	32
2=sedang	1,2,3,4,5,7,11,12,13,14,17,21,22,23,24,31,32,33,34,41,42,43,44,51,52,54,61,62,63,64,71,72,73,74,80,81,82,83,89,90,91,92,93,96	44
3=rendah	6,15,16,25,26,27,35,36,39,45,46,53,55,56,65,66,75,76,84,85,94	21

Untuk metode klasifikasi *Decision Stump* dan *Adaptive Boosting* pada *Decision Stump* (*AB_Ddecision Stump*), hasil pengujian ditunjukkan pada Tabel VI. Ukuran kinerja yang paling optimal adalah pada *fold* 3, 10, 11, dan 12 untuk *DecisionStump* dan pada *fold* 7 untuk *AB_DdecisionStump*. Pada *fold-fold* ini, *Decision Stump* mampu mencapai nilai maksimal *Kappa* = 0,577, *Precision* = 0,598, *Recall* = 0,742, *F-Measure* = 0,652, dan *ROC* = 0,812, serta mencapai nilai minimal *MAE* = 0,223. Sedangkan metode *AB_Ddecision Stump* mampu mencapai nilai maksimal *Kappa* = 0,788, *Precision* = 0,889, *Recall* = 0,866, *F-Measure* = 0,859, *ROC* = 0,937, dan nilai minimal *MAE* = 0,175. Sebaliknya, nilai terendah dari kinerja terjadi pada *fold* 4 dan 8 untuk *Decision Stump* dan pada *fold* 3 untuk *AB_Ddecision Stump*.

TABEL VI
UKURAN-UKURAN KINERJA PADA METODE *DECISION STUMP*

<i>Decision Stump</i>						
<i>Fold</i>	<i>Kappa</i>	<i>MAE</i>	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	<i>ROC</i>
3	0,5766	0,223	0,598	0,742	0,652	0,812
4	0,5587	0,231	0,6	0,732	0,646	0,809
5	0,5606	0,231	0,58	0,732	0,64	0,794
6	0,5601	0,23	0,585	0,732	0,642	0,796
7	0,5596	0,228	0,591	0,732	0,644	0,805
8	0,5587	0,231	0,6	0,732	0,646	0,809
9	0,5606	0,231	0,58	0,732	0,64	0,794
10	0,5766	0,223	0,598	0,742	0,652	0,812
11	0,5766	0,223	0,598	0,742	0,652	0,812
12	0,5766	0,223	0,598	0,742	0,652	0,812
<i>AB_Ddecision Stump</i>						
3	0,6662	0,171	0,857	0,794	0,753	0,92
4	0,7045	0,174	0,835	0,814	0,799	0,934
5	0,7079	0,181	0,816	0,814	0,803	0,893
6	0,7042	0,191	0,839	0,814	0,796	0,905
7	0,7882	0,175	0,889	0,866	0,859	0,937
8	0,7193	0,183	0,871	0,825	0,806	0,899
9	0,7079	0,179	0,816	0,814	0,803	0,909
10	0,7084	0,174	0,813	0,814	0,805	0,912
11	0,7549	0,164	0,863	0,845	0,832	0,93
12	0,7376	0,165	0,859	0,835	0,82	0,929

Hasil klasifikasi dari metode ini diperlihatkan pada Tabel VII dan menunjukkan bahwa banyaknya mahasiswa berada pada kelas yang salah adalah 25 mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-5, ke-6, ke-7, ke-15, ke-17, ke-25, ke-26, ke-27, ke-35, ke-36, ke-45, ke-46, ke-55, ke-56, ke-65, ke-66, ke-75, ke-76, ke-84, ke-85, dan ke-93 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-16 dan ke-94 seharusnya berada di kelas 3, tetapi berada di kelas 1; dan
- mahasiswa ke-39 dan ke-96 seharusnya berada di kelas 1, tetapi berada di kelas 2.

Sedangkan hasil klasifikasi dari metode *AB_Ddecision Stump* memperlihatkan bahwa banyaknya mahasiswa berada pada kelas yang salah adalah 13 mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-8, ke-23, ke-37, ke-65, ke-76, ke-78, ke-79, ke-92, dan ke-93 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-39 seharusnya berada di kelas 1, tetapi berada di kelas 3;
- mahasiswa ke-90 seharusnya berada di kelas 2, tetapi berada di kelas 3; dan
- mahasiswa ke-96 seharusnya berada di kelas 1, tetapi berada di kelas 2.

TABEL VII
HASIL KLASIFIKASI DARI METODE *DECISION STUMP*

<i>Decision Stump</i>		
<i>Kelas:</i>	<i>Mahasiswa ke-</i>	<i>Jumlah</i>
1=tinggi	8,9,10,16,18,19,20,28,29,30,37,38,40,47,48,49,50,57,58,59,60,67,68,69,70,77,78,79,86,87,88,94,97	33
2=sedang	1,2,3,4,5,6,7,11,12,13,14,15,17,21,22,23,24,25,26,27,31,32,33,34,35,36,39,41,42,43,44,45,46,51,52,53,54,55,56,61,62,63,64,65,66,71,72,73,74,75,76,80,81,82,83,84,85,89,90,91,92,93,96	64
3=rendah		0
<i>AB_Ddecision Stump</i>		
1=tinggi	3,10,11,12,13,14,24,25,26,27,28,38,39,41,42,51,52,53,54,55,56,66,67,68,69,70,80,81,82,83,84,94,95,97	34
2=sedang	1,2,4,5,6,8,15,16,17,18,19,20,23,29,30,31,32,33,34,37,40,43,44,45,46,47,48,57,58,59,60,61,62,65,71,72,73,74,75,76,78,79,85,86,87,88,89,92,93,96	50
3=rendah	7,9,21,22,35,36,49,50,63,64,77,90,91	13

Dari hasil klasifikasi pada metode *Decision Stump* ini, dapat dilihat bahwa penerapan *adaptive boosting* dapat mereduksi jumlah mahasiswa yang salah kelas dari 25 mahasiswa menjadi hanya 13 mahasiswa.

Untuk metode klasifikasi *Simple Cart* dan *Adaptive Boosting* pada *Simple Cart (AB_Simple Cart)*, hasil pengujian ditampilkan pada Tabel VIII. Ukuran kinerja yang paling optimal adalah pada *fold* 4 untuk *Simple Cart* dan pada *fold* 3 untuk *AB_Simple Cart*. Pada *fold* ini, *Simple Cart* mampu memaksimalkan nilai *Kappa* = 0,366, *Precision* = 0,599, *Recall* = 0,598, *F-Measure* = 0,578, *ROC* = 0,809, dan meminimalkan *MAE* = 0,269. Sementara metode *AB_Simple Cart* mampu mencapai nilai maksimal *Kappa* = 0,741, *Precision* = 0,848, *Recall* = 0,835, *F-Measure* = 0,831, *ROC* = 0,936, dan nilai minimal *MAE* = 0,119. Sebaliknya, nilai terendah dari kinerja terjadi pada *fold* 9 untuk *Simple Cart* dan *fold* 9 untuk *AB_Simple Cart*.

Hasil klasifikasi dari metode ini diperlihatkan pada Tabel IX. Hasil menunjukkan bahwa banyaknya mahasiswa berada pada kelas yang salah adalah 39 mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-1, ke-2, ke-4, ke-5, ke-6, ke-8, ke-9, ke-10, ke-75, ke-76, ke-78, dan ke-83 seharusnya berada di kelas 2, tetapi berada di kelas 1;
- mahasiswa ke-11, ke-12, ke-13, ke-14, ke-16, dan ke-65 seharusnya berada di kelas 3, tetapi berada di kelas 1;
- mahasiswa ke-43, ke-44, ke-45, ke-47, ke-48, dan ke-49 seharusnya berada di kelas 1, tetapi berada di kelas 2.
- mahasiswa ke-36, ke-37, ke-38, ke-39, ke-40, ke-41, ke-60, ke-61, ke-62, ke-63, dan ke-64 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-79 dan ke-80 seharusnya berada di kelas 2, tetapi berada di kelas 3; dan
- mahasiswa ke-46 dan ke-94 seharusnya berada di kelas 1, tetapi berada di kelas 3.

TABEL VIII
UKURAN-UKURAN KINERJA PADA METODE SIMPLE CART

Simple Cart						
Fold	Kappa	MAE	Precision	Recall	F-Measure	ROC
3	0,325	0,289	0,594	0,567	0,527	0,66
4	0,366	0,269	0,599	0,598	0,578	0,81
5	0,278	0,318	0,68	0,526	0,491	0,63
6	0,361	0,276	0,597	0,588	0,562	0,67
7	0,24	0,331	0,676	0,505	0,459	0,61
8	0,261	0,325	0,713	0,515	0,481	0,62
9	0,213	0,345	0,659	0,485	0,437	0,6
10	0,243	0,332	0,632	0,505	0,465	0,61
11	0,224	0,339	0,672	0,495	0,453	0,6
12	0,315	0,297	0,653	0,557	0,529	0,65
AB_Simple Cart						
3	0,741	0,119	0,848	0,835	0,831	0,94
4	0,74	0,128	0,845	0,835	0,829	0,93
5	0,684	0,129	0,861	0,804	0,777	0,96
6	0,693	0,139	0,806	0,804	0,796	0,92
7	0,694	0,133	0,809	0,804	0,801	0,93
8	0,723	0,132	0,842	0,825	0,815	0,9
9	0,63	0,167	0,76	0,763	0,756	0,91
10	0,741	0,115	0,836	0,835	0,828	0,91
11	0,716	0,135	0,819	0,814	0,817	0,93
12	0,708	0,129	0,825	0,814	0,81	0,94

Untuk hasil klasifikasi dari metode *AB_Simple Cart*, dapat dilihat bahwa banyaknya mahasiswa berada pada kelas yang salah adalah 15 mahasiswa, dengan perincian sebagai berikut:

- mahasiswa ke-5, ke-7, ke-17, ke-26, ke-45, ke-55, ke-56, ke-65, ke-84, dan ke-93 seharusnya berada di kelas 3, tetapi berada di kelas 2;
- mahasiswa ke-27 seharusnya berada di kelas 3, tetapi berada di kelas 1;
- mahasiswa ke-39 seharusnya berada di kelas 1, tetapi berada di kelas 3;
- mahasiswa ke-72 dan ke-73 seharusnya berada di kelas 2, tetapi berada di kelas 3; dan
- mahasiswa ke-96 seharusnya berada di kelas 1, tetapi berada di kelas 2.

Dari hasil klasifikasi pada metode *Simple Cart* ini, diketahui bahwa penerapan *adaptive boosting* ini dapat mereduksi jumlah mahasiswa yang salah kelas dari 39 mahasiswa menjadi hanya 15 mahasiswa.

Gbr. 4 menampilkan pengurangan jumlah mahasiswa yang berada di kelas yang salah dari semua metode.

Hampir semua metode yang menerapkan *adaptive boosting* mengindikasikan kemampuan dalam mereduksi banyaknya mahasiswa yang salah kelas. Kemampuan yang paling optimal ditunjukkan oleh metode *Simple Cart* sebanyak 24 mahasiswa, kemudian diikuti oleh metode *Decision Stump* sebanyak 12 mahasiswa, dan terakhir *J-48* sebanyak tiga mahasiswa.

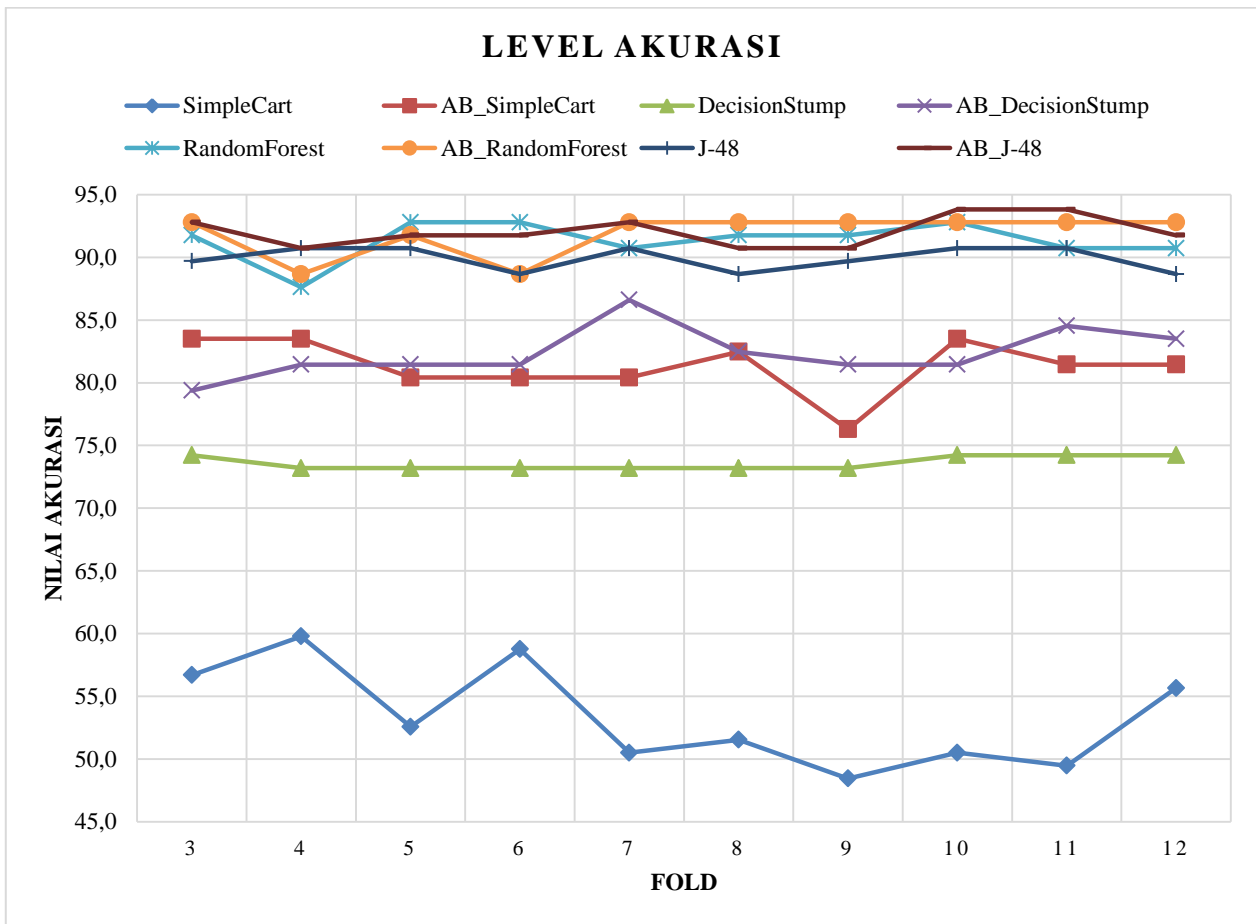
TABEL IX
HASIL KLASIFIKASI DARI METODE SIMPLE CART

Simple Cart		
Kelas:	Mahasiswa ke-	Jumlah
1=tinggi	1,2,4,5,6,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,42,65,66,67,68,69,70,71,72,73,75,76,78,83,89,90,91,92,93,95,96,97	44
2=sedang	3,7,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,74,77,81,82	43
3=rendah	15,46,79,80,84,85,86,87,88,94	10
AB_Simple Cart		
1=tinggi	8,9,10,18,19,20,27,28,29,30,37,38,40,47,48,49,50,57,58,59,60,67,68,69,70,77,78,79,86,87,88,95,97	33
2=sedang	1,2,3,4,5,7,11,12,13,14,17,21,22,23,24,26,31,32,33,34,41,42,43,44,45,51,52,53,54,55,56,61,63,64,65,71,74,80,81,82,83,84,89,90,91,92,93,96	48
3=rendah	6,15,16,25,35,36,39,46,62,66,72,73,75,76,85,94	16

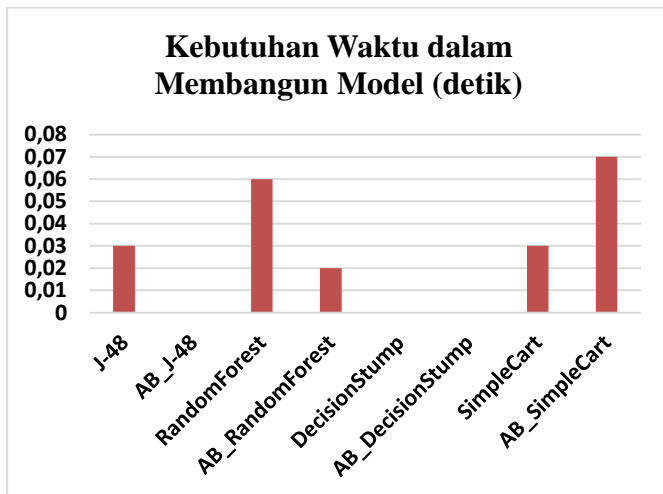


Gbr. 4 Jumlah mahasiswa yang berada pada kelas yang salah pada semua metode.

Sementara itu, level akurasi dari semua metode dideskripsikan pada Gbr. 5. Pada *J-48*, akurasi tertinggi adalah 90,7. Nilai ini dicapai oleh *fold* 4, 5, 7, 10, dan 11. Sebaliknya, akurasi terendah = 88,7 terjadi pada *fold* 6, 8, dan 12. Untuk *AB_J-48*, akurasi tertinggi = 93,8 dicapai oleh *fold* 10 dan 11. Sebaliknya, akurasi terendah terjadi pada *fold* 4, 8, dan 9, sedangkan akurasi tertinggi pada *Random Forest* adalah 92,78. Nilai ini dicapai oleh *fold* 5, 6, dan 10. Sebaliknya, akurasi terendah = 87,63. Nilai ini terjadi pada *fold* 4. Untuk *AB_Random Forest*, akurasi tertinggi yang dicapai sama dengan metode *Random Forest*, yaitu 92,78. Nilai ini dicapai oleh *fold* 3, 7, 8, 9, 10, 11, dan 12. Sebaliknya, akurasi terendah adalah 88,66 yang terjadi pada *fold* 4 dan 6.



Gbr. 5 Level akurasi dari semua metode klasifikasi.



Gbr. 6 Kebutuhan waktu dalam pembangunan model klasifikasi pada semua metode.

Untuk metode *Decision Stump*, akurasi tertinggi adalah 74,27. Nilai ini dicapai pada *fold* 3, 10, 11, dan 12. Sebaliknya, akurasi terendah = 73,19 terjadi pada *fold* 4, 5, 6, 7, 8, dan 9. Sedangkan akurasi tertinggi dari *AB_Decision Stump* adalah 86,59 yang dicapai oleh *fold* 7. Sebaliknya, akurasi terendah adalah 79,381 yang terjadi pada *fold* 3. Terakhir, akurasi dari *Simple Cart* tertinggi = 59,79 dicapai oleh *fold* 4. Sebaliknya,

akurasi terendah = 48,45 terjadi pada *fold* 9. Untuk *AB_Simple Cart*, akurasi tertinggi = 83,51 dicapai oleh *fold* 3, 4, dan 10. Sebaliknya, akurasi terendah terjadi pada *fold* 9 yang nilainya = 76,29.

B. Kebutuhan Waktu untuk Membangun Model

Pada makalah ini, selain enam ukuran kinerja yang sudah dibahas sebelumnya, dieksplorasi juga waktu yang dibutuhkan untuk membangun model *classifier*. Hasil uji coba kebutuhan waktu untuk pembangunan model ini diperlihatkan dengan grafik batang pada Gbr. 6.

Dapat dilihat bahwa metode *AB_J-48*, *Decision Stump*, dan *AB_Decision Stump* membutuhkan waktu paling singkat untuk membangun model, yaitu kurang dari 0,01 detik. Sedangkan kebutuhan waktu paling banyak terjadi pada pembangunan model dengan metode *AB_Simple Cart*, yaitu sebesar 0,07 detik.

Untuk kebutuhan waktu ini, jika dibandingkan antara metode asli dengan metode yang telah diterapkan meta-algoritme *Adaptive Boosting*, dapat dilihat bahwa penerapan meta-algoritme terhadap metode *J-48* dan *Random Forest* dapat mereduksi kebutuhan waktu ini. Namun, penerapan ini tidak membawa pengaruh terhadap kebutuhan waktu pada metode *Decision Stump*. Sebaliknya, untuk metode *Simple Cart*, kebutuhan waktu meningkat dengan penerapan meta-algoritme ini.

V. KESIMPULAN

Hampir semua metode klasifikasi dari prestasi belajar mahasiswa pada makalah ini menunjukkan peningkatan kinerja dengan penerapan meta-algoritme *adaptive boosting*. Peningkatan kinerja ini diindikasikan dengan meningkatnya level akurasi dan ukuran-ukuran *classifier* yang signifikan pada metode klasifikasi J-48, *Decision Stump*, dan *Simple Cart*. Semua metode ini mempunyai kemampuan untuk mereduksi banyaknya kesalahan kelas pada prestasi belajar mahasiswa sesuai dengan kinerja yang telah dicapainya. Sedangkan metode *Random Forest* tidak mengalami peningkatan kinerja dengan penerapan meta algoritme ini, yang berakibat pada ketidakmampuannya dalam mereduksi banyaknya mahasiswa yang salah kelas.

Untuk penelitian selanjutnya, meta-algoritme *adaptive boosting* ini akan dikembangkan agar tidak terlalu sensitif terhadap pencilan data (*data outlier*), agar kinerja dari meta-algoritme lebih optimal lagi.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada BUDI-DN/LPDP atas dukungan dana yang telah diberikan dalam penelitian ini.

REFERENSI

- [1] P. Wanarti, E. Ismayanti, H. Peni, and Y. Yamasari, "The Enhancement of Teaching-Learning Process Effectiveness through the Development of Instructional Media Based on E-learning of Surabaya 's Vocational Student," *Advances in Economics, Business and Management Research, International Conference on Educational, Management, Administration and Leadership (ICEMAL 2016)*, vol. 14, pp. 342–346, 2016.
- [2] T. Mchichi, P. Estrailier, and K. Afdel, "Web 2.0 Based E-Learning," *2011 International Conference on Multimedia Computing and Systems*, 2010.
- [3] R. Yunis and K. Telaumbanua, "Pengembangan E-Learning Berbasis LMS untuk Sekolah, Studi Kasus SMA/SMK di Sumatera Utara," *J. Nas. Tek. Elektro dan Teknol. Inf. (JNTETI)*, vol. 6, no. 1, pp. 32–36, 2017.
- [4] A. Zeileis, N. Umlauf, and F. Leisch, "Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond," *J. Stat. Softw.*, 2014.
- [5] K. F. Hew and W. S. Cheung, "Students' and instructors' use of massive open online courses (MOOCs): Motivations and challenges," *Educ. Res. Rev.*, vol. 12, pp. 45–58, 2014.
- [6] K. Hastuti, "Analisis komparasi algoritma klasifikasi data mining untuk prediksi mahasiswa non aktif," *Seminar Nasional Teknoogi Informasi & Komunikasi Terapan*, Semantik, pp. 241–249, 2012.
- [7] J. N. Purwaningsih and Y. Suwarno, "Predicting students achievement based on motivation in vocational school using data mining approach," *2016 4th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–5, 2016.
- [8] A. Ktona, D. Khaja, and I. Ninka, "Extracting Relationships Between Students' Academic Performance and Their Area of Interest Using Data Mining Techniques," *Sixth International Conference on Computational Intelligence, Communication System and Network*, pp. 6–11, 2014.
- [9] M. Imtiyaz, "Evaluating the Quality of Teaching in Higher Education Institutes using Clustering Approach and Set Pair Analysis," *1st International Conference on Next Generation Computing Technologies (NGCT-2015)*, no. September, pp. 4–5, 2015.
- [10] S. Rana and R. Garg, "Application of Hierarchical Clustering Algorithm to Evaluate Students Performance of an Institute," *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, pp. 692–697, 2016.
- [11] Y. Yamasari, D. Siahaan, and A. Z. Arifin, "Association Analysis using the Genetic Algorithms and Hybrid Measure for An Evaluation Process in the e-Learning System," *International Conference on Information and Communication Technology and Systems, ICTS 2008*, 2008.
- [12] J. Zhou, L. Chen, C. L. P. Chen, Y. Zhang, and H. X. Li, "Fuzzy clustering with the entropy of attribute weights," *Neurocomputing*, pp. 1–10, 2015.
- [13] C.-H. Li, B.-C. Kuo, and C.-T. Lin, "LDA-Based Clustering Algorithm and Its Application to an Unsupervised Feature Extraction," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 152–163, 2011.
- [14] R. Kang, T. Zhang, H. Tang, and W. Zhao, "Adaptive Region Boosting method with biased entropy for path planning in changing environment," *CAAI Trans. Intell. Technol.*, vol. 1, pp. 179–188, 2016.
- [15] Y. Chang and M. Hsu, "Development of a Visual Compressive Tracking System Enhanced by Adaptive Boosting," in *41st Annual Conference of the IEEE Industrial Electronics Society (IECON 2015)*, pp. 3678–3682, 2015.
- [16] R. A. Galvan, "Integrase Inhibition Using Differential Evolution-Binary Particle Swarm Optimization and Non-Linear Adaptive Boosting Random Forest Regression," *16th International Conference on Information Reuse and Integration*, pp. 485–490, 2015.
- [17] K. Chen, Y. Li, and X. Xu, "Rotating Target Classification base on Micro-doppler Features Using a Modified Adaptive Boosting Algorithm," *International Conference on Computers, Communication, and Systems*, pp. 236–240, 2015.
- [18] A. R. Hassan, "Biomedical Signal Processing and Control Computer-aided obstructive sleep apnea detection using normal inverse Gaussian parameters and adaptive boosting," *Biomed. Signal Process. Control*, vol. 29, pp. 22–30, 2016.
- [19] B. Wang, T. Lu, and Z. Xiong, "Adaptive Boosting for Image Denoising: Beyond Low-Rank Representation and Sparse Coding," *23rd International Conference on Pattern Recognition (ICPR)*, 2016.
- [20] Y. Yamasari, S. M. S. Nugroho, I. N. Sukajaya, and M. H. Purnomo, "Features extraction to improve performance of clustering process on student achievement," *2016 International Computer Science and Engineering Conference (ICSEC)*, pp. 1–5, 2016.
- [21] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *J. Japanese Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, 1999.