
Teknik *Bagging* Dan *Boosting* Pada Algoritma *CART* Untuk Klasifikasi Masa Studi Mahasiswa

Ahmad Rusadi Arrahimi¹⁾, Muhammad Khairi Ihsan²⁾, Dwi Kartini³⁾,
Mohammad Reza Faisal⁴⁾, Fatma Indriani⁵⁾

¹⁾²⁾³⁾⁴⁾⁵⁾Program Studi Ilmu Komputer FMIPA ULM
Jl. A. Yani Km 36, Banjarbaru, Kalimantan Selatan

¹⁾ahmadrusadi@ulm.ac.id

²⁾khairiihsan@gmail.com

³⁾dwikartini@ulm.ac.id

⁴⁾reza.faisal@gmail.com

⁵⁾f.indriani@ulm.ac.id

Abstrak

Data mahasiswa yang ada pada sistem informasi akademik selalu mengalami peningkatan setiap tahunnya. Data yang terkumpul dapat diolah menggunakan data mining untuk mendapatkan pengetahuan baru. Penulis berusaha menambang data mahasiswa untuk mengklasifikasi masa studi tepat waktu atau tidak tepat waktu. Data yang ada dianalisis menggunakan algoritma *CART* dengan teknik *bagging* dan algoritma *CART* dengan teknik *boosting*. Hasil klasifikasi dengan menggunakan data testing sejumlah 49 data, yaitu pada algoritma *CART* dengan teknik *bagging* 13 data (26,531%) masuk ke dalam klasifikasi tepat waktu dan 36 data (73,469%) masuk ke dalam klasifikasi tidak tepat waktu. Pada algoritma *CART* dengan teknik *boosting* 16 data (32,653%) masuk ke dalam klasifikasi tepat waktu dan 33 data (67,347%) masuk ke dalam klasifikasi tidak tepat waktu. Nilai accuracy klasifikasi masa studi mahasiswa menggunakan algoritma *CART* sebesar 79,592%, algoritma *CART* dengan teknik *bagging* sebesar 81,633%, dan algoritma *CART* dengan teknik *boosting* sebesar 87,755%. Dalam penelitian ini, algoritma *CART* dengan teknik *boosting* memiliki nilai accuracy yang paling baik.

Kata Kunci: Masa Studi, Klasifikasi, *CART*, *Bagging*, *Boosting*.

Abstract

Undergraduate Students data in academic information systems always increases every year. Data collected can be processed using data mining to gain new knowledge. The author tries to mine undergraduate students data to classify the study period on time or not on time. The data is analyzed using CART with bagging technique, and CART with boosting technique. The classification results using 49 testing data, in the CART algorithm with bagging techniques 13 data (26.531%) entered into the classification on time and 36 data (73.469%) entered into the classification not on time. In the CART algorithm with boosting technique 16 data (32,653%) entered into the classification on time and 33 data (67,347%) entered into the classification not on time. The accuracy value of the classification of study period of undergraduate students using the CART algorithm is 79.592%, the CART algorithm with bagging technique is 81.633%, and the CART algorithm with boosting technique is 87.755%. In this study, the CART algorithm with boosting technique has the best accuracy value.

Keywords: Study Period, Classification, *CART*, *Bagging*, *Boosting*

1. PENDAHULUAN

Data mahasiswa yang ada pada sistem informasi akademik setiap tahun mengalami peningkatan. Data yang terkumpul dapat dilakukan klasifikasi salah satunya dengan algoritma *CART*. Data yang didapatkan terdapat ketidakseimbangan data yang berarti salah satu kelas memiliki jumlah lebih sedikit dibandingkan dengan kelas lain (Chao, Liu, & Ding, 2013). Pendekatan yang dapat dilakukan untuk menangani data tidak seimbang adalah dengan memasang metode *ensemble* misalnya *bagging* dan *boosting* (Yap et al., 2014). *Bagging* dan *boosting* berhasil meningkatkan akurasi dari hasil klasifikasi data buatan maupun sebenarnya (Liang & Zhang, 2011). Hasil yang serupa juga didapat oleh Prasetio dan Pratiwi (2015) dan penelitian Bisri dan Wahono (2015) yang menerapkan metode *adaboost* sebagai metode *boosting*.

Berdasarkan uraian di atas, maka perlu adanya penelitian tentang teknik *bagging* dan *boosting* pada algoritma *CART* untuk klasifikasi masa studi mahasiswa berdasarkan nilai IP dan banyak SKS yang diambil semester 1 sampai semester 4.

2. TINJAUAN PUSTAKA

2.1. Literatur Terdahulu

Beberapa penelitian terdahulu yang berhubungan dengan penelitian yang dilakukan adalah jurnal oleh Nasution, Djahara dan Zamsuri (2015) yang menerapkan algoritma *Naïve Bayes* dalam penentuan kelulusan dengan melibatkan atribut jenis kelamin, asal sekolah, program studi, IP semester satu sampai lima, IPK, dan target kelulusan. Dari penelitian ini dihasilkan akurasi terbaik sebesar 76,001% kemudian setelah menggunakan atribut terpilih akurasi meningkat menjadi 76,67%.

Jurnal oleh Salmu dan Solichin (2017) menggunakan atribut jenis kelamin, pendapatan ayah, pendidikan ibu, IP semester satu sampai empat, jumlah SKS yang diambil pada semester satu sampai empat dengan akurasi 80,72% dengan menggunakan algoritma *Naïve Bayes*.

Jurnal oleh Yamasari et al (2017) yang bertujuan untuk meningkatkan kinerja klasifikasi prestasi belajar mahasiswa yang bermasalah pada ketidaktepatan kelas. Metode yang digunakan *Simple CART*, *Random Forest*, *Decision Stump*, dan *J-48*. Hampir semua metode klasifikasi yang digunakan menunjukkan peningkatan akurasi kecuali Metode *Random Forest*.

2.2. Classification and Regression Tree

Classification and Regression Tree (CART) adalah salah satu algoritma *decision tree*. *CART* dikembangkan untuk melakukan analisis klasifikasi pada variable respon baik yang nominal, ordinal, maupun kontinu. *CART* menghasilkan suatu pohon klasifikasi jika variable responnya kategorik dan menghasilkan pohon regresi jika variable responnya kontinu. Tujuan utama *CART* untuk mendapatkan suatu kelompok data yang akurat sebagai pencari dari suatu pengklasifikasian (Timofeev, 2004).

Menurut Lewis (2000), *CART* mempunyai beberapa kelebihan yaitu hasilnya lebih mudah diinterpretasikan, lebih akurat dan lebih cepat perhitungannya, selain itu *CART* bisa diterapkan untuk himpunan data yang mempunyai jumlah besar, variabel yang sangat banyak dengan skala variabel campuran melalui prosedur pemilahan biner.

Menurut Susanto dan Suryadi (2010) langkah-langkah algoritma *CART* adalah:

- Menyusun *candidate split* yang dilakukan terhadap seluruh variabel predictor.
- Memberikan penilaian keseluruhan calon cabang mutakhir dengan menghitung besaran $\Phi(s|t)$.
- Menentukan cabang yang memiliki kesesuaian $\Phi(s|t)$. Setelah noktah keputusan tidak ada lagi, *CART* dihentikan.

Kesesuaian $\Phi(s|t)$ dari calon cabang s pada noktah keputusan t , didefinisikan sebagai persamaan i-vi berikut :

$$\Phi(s|t) = 2P_L P_R Q(s|t) \quad (1)$$

$$Q(s|t) = \sum_{j=1}^{\text{jumlah kategori}} |P(j|t_L) - P(j|t_R)| \quad (2)$$

t_L = Cabang kiri dari noktah keputusan t
 t_R = cabang kanan dari noktah keputusan t

$$P_L = \frac{\text{calon cabang kiri } t_L}{\text{data latihan}} \quad (3)$$

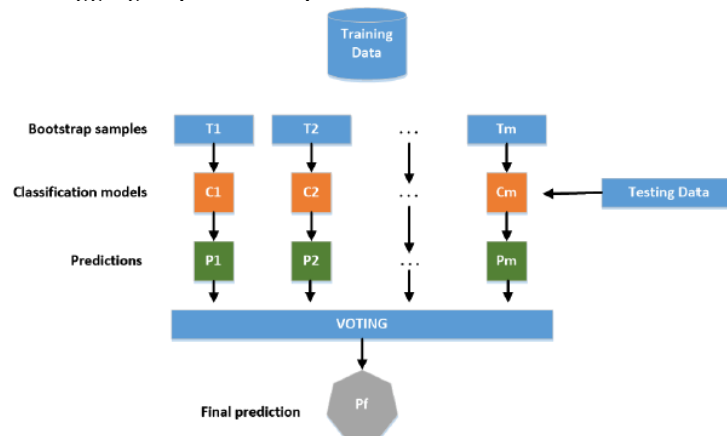
$$P(j|t_L) = \frac{j \text{ calon cabang kiri } t_L}{\text{noktah keputusan } t} \quad (4)$$

$$P_R = \frac{\text{calon cabang kanan } t_R}{\text{data latihan}} \quad (5)$$

$$(j|t_R) = \frac{j \text{ calon cabang kanan } t_R}{\text{noktah keputusan } t} \quad (6)$$

2.3. Bagging

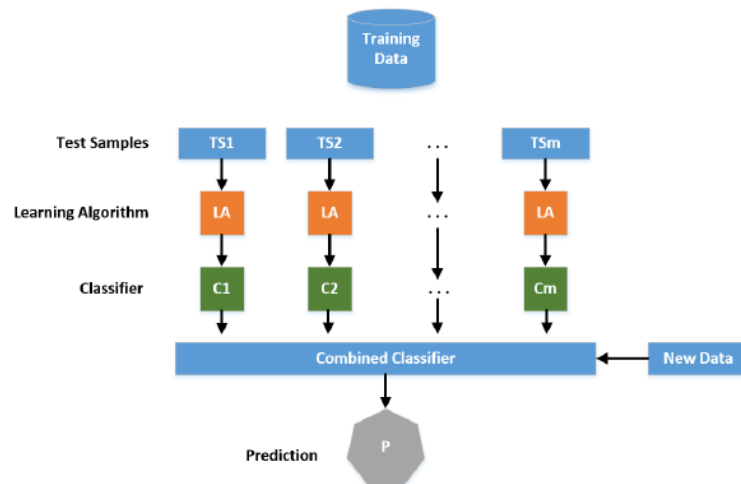
Bagging (bootstrap aggregating), menggunakan subdataset (*bootstrap*) untuk menghasilkan set pelatihan L (*learning*). L melatih dasar belajar menggunakan prosedur pembelajaran yang tidak stabil, dan kemudian, selama pengujian, mengambil rata-rata. *Noise* yang tinggi mempengaruhi klasifikasi baru yang dihasilkan, sehingga menyebabkan *miss-classification* (Prasetyo & Pratiwi, 2015). *Bagging* berguna untuk membangun pengklasifikasi menjadi lebih baik bila pada pengamatan kumpulan data latih yang terdapat *noise* (Alfaro, Gámez, & García, 2013). Skema teknik *bagging* dapat dilihat pada Gambar 1.



Gambar 1. Skema teknik *bagging* (Faisal, 2016)

2.4. Boosting

Algoritma *boosting* adalah algoritma iteratif yang memberikan bobot yang berbeda pada distribusi training data pada setiap iterasi. Setiap iterasi *boosting* menambahkan bobot pada contoh-contoh kesalahan klasifikasi dan menurunkan bobot pada contoh klasifikasi yang benar, sehingga secara efektif dapat merubah distribusi pada data *training* (S. Kotsiantis, Kanellopoulos, & Pintelas, 2006). Metode *Boosting (AdaBoost)* yang diusulkan (S. B. Kotsiantis & Pintelas, 2009) dengan *selective costing ensemble* dapat menjadi solusi yang lebih efektif untuk masalah *class imbalance* dan memungkinkan untuk meningkatkan identifikasi dari *class* minoritas yang sulit serta menjaga kemampuan klasifikasi dari *class* mayoritas. Pendekatan untuk mengatasi masalah tersebut dapat dilakukan dengan beberapa metode (Weiss, McCarthy, & Zabar, 2007) yaitu *over-sampling*, *under-sampling*, dan *cost-sensitive*. Adapun skema teknik *boosting* dapat dilihat pada Gambar 2.



Gambar 2. Skema teknik *boosting* (Faisal, 2016)

3. METODE PENELITIAN

3.1. Pencarian Literatur

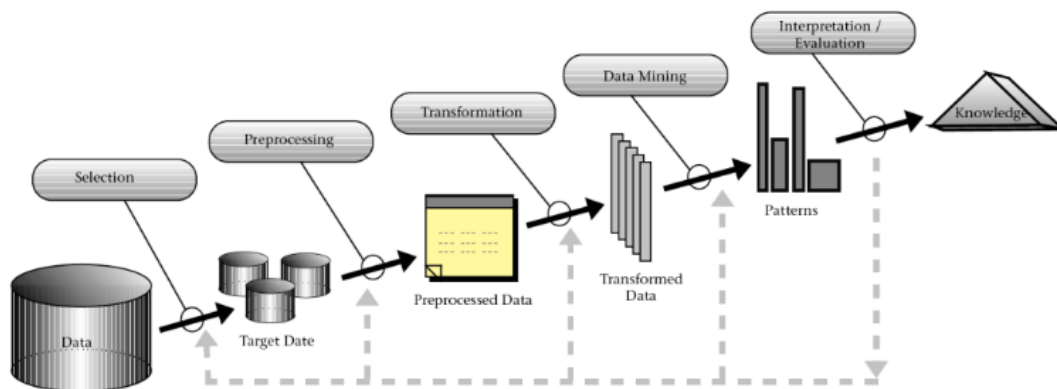
Literatur diperoleh dari buku-buku maupun referensi tentang masa studi mahasiswa, *data mining*, algoritma *CART*, teknik *bagging* dan *boosting*.

3.2. Pengumpulan Data

Data akademik mahasiswa dan data alumni mahasiswa Program Studi Ilmu Komputer FMIPA Universitas Lambung Mangkurat 2009-2014 yang diperoleh dari UPT PPTIK Universitas Lambung Mangkurat Banjarbaru.

3.3. Pengolahan Data

Data yang digunakan dalam penelitian ini diolah dengan menerapkan prinsip *Knowledge Discovery in Database* (KDD) yang meliputi tahapan sebagai berikut yang dapat dilihat pada Gambar 3.



Gambar 3. *Knowledge discovery in database*

4. PEMBAHASAN

4.1 Data Selection

Data yang digunakan berasal dari *database* Sistem Informasi Akademik di Program Studi Ilmu Komputer FMIPA ULM. Atribut yang digunakan yaitu, NIM, IP semester satu sampai empat, jumlah sks yang diambil pada semester satu sampai empat, dan lama studi.

4.2 Pre-processing/Cleaning

Data yang dihapus sebanyak empat data dikarenakan terdapat nilai kosong pada atribut. Data yang telah melalui tahap *cleaning* dapat dilihat pada Tabel 1.

Tabel 1. Data yang di-cleaning

No	NIM	SKS Semester				IP Semester				Masa Studi (Semester)
		1	2	3	4	1	2	3	4	
1	MAHASISWA 205	9	-	10	13	2,17	-	1,85	1,96	13
2	MAHASISWA 701	-	-	16	23	-	-	3,50	3,33	4
3	MAHASISWA 61	16	3	10	8	2,53	0,00	2,25	2,94	13
4	MAHASISWA 238	16	9	-	21	3,03	3,61	-	2,81	14

4.3 Transformation

Data diubah menjadi bentuk yang sesuai sehingga dapat dilakukan proses *data mining*. Pada Lama masa studi ditentukan bahwa untuk 8 dan 9 semester masuk dalam kelas “Tepat Waktu” yang ditulis kelas klasifikasi “A” dan untuk di atas 9 semester masuk dalam kelas “Tidak Tepat Waktu” yang ditulis kelas klasifikasi “B”.

Data yang telah ditransformasi dibagi menjadi dua bagian data, yaitu data *training* dan data *testing*. Data *training* berjumlah 146 data yang terdiri dari mahasiswa angkatan 2009-2011 dan data *testing* berjumlah 49 data yang terdiri dari mahasiswa angkatan 2012-2014.

4.4 Data Mining

4.4.1 Algoritma CART

Pertama-tama menyusun seluruh variabel yang menjadi calon cabang (*candidate split*) yang ada pada data *training*. Daftar calon cabang dapat dilihat pada Tabel 2.

Tabel 2. Daftar calon cabang

	Calon Cabang Kiri	Calon Cabang Kanan
IP Semester 1,	IPS1 ≤ 1,31	IPS1 > 1,31
...	... ≤ > ...
SKS Semester 4	SKS4 ≤ 23	SKS4 > 23

Kemudian menghitung nilai calon cabang untuk probabilitas setiap cabang P_L dan P_R . Untuk calon cabang kiri $IPS\ 1 \leq 1,31$ yang memenuhi ada 1 data dan calon cabang kanan $IPS\ 2 > 1,31$ yang memenuhi ada 145 data, maka dapat dihitung P_L dan P_R sebagai berikut.

$$P_L = \frac{1}{146} = 0,006849$$

$$P_R = \frac{145}{146} = 0,993151$$

Setelah mendapatkan nilai P_L dan P_R maka selanjutnya menghitung $P(j|t_L)$ dan $P(j|t_R)$ untuk kelas A dan kelas B. Jumlah $P(j|t_L)$ untuk semua kelas harus sama dengan jumlah data yang memenuhi calon cabang kiri P_L dan jumlah $P(j|t_R)$ untuk semua kelas juga harus sama dengan jumlah data yang memenuhi calon cabang kanan P_R . Pada $P(j|t_L)$ A tidak memiliki data dan $P(j|t_L)$ B memiliki 1 data. Pada $P(j|t_R)$ A memiliki 48 data dan $P(j|t_R)$ B memiliki 97 data.

$$P(j|t_L)\ A = \frac{0}{1} = 0$$

$$P(j|t_R)\ A = \frac{48}{145} = 0,33$$

$$P(j|t_L)\ B = \frac{1}{1} = 1$$

$$P(j|t_R)\ B = \frac{97}{145} = 0,67$$

Kemudian menghitung nilai $Q(s|t)$ dan kesesuaian $\Phi(s|t)$ sebagai berikut.

$$\begin{aligned} Q(s|t) &= |P(j|t_L)\ A - P(j|t_R)\ A| + |P(j|t_L)\ B - P(j|t_R)\ B| \\ &= |0 - 0,33| + |1 - 0,67| \\ &= 0,33 + 0,33 \\ &= 0,66 \end{aligned}$$

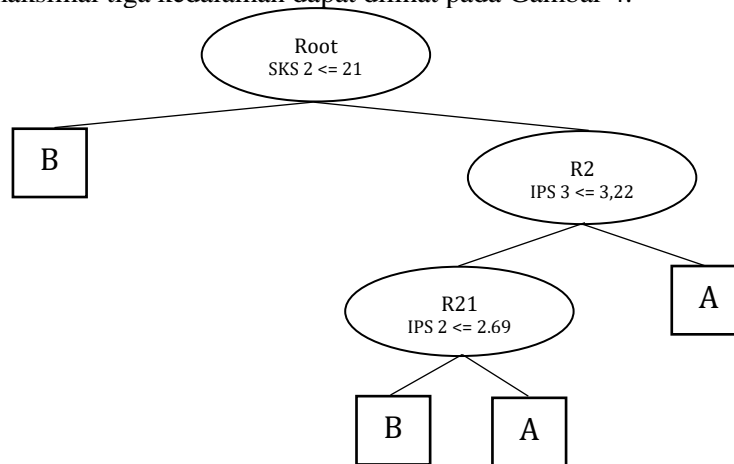
$$\begin{aligned}\Phi(s|t) &= 2 P_L P_R Q(s|t) \\ &= 2 \times 0,006849 \times 0,993151 \times 0,66 \\ &= 0,01\end{aligned}$$

Kemudian menghitung P_L , P_R , $P(j|t_L)$, $P(j|t_R)$, nilai $Q(s|t)$, dan kesesuaian $\Phi(s|t)$ dengan cara yang sama untuk semua calon cabang yang ada. Kesesuaian (*goodness*) $\Phi(s|t)$ yang paling tinggi dipilih menjadi cabang. Adapun hasil perhitungan dapat dilihat pada Tabel 3.

Tabel 3. Tabel perhitungan P_L , P_R , $P(j|t_L)$, $P(j|t_R)$, $Q(s|t)$, dan $\Phi(s|t)$

	IP Semester 1						...	SKS Semester 4				
	1,31	1,69	1,71	1,79	1,83	1,86		19	20	21	22	23
P_L	0,01	0,01	0,02	0,03	0,03	0,04	...	0,23	0,27	0,47	0,63	0,68
P_R	0,99	0,99	0,98	0,97	0,97	0,96	...	0,77	0,73	0,53	0,37	0,32
$P(j t_L)$ -A	0	0	0,33	0,25	0,2	0,17	...	0,03	0,05	0,13	0,17	0,22
$P(j t_L)$ -B	1	1	0,67	0,75	0,8	0,83	...	0,97	0,95	0,87	0,83	0,78
$P(j t_R)$ -A	0,33	0,33	0,33	0,33	0,33	0,34	...	0,42	0,43	0,51	0,59	0,57
$P(j t_R)$ -B	0,67	0,67	0,67	0,67	0,67	0,66	...	0,58	0,57	0,49	0,41	0,43
$Q(s t)$	0,66	0,67	0,01	0,16	0,27	0,34	...	0,78	0,77	0,75	0,84	0,69
$\Phi(s t)$	0,01	0,02	0	0,01	0,02	0,03	...	0,08	0,09	0,1	...	0,3

Hasil perhitungan kesesuaian $\Phi(s|t)$ menunjukkan bahwa calon cabang tertinggi nilai besarnya adalah 0,42, yaitu cabang kiri SKS Semester 2 ≤ 21 dan cabang kanan SKS Semester 2 > 21 , maka calon cabang ini dipilih menjadi *root node*. Kemudian cabang yang lainnya akan terus dihitung dengan cara yang sama menggunakan iterasi selanjutnya sampai dengan maksimal tiga kedalaman. Berikut pohon keputusan yang dihasilkan setelah dilakukan iterasi dengan maksimal tiga kedalaman dapat dilihat pada Gambar 4.



Gambar 4. Pohon keputusan algoritma *CART*

Setelah selesai melakukan perhitungan algoritma *CART*, maka dilakukan percobaan untuk mengetahui hasil klasifikasi pada data *testing*. Hasil klasifikasi dapat dilihat pada Tabel 4.

Tabel 4. Hasil klasifikasi untuk data *testing* menggunakan algoritma *CART*

No	NIM	IP Semester				SKS Semester				Kelas Aktual	Kelas CART
		1	2	3	4	1	2	3	4		
1	MAHASISWA 3	3,16	3,17	2,92	3,12	16	12	18	21	B	B
2	MAHASISWA 4	3,16	2,57	2,75	3,24	19	21	20	21	B	B
3	MAHASISWA 6	3,07	2,62	2,66	3,25	22	21	16	16	B	B
...
49	MAHASISWA 214	3,59	3,69	3,76	3,83	22	24	23	24	A	A

4.4.2 Algoritma CART dengan Teknik Bagging

Data *training* dibagi menjadi 10 *bagging* yang setiap data pada *bagging* diambil secara acak dari data *training* awal dan jumlahnya sama dengan jumlah data pada data *training* awal. Setiap *bagging* dilakukan perhitungan algoritma *CART* yang kemudian menghasilkan satu pohon keputusan pada setiap *bagging*. Pada proses *data mining* menggunakan algoritma *CART* dengan teknik *bagging* ini menghasilkan sebanyak 10 pohon keputusan yang berbeda-beda.

Setelah mendapatkan 10 pohon keputusan, langkah selanjutnya adalah mengklasifikasikan data *testing* ke dalam setiap *bagging* yang berisi algoritma *CART*. Penentuan klasifikasi akhir dengan cara *voting* menggunakan klasifikasi kelas yang paling banyak muncul pada data *testing* dalam 10 *bagging* tersebut. Hasil klasifikasi dapat dilihat pada Tabel 5.

Tabel 5. Hasil klasifikasi data *testing* menggunakan algoritma *CART* dengan teknik *bagging*

No	NIM	Bagging										Kelas Prediksi Bagging
		1	2	3	4	5	6	7	8	9	10	
1	MAHASISWA 3	B	B	A	B	B	A	B	B	B	B	B
2	MAHASISWA 4	B	B	A	B	B	A	B	A	B	A	B
3	MAHASISWA 6	B	B	B	B	B	B	B	A	B	B	B
...
49	MAHASISWA 214	A	A	A	A	A	B	A	A	A	A	A

4.4.3 Algoritma CART dengan Teknik Boosting

Pertama-tama data *training* diberi bobot dengan bobot awal

$$D_1(i) = \frac{1}{146}$$

$$D_1(i) = 0,006849$$

Kemudian dilakukan pengambilan data secara acak menjadi *boosting* 1 yang diambil dari data *training*. Data *boosting* 1 dapat dilihat pada Tabel 6.

Tabel 6. Data pada *boosting* 1

No	NIM	IP Semester				SKS Semester				Kelas Klasifikasi
		1	2	3	4	1	2	3	4	
73	MAHASISWA 211	3,48	2,90	3,53	3,58	21	24	20	24	B
98	MAHASISWA 23	2,57	3,17	3,02	3,33	21	21	23	24	A
38	MAHASISWA 219	2,43	2,55	2,64	3,05	21	19	22	21	B
8	MAHASISWA 11	2,71	1,50	2,94	2,45	21	21	16	20	B
25	MAHASISWA 201	2,86	2,93	3,65	3,31	21	21	24	24	B
...
130	MAHASISWA 233	2,36	1,78	3,03	2,75	21	18	15	24	B

Selanjutnya dilakukan perhitungan algoritma *CART* pada *boosting* 1 dan didapatkan pohon keputusan *boosting* 1. Kemudian dilakukan pengujian data *training* awal terhadap pohon keputusan yang dihasilkan algoritma *CART* pada *boosting* 1. Hasil klasifikasi data *training* pada *boosting* 1 dapat dilihat pada Tabel 7.

Tabel 7. Hasil klasifikasi data *training* dengan menggunakan algoritma *CART* pada *boosting* 1

No	NIM	IP Semester				SKS Semester				Kelas Aktual	Prediksi Kelas
		1	2	3	4	1	2	3	4		
1	MAHASISWA 1	2,79	2,98	3,48	3,00	21	21	21	24	B	B
2	MAHASISWA 2	2,79	2,55	3,71	3,23	21	21	19	24	B	B
3	MAHASISWA 3	2,36	1,17	1,65	2,91	21	18	13	16	B	B
...
146	MAHASISWA 11	3,30	2,78	3,25	3,56	22	18	14	17	B	B

Setelah didapatkan hasil pengujian data *training*. Kemudian menghitung nilai error (ϵ_t) dari hasil prediksi kelas yang tidak sama dengan kelas aktual. Hasil pengujian data *training* menggunakan hasil algoritma *CART* pada *boosting* 1 terdapat 42 data yang berbeda.

$$\epsilon_t = \frac{42}{146} = 0,287671$$

Kemudian menemukan nilai *Alpha* (α_t) pada *boosting* 1

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - 0,287671}{0,287671} \right) = 0,453361$$

Setelah mendapatkan nilai *Alpha* (α_t) pada *boosting* 1, kemudian memperbarui bobot dari masing-masing data pada data *training* dengan persamaan berikut.

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \end{cases}$$

Pembaruan bobot data *training* dapat dilihat pada Tabel 8.

Tabel 8. Pembaruan bobot setiap data pada data *training* setelah *boosting* 1

No	NIM	IP Semester				SKS Semester				$D_t(i)$	$D_{t+1}(i)$
		1	2	3	4	1	2	3	4		
1	MAHASISWA 1	2,79	2,98	3,48	3,00	21	21	21	24	0,006849	0,004808
2	MAHASISWA 2	2,79	2,55	3,71	3,23	21	21	19	24	0,006849	0,004808
3	MAHASISWA 3	2,36	1,17	1,65	2,91	21	18	13	16	0,006849	0,004808
...
146	MAHASISWA 11	3,30	2,78	3,25	3,56	22	18	14	17	0,006849	0,004808

Kemudian dilakukan pengambilan data secara acak untuk menjadi *boosting* 2 yang diambil dari data *training* namun data yang memiliki bobot lebih besar memiliki kemungkinan terpilih lebih besar pula.

Selanjutnya mengulangi proses yang sama seperti *boosting* 1 sebanyak 10 *boosting* sehingga didapatkan 10 pohon keputusan yang dihasilkan dari algoritma *CART* dan 10 nilai *alpha* yang dihasilkan dari masing-masing pengujian data *training* pada setiap *boosting*. Setelah didapatkan 10 *boosting*, maka akan dilakukan pengujian dengan menggunakan data *testing* terhadap setiap algoritma *CART* pada masing-masing *boosting* yang telah ditemukan. Hasil pengujian data *testing* pada algoritma *CART* dengan teknik *boosting* dapat dilihat pada Tabel 9.

Tabel 9. Hasil pengujian data *testing* pada algoritma *CART* dengan teknik *boosting*

No	NIM	Dataset Boosting									
		1	2	3	4	5	6	7	8	9	10
1	MAHASISWA 3	B	B	B	A	B	B	B	B	B	B
2	MAHASISWA 4	B	B	B	A	B	B	B	B	B	B
3	MAHASISWA 6	B	B	B	A	B	B	B	B	B	B
...
49	MAHASISWA 214	A	A	A	A	A	A	A	B	B	A

Klasifikasi akhir pada teknik *boosting* menggunakan persamaan berikut.

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

Klasifikasi yang menghasilkan A bernilai positif dan B bernilai negatif. Kemudian dikalikan dengan nilai *alpha* dari masing-masing *boosting*. Kemudian menjumlahkan semua hasil perkalian tersebut. Jika hasil akhir $H(x)$ bernilai positif maka klasifikasi akhir bernilai A, jika hasil akhir $H(x)$ bernilai negatif maka klasifikasi akhir bernilai B. Klasifikasi akhir pada *Boosting* dapat dilihat pada Tabel 10.

Tabel 12. Klasifikasi akhir algoritma CART dengan teknik *boosting*

No	NIM	$\alpha_t h_t(x)$										H(x)	Kelas Prediksi Boosting
		1	2	3	4	5	6	7	8	9	10		
1	MAHASIS WA3	-0,4534	-0,6762	-0,3883	0,5585	-0,5771	-0,6762	-0,6555	-0,4042	-0,5771	-0,4534	-4,30278	B
2	MAHASIS WA4	-0,4534	-0,6762	-0,3883	0,5585	-0,5771	-0,6762	-0,6555	-0,4042	-0,5771	-0,4534	-4,30278	B
3	MAHASIS WA6	-0,4534	-0,6762	-0,3883	0,5585	-0,5771	-0,6762	-0,6555	-0,4042	-0,5771	-0,4534	-4,30278	B
4	MAHASIS WA8	-0,4534	-0,6762	-0,3883	-0,5585	-0,5771	-0,6762	-0,6555	-0,4042	-0,5771	0,4534	-4,51302	B
5	MAHASIS WA11	-0,4534	-0,6762	-0,3883	0,5585	0,5771	-0,6762	0,6555	-0,4042	-0,5771	0,4534	-0,93093	B
...
49	MAHASIS WA214	0,4534	0,6762	0,3883	0,5585	0,5771	0,6762	0,6555	-0,4042	-0,5771	0,4534	3,457103	A

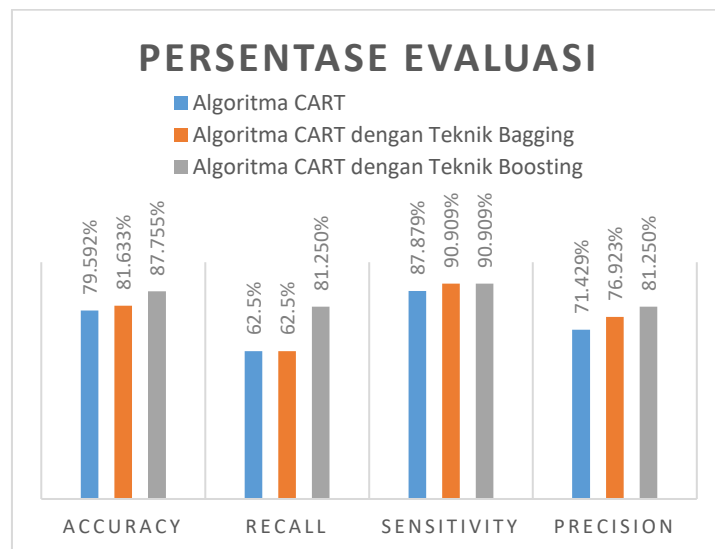
4.5 Evaluation

Evaluation dilakukan dengan menggunakan metode *confusion matrix*. Berikut ini *confusion matrix* dari hasil klasifikasi pada data *testing* dapat dilihat pada Tabel 11.

Tabel 11. *Confusion matrix* hasil klasifikasi pada data *testing*

	TP	FN	FP	TN
Algoritma CART	10	6	4	29
Algoritma CART dengan Teknik Bagging	10	6	3	30
Algoritma CART dengan Teknik Boosting	13	3	3	30

Perbandingan hasil evaluasi dari algoritma *CART*, algoritma *CART* dengan teknik *bagging*, dan algoritma *CART* dengan teknik *boosting* menggunakan *confusion matrix* dapat dilihat pada Gambar 5.



Gambar 5. Persentase evaluasi menggunakan *confusion matrix*

5. KESIMPULAN

Nilai *accuracy* keseluruhan klasifikasi masa studi mahasiswa menggunakan algoritma *CART* sebesar 79,592%, algoritma *CART* dengan teknik *bagging* 81,633%, dan algoritma *CART* dengan teknik *boosting* 87,755%. Nilai *Accuracy* untuk kelas minoritas yaitu kelas “A” atau “Tepat Waktu” dilihat dari nilai *recall*, model klasifikasi algoritma *CART* dengan teknik *boosting* merupakan model klasifikasi yang paling baik di antara tiga model klasifikasi yang diteliti untuk mengatasi kelas tidak seimbang yaitu sebesar 81,25%.

DAFTAR PUSTAKA

- Alfaro, E., Gámez, M., & García, N. (2013). adabag: An R Package for Classification with Boosting and Bagging. *Journal of Statistical Software*, 54(2), 1–35.
- Bisri, A., & Wahono, R. S. (2015). Penerapan Adaboost untuk penyelesaian ketidakseimbangan kelas pada Penentuan kelulusan mahasiswa dengan metode Decision Tree. *Journal of Intelligent Systems*, 1(1), 27–32.
- Chao, W.-L., Liu, J.-Z., & Ding, J.-J. (2013). Facial age estimation based on label-sensitive learning and age-oriented regression. *Pattern Recognition*, 46(3), 628–641.
- Faisal, M. R. (2016). *Data Scienci Klasifikasi dengan Bahasa Pemrograman*. Banjarmasin: INDC.
- Kotsiantis, S. B., & Pintelas, P. E. (2009). Selective costing ensemble for handling imbalanced data sets. *International Journal of Hybrid Intelligent Systems*, 6(3), 123–133.
- Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1), 25–36.
- Lewis, R. J. (2000). An introduction to classification and regression tree (CART) analysis. In *Annual meeting of the society for academic emergency medicine in San Francisco* (Vol. 14).
- Liang, G., & Zhang, C. (2011). An Empirical Study of Bagging Predictors on Medical Data. In *Proceedings of the Ninth Australasian Data Mining Conference* (Vol. 121, pp. 31–40).
- Nasution, N., Djahara, K., & Zamsuri, A. (2015). Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naïve Bayes (Studi Kasus: Fasilkom Unilak). *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 6(2), 1–11.
- Prasetio, R. T., & Pratiwi, P. (2015). Penerapan Teknik Bagging pada Algoritma Klasifikasi untuk Mengatasi Ketidakseimbangan Kelas Dataset Medis. *Jurnal Informatika*, 2(2), 395–403.
- Salmu, S., & Solichin, A. (2017). Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu Menggunakan Naïve Bayes : Studi Kasus UIN Syarif Hidayatullah Jakarta. In *Prosiding Seminar Nasional Multidisiplin Ilmu* (Vol. 22).
- Susanto, S., & Suryadi, D. (2010). *Pengantar Data Mining*. Yogyakarta: Penerbit Andi.
- Timofeev, R. (2004). *Classification an Regresion Trees Theory and Application*. Humboldt University.
- Weiss, G. M., McCarthy, K., & Zabar, B. (2007). Cost- Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs? *DMIN*, 7, 35–41.
- Yamasari, Y., Nugroho, S. M., Suyatno, D. F., & Purnomo, M. H. (2017). Meta-Algoritme Adaptive Boosting untuk Meningkatkan Kinerja Metode Klasifikasi pada Prestasi Belajar Mahasiswa. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 6(3), 333–341.
- Yap, B. W., Rani, K. A., Rahman, H. A. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. In *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)* (pp. 13–22). Singapore: Spriger.