Penerapan Teknik Bagging Berbasis Naïve Bayes untuk Seleksi Penerimaan Mahasiswa

Yum Leifita Nursimpati[1], Aries Saifudin[2]
Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
e-mail: [1]yumleifita@gmail.com, [2]aries.saifudin@unpam.ac.id

Abstract

Students who graduate not on time create an imbalanced ratio between lecturers and students. The current selection system is ineffective because it has not been able to detect prospective students who have the possibility of not being able to complete their education on time so that many students who are accepted do not graduate on time and leave without completing their education. This condition causes a decrease in performance of study programs and institutions. The classification algorithm can use for classifying new students as graduate timely or not. Naïve Bayes classification algorithm can use to classify data in certain classes, using the history of alumni of informatics engineering at Pamulang university as training data and prospective student data as test data. Some attributes used to determine which label class to graduate on time and not on time are gender, school majors, year difference, math grades, English, Indonesian. To improve the results of the classification of Naïve Bayes, Bagging (Bootstrap Aggregating) technique is used. From the test results of the alumni dataset, the informatics study program using bagging techniques as an optimization of the Naïve Bayes classification algorithm has a lower failure rate than without using bagging techniques. The results of the calculation of performance data using bagging techniques can increase accuracy by 2.381% and AUC by 1.470% on the student graduation prediction model for new student selection using the Naïve Bayes classification.

Kata kunci: Bagging, Data Mining, Naïve Bayes, Student Selection

1 Pendahuluan

Seleksi mahasiswa baru adalah acuan utama perguruan tinggi dalam memilah-milah calon mahasiswa sesuai dengan syarat yang ditentukan oleh perguruan tinggi. Penyeleksian mahasiswa baru dilakukan agar mendapatkan calon mahasiswa yang berkualitas. Mahasiswa yang berkualitas akan lulus tepat waktu sehingga membuat kualitas yang baik bagi perguruan tinggi, hal ini dapat meningkatkan akreditasi perguruan tinggi. Perguruan tinggi harus memiliki dan menerapkan kebijakan tentang rekrutmen dan seleksi mahasiswa baru, serta pengelolaan lulusan sebagai satu kesatuan mutu yang terintegrasi, dan menyelenggarakan kegiatan akademik untuk mewujudkan visi, melaksanakan misi, dan mencapai tujuan melalui strategi-strategi yang dikembangkan (BAN-PT, 2011). Upaya perguruan tinggi untuk memperoleh mahasiswa yang bermutu yaitu melalui sistem dan program rekrutmen atau seleksi penerimaan mahasiswa baru.

Universitas Pamulang adalah salah satu perguruan tinggi swasta yang banyak diminati oleh calon mahasiswa, khususnya program studi Teknik Informatika. Mengingat tujuan dari universitas Pamulang yaitu memberikan pendidikan dengan biaya yang murah agar semua kalangan dapat mengenyam pendidikan sampai ke jenjang sarjana. Karena hal ini, maka banyak calon mahasiswa yang coba-coba kuliah sehingga tidak meneruskan studi sampai selesai. Saat ini penerimaan mahasiswa baru di universitas Pamulang dilakukan dengan cara yang kurang efektif dan rumit. Seleksi mahasiswa baru saat ini memerlukan waktu yang lama untuk memutuskan diterima atau tidaknya calon mahasiswa serta memerlukan biaya yang lebih banyak untuk mencetak soal ujian. Sistem yang ada saat ini tidak efektif sehingga banyak mahasiswa yang lulus tidak tepat waktu. Hal ini menyebabkan ketidakseimbangan antara mahasiswa dengan dosen pengajar, mengingat bahwa banyaknya calon mahasiswa yang diterima setiap semesternya. Untuk mengatasi masalah tersebut akan digunakan teknik *data mining* untuk mengklasifikasi calon mahasiswa menggunakan metode klasifikasi. Teknik *data mining* dan *machine learning* dapat digunakan untuk memprediksi berdasarkan data-data masa lalu (Saifudin, 2018).

Untuk melakukan seleksi penerimaan mahasiswa baru dapat menggunakan algoritma klasifikasi *naïve bayes* (Arti, 2009). Di mana akan

dilakukan analisis untuk memperoleh informasi terhadap studi kasus penerimaan mahasiswa baru berdasarkan histori alumni pada program studi Teknik Informatika. *Naive Bayes* dapat melakukan klasifikasi dengan metode probabilitas dan statistik, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya (Bustami, 2013). Algoritma *Naïve Bayes Classifier* bertujuan untuk melakukan klasifikasi data pada kelas tertentu. Unjuk kerja pengklasifikasi diukur dengan nilai *predictive accuracy* (Zhang & Su, 2006). Kelemahan *data mining* terdapat pada *imbalance class* yang merupakan suatu masalah atau tantangan karena biasanya mesin *learning* akan menghasilkan suatu akurasi prediksi yang baik terhadap kelas data latih yang banyak (kelas mayor), sedangkan untuk kelas data latih yang sedikit (kelas minor) akan dihasilkan akurasi prediksi yang buruk (Baizal, Bijaksana, & Nasihati, 2009).

Dalam data mining terdapat dua metode untuk meningkatkan tingat akurasi prediksi dan ketepatan dalam klasifikasi yaitu *Boosting* dan *Bootstrap Aggregating (Bagging)* (Wirayuda, Hidayat, & Shaufiah, 2010). Pemilihan metode *bagging (Bootstrap Aggregating)* sangat tepat untuk melakukan klasifikasi pada dataset. Teknik *Bagging* merupakan metode yang dapat memperbaiki hasil dari algoritma klasifikasi *machine learning* (Wicaksono, Oranova S, & Sawosri, 2010). Dengan penerapan teknik ini maka hasil klasifikasi ataupun prediksi terhadap data memiliki tingkat kegagalan yang lebih rendah.

Berdasarkan uaraian pada latar belakang dapat diidentifikasikan beberapa masalah sebagai berikut: mahasiswa yang lulus tidak tepat waktu membuat ketidakseimbangan antara dosen dan mahasiswa mengingat banyaknya mahasiswa baru yang diterima tiap semesternya, sistem seleksi saat ini tidak efektif karena banyak mahasiswa yang tidak lulus tepat waktu dan keluar tanpa menyelesaikan pendidikanya, pengklasifikasian *Naïve Bayes* dapat digunakan untuk memprediksi (menyeleksi) calon mahasiswa, tetapi hasilnya belum akurat.

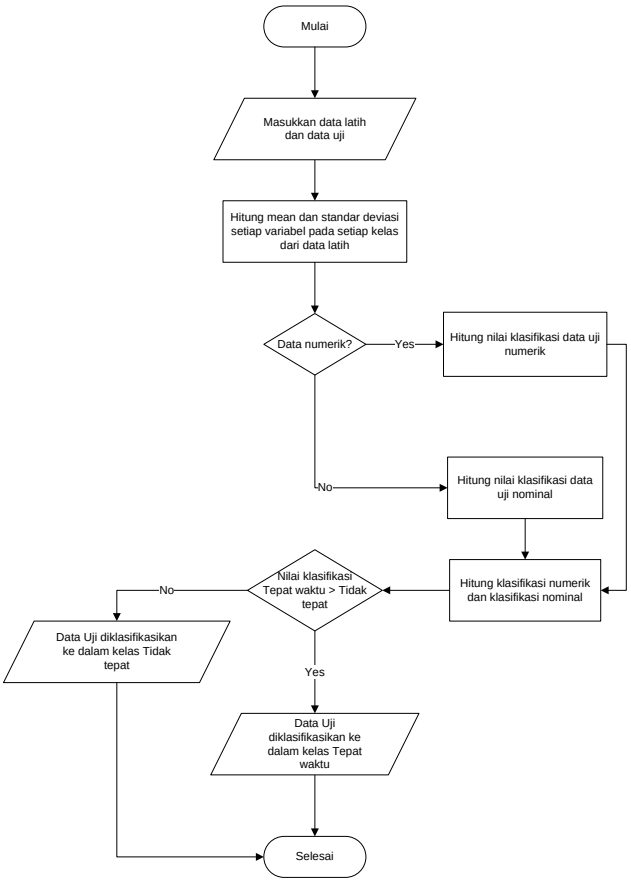
Tujuan dari penelitian ini adalah menerapkan teknik *bagging* untuk memanipulasi data *training*, agar kinerja algoritma pengklasifikasi (*Naïve Bayes*) pada seleksi penerimaan mahasiswa baru dapat meningkat.

2 Penelitian Terkait

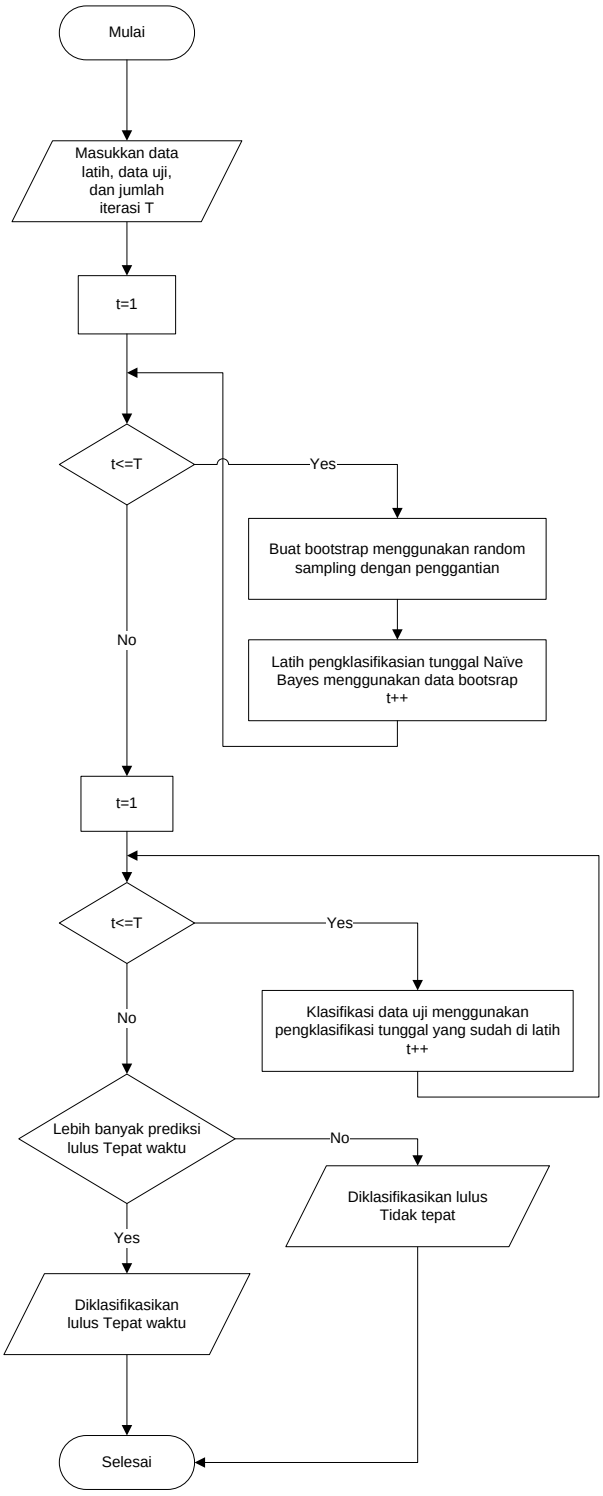
Kelemahan metode *Naïve Bayes* terdapat pada *imbalance class*, dimana akan menghasilkan suatu akurasi prediksi yang baik terhadap kelas data latih yang banyak (kelas mayor), sedangkan untuk kelas data latih yang sedikit (kelas minor) akan dihasilkan akurasi prediksi yang buruk (Baizal, Bijaksana, & Nasihati, 2009). Data latih yang tidak konsisten dapat mempengaruhi hasil dari metode *Naïve Bayes.*

Dalam penelitian ini diusulkan menggunakan teknik *bagging* sebagai perbaikan hasil dari metode *Naïve Bayes.* Ini dikarenakan teknik *Bagging* merupakan metode yang dapat memperbaiki hasil dari algoritma klasifikasi *machine learning* (Wicaksono, Oranova S, & Sawosri, 2010).

Berikut ini adalah diagram alur metode *Naïve Bayes* dan *Naïve Bayes* dengan *Bagging* dalam menyeleksi mahasiswa baru.



Gambar 1 Diagram alur *Naïve Bayes*



Gambar 2 Diagram Alur *Bagging* berbasis *Naïve Bayes*

3 Metode yang Diusulkan

Pada penelitian ini diusulkan yaitu menggunakan metode *Naïve Bayes*. *Naïve Bayes* merupakan teknik *data mining* dengan pendekatan teori probabilitas untuk membangun sebuah model klasifikasi berdasarkan pada kejadian masa lalu yang mempunyai potensi membentuk sebuah objek baru yang dikategorikan sebagai kelas yang memiliki probabilitas terbaik (Turban, Aronson, & Liang, 2007).

Naive Bayes merupakan teknik klasifikasi dengan metode probabilitas dan statistik yang dikemukakan oleh *Thomas Bayes*, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes*. Teorema tersebut dikombinasikan dengan *Naïve* di mana diasumsikan kondisi antara atribut saling bebas. Klasifikasi *Naive Bayes* diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya (Bustami, 2013).

Dalam Bayes (terutama *Naive bayes*), maksud independensi yang kuat adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama (Kusrini & Luthfi, 2009).

Teorema keputusan *bayes* adalah pendekatan statistik yang fundamental dalam pengenalan pola (*pattern recognition*). *Naïve bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai *output*. Dengan kata lain, probabilitas mengamati secara bersama adalah produk dari probabilitas individu (Mujib, Suyono, & Sarosa, 2013). Prediksi *Bayes* didasarkan pada teorema *Bayes* dengan formula umum sebagai berikut:

$$P(H|X) = \frac{P(X|H) \text{ x } P(H)}{P(X)}$$

Penjelasan dari formula tersebut pada tabel 2.4 (Kusrini & Luthfi, 2009).

Tabel 1 Keterangan Naïve Bayes Classifier

Parameter	Keterangan
X	Adalah data sampel dengan kelas (label) yang tidak diketahui.
H	Merupakan hipotesa bahwa X adalah data dengan kelas (label) C.
P(H)	Adalah peluang dari hipotesa H
P(X)	Adalah peluang data sampel yang diamati
P(H|X)	Adalah peluang data sampel X, bila diasumsikan bahwa hipotesa benar (valid)

Jika seseorang berhadapan dengan data kontinu, asumsi khas adalah distribusi Gaussian,

dengan parameter model dari *mean* dan *varians*. *Mean*, μ, dihitung dengan:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Di mana N adalah jumlah sampel dan xi adalah nilai dari suatu contoh yang diberikan. Untuk setiap kelas y_j, probabilitas bersyarat kelas y_j untuk fitur X_i adalah (Salim, 2012):

$$P(X_i = x_i | Y = y_i) = \frac{1}{\sigma_{ij}.\sqrt{2\pi}} \exp^{\frac{(x_i - \mu_{ij})^2}{2\sigma ij^2}}$$

Varian, σ^2 dihitung dengan:

$$\sigma^2 = \frac{1}{(N-1)}\sum_{i=1}^{N}(x_i - \mu)^2$$

Berikut ini adalah algoritma Naïve Bayes (Mulyati, Yulianti, & Saifudin, 2017):

a. Masukan: Data latih T, data uji x
b. Hitung mean (rata-rata) dan standar deviasi setiap kelas
c. Hitung nilai probabilitas data uji untuk setiap kelas
d. Klasifikasikan data uji sesuai nilai probabilitas kelas yang tertinggi
e. Keluaran: Hasil klasifikasi

Untuk meningkatkan akurasi dari proses algoritma klasifikasi, salah satu cara yang dapat dilakukan adalah dengan menggunakan model prediksi yang terdiri atas banyak *classifier* (Tan, Steinbach, & Kumar, 2014). Bagging (*Bootstrap Agregating*) adalah salah satu metode *ensemble* yang cara kerjanya adalah membuat beberapa sampel data baru dari data latih asli. *Bagging* dapat meningkatkan pengklasifikasi Naïve Bayes dan lebih baik dari pada *AdaBoost* (Saifudin & Wahono, 2015). Sampel data dipilih secara acak berdasarkan distribusi *uniform*. Sampel data dibuat dengan cara *sampling with replacement*, yaitu beberapa *record* pada data latih yang sudah pernah diambil untuk satu sampel data bisa diambil lagi untuk sampel data tersebut, atau dengan kata lain pada satu sampel data bisa terdapat record yang nilainya sama.

Sampel himpunan data baru yang dihasilkan disebut dengan *bootstrap sample*. Masing-masing *bootstrap sample* yang dihasilkan kemudian dilatih untuk menghasilkan model klasifikasi (Nuha, Arieshanti, & Purwananto, 2012). *Uniform probability distribution* berarti bahwa setiap *sample* dari *data taraining* asli memiliki kemungkinan yang sama untuk diambil. Secara rata-rata, setiap *bootstrap* mengandung 63% *data training* asli karena setiap elemen *data training* memiliki peluang $1-(1-1/N)^N$ dengan N adalah ukuran *data training* (Wicaksono, Oranova S, & Sawosri, 2010).

Algoritma *Bagging*:

a. *For* i=1 *to* k *do* // k adalah banyaknya *bootstrap*
b. Buat sebuah *bootstrap sample* Di berukuran N
c. Buatlah sebuah *classifier* Ci menggunakan *bootstrap sample* Di
d. *End For*
e. $C^*(x)=\arg\max_y \sum_i \delta\,(C_i(x)=y)$
f. (δ (.)=1) jika argumennya bernilai *true* dan sebaliknya

Algoritma *bagging* adalah algoritma pembuatan *ensemble classifier* menggunakan metode *bagging*. Mula-mula ditentukan dulu banyaknya *bootstrap* atau *classifier* yang akan dibuat yang ditunjukkan pada baris 1. Pada baris 2 sampai 5, untuk setiap iterasi dilakukan pembuatan *bootstrap* menggunakan teknik *sampling with replacement* dengan *uniform probability distribution*. Dari *bootstrap* yang didapatkan pada setiap iterasi, dibuat sebuah *classifier* untuk masing-masing *bootstrap*. Model prediksi yang terbentuk terdiri atas banyak *classifier*. Jika terdapat data yang akan diklasifikasikan atau diprediksi *class* labelnya, maka proses klasifikasi dilakukan dengan melakukan *voting* dari *classifier-classifier* penyusun model prediksi (Wirayuda, Hidayat, & Shaufiah, 2010).

N-fold cross validation merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut masukan yang acak. Metode *cross validation* memberi kesempatan yang sama pada setiap data agar tervalidasi, sehingga kumpulan pelatihan dan validasi dibuat crossover (Yulianti, 2018).

N-fold cross validation diawali dengan membagi data sejumlah *n-fold* yang diinginkan. Dalam proses *cross validation* data akan dibagi dalam n buah partisi dengan ukuran yang sama D_1, D_2, D_3, …, D_n selanjutnya proses testing dan training dilakukan sebanyak n kali. *Cross validation* merupakan pengujian standar yang

dilakukan untuk memprediksi *error rate*. Setiap kelas pada data set harus diwakili dalam proporsi yang tepat antara data *training* dan data *testing*. Data dibagi secara acak pada masing-masing kelas dengan perbandingan yang sama.

Untuk mengurangi bias yang disebabkan oleh sampel tertentu, seluruh proses *training* dan *testing* diulang beberapa kali dengan sampel yang berbeda. Tingkat kesalahan pada iterasi yang berbeda akan dihitung rata-ratanya untuk menghasilkan *error rate* secara keseluruhan (Hastuti, 2012).

Normalisasi min-max yaitu standarisasi data dengan menempatkan data dalam range 0 sampai 1 atau -1 sampai 1. Umumnya nilai $x'_{min,j} = -1$ dan $x'_{max,j} = 1$, atau $x'_{min,j} = 0$ dan $x'_{max,j} = 1$.

$$x'_{i,j} = \left(\frac{x_{i,j} - x_{min,j}}{x_{max,j} - x_{min,j}}\right)(x'_{max,j} - x'_{min,j}) + x'_{min,j}$$

Normalisasi z-index, membuat skala data dengan nilai deviasi yang biasanya merupakan rata-rata dari nilai suatu atribut.

$$x'_{i,j} = \frac{x_{i,j} - \bar{\mu}_j}{\bar{\sigma}_j}$$

Normalisasi skala desimal bekerja dengan menggeser nilai decimal hingga data memiliki satuan yang sama. Di mana h adalah bilangan bulat terkecil sehingga Max (| ν '|) < 1 atau dalam rentang [-1, 1].

$$x'_{i,j} = \frac{x_{i,j}}{10^h}$$

4 Hasil Dan Pembahasan

Untuk menguji aplikasi yang telah dibuat, maka dibuat perhitungan manual terlebih dahulu, hasil hitungan manual akan dibandingkan dengan perhitungan aplikasi. Dataset yang digunakan ditunjukkan pada Tabel 2.

Tabel 2 Dataset Uji Aplikasi

Jk	Jurusan	Selisih_th	Mtk	B.ing	B.ind	Kelulusan
L	ipa	0	6.3	7.8	7.6	Tepat waktu
P	ips	0	6.8	6.2	7	Tepat waktu
P	mm	0	7	8.6	6.8	Tepat waktu
L	mm	0	7.8	7.8	7.6	Tepat waktu
L	ips	0	8	5.6	7.4	Tidak tepat
P	ipa	0	8	8.4	8.6	Tepat waktu
L	ips	0	8.3	7.2	7.6	Tidak tepat
L	mm	0	8.3	7.4	5.2	Tidak tepat
P	ipa	0	8.3	8	8.4	Tepat waktu
P	ipa	0	8.5	7.8	8	Tepat waktu
P	ips	0	8.8	7.8	7.2	Tepat waktu
L	mm	0	8.8	7.8	7.8	Tepat waktu
L	ipa	0	9	8.6	8.6	Tidak tepat
L	ips	0	9.5	5.2	7.6	Tepat waktu
L	mm	0	9.8	7.4	5.6	Tepat waktu
P	ips	1	6.5	7.6	6.4	Tidak tepat
P	ipa	1	7.3	7.6	5.8	Tepat waktu
L	ips	2	5.2	7.4	7	Tidak tepat
L	ips	3	8	6	5.4	Tidak tepat
L	ipa	4	6.7	6.33	6.83	Tidak tepat

Sebelum dilakuakan perhitungan, dataset dilakukan standarisasi dengan teknik min-max dengan nilai minimal -1 dan maksimal 1 sehingga nilainya pada tipe numerik seperti Tabel 4.2. Teknik validasi yang digunakan adalah *10-fold cross validation*, sehingga dataset dibagi menjadi 10, kemudian diambil 1 data untuk dijadikan data uji kemudian sisanya dijadikan data latih.

Tabel 3*Normalisasi Dataset Numerik*

split	Mtk	b.ing	b.ind	kelulusan
1	-0.52173913	0.529411765	0.411764706	Tepat waktu
	-0.304347826	-0.411764706	0.058823529	Tepat waktu
2	-0.217391304	1	-0.058823529	Tepat waktu
	0.130434783	0.529411765	0.411764706	Tepat waktu
3	0.217391304	-0.764705882	0.294117647	Tidak tepat
	0.217391304	0.882352941	1	Tepat waktu
4	0.347826087	0.176470588	0.411764706	Tidak tepat
	0.347826087	0.294117647	-1	Tidak tepat
5	0.347826087	0.647058824	0.882352941	Tepat waktu
	0.434782609	0.529411765	0.647058824	Tepat waktu
6	0.565217391	0.529411765	0.176470588	Tepat waktu
	0.565217391	0.529411765	0.529411765	Tepat waktu
7	0.652173913	1	1	Tidak tepat
	0.869565217	-1	0.411764706	Tepat waktu
8	1	0.294117647	-0.764705882	Tepat waktu
	-0.434782609	0.411764706	-0.294117647	Tidak tepat
9	-0.086956522	0.411764706	-0.647058824	Tepat waktu
	-1	0.294117647	0.058823529	Tidak tepat
10	0.217391304	-0.529411765	-0.882352941	Tidak tepat
	-0.347826087	-0.335294118	-0.041176471	Tidak tepat

Bagian (*split*) ke-1 sebagai data uji dan lainnya sebagai data latih. Pertama dilakukan perhitungan probabilitas awal masing-masing kelas tanpa memandang bukti apapun. Selanjutnya dilakukan perhitungan *mean* (μ) dan standar deviasi (σ) untuk masing-masing kelas berdasarkan bukti pada setiap attribut.

Probabilitas awal

$$P_{(tepat\ wkt)} = \frac{10}{18} = 0.555555556$$

$$P_{(tdk\ tepat)} = \frac{8}{18} = 0.444444444$$

Tabel 4 Probabilitas Fitur Kelas Nominal Bagian (*Split*) Ke-1

Jenis Kelamin		Jurusan	
Tepat waktu	**Tidak tepat**	**Tepat waktu**	**Tidak tepat**
L=4 P=6	L=7 P=1	Ipa=4 Ips=2 Mm=4	Ipa=2 Ips=5 Mm=1
P(JK=L|tepat wkt)=4/10=0.4 P(JK=P|tepat wkt)=6/10=0.6	P(JK=L|tdk tepat) = 7/8 = 0.875 P(JK=P|tdk tepat) = 1/8 = 0.125	P(Jrsn=ipa|tepat wkt) = 4/10 = 0.4 P(Jrsn=ips|tepat wkt) = 2/10 = 0.2 P(Jrsn=mm|tepat wkt) = 4/10 = 0.4	P(Jrsn=ipa|tdk tepat) = 2/8 = 0.25 P(Jrsn=ips|tdk tepat) = 5/8 = 0.625 P(Jrsn=mm|tdk tepat) = 1/8 = 0.125

Tabel 5 Probabilitas Fitur Kelas Numerik Bagian (*Split*) Ke-1

class	statistik	Selisih tahun	matematika	b.inggris	b.indonesia
Tepat waktu	μ	-0.95	0.382608696	0.435294118	0.258823529
	σ^2	0.025	0.151144717	0.297731642	0.353863899
	σ	0.158113883	0.38877335	0.54564791	0.594864605
Tidak tepat	μ	-0.375	-3.60822E-16	0.068382353	-0.056617647
	σ^2	0.625	0.295976235	0.330538186	0.442945502
	σ	0.790569415	0.54403698	0.574924504	0.66554151

Berdasarkan pembagian *fold cross validation*, maka data bagian (*split*) pertama yang dijadikan data uji dihitung sebagai berikut:

$$P(X_i = x_i | Y = y_i) = \frac{1}{\sigma_{ij}.\sqrt{2\pi}} \exp^{-\frac{(x_i - \mu_{ij})^2}{2\sigma ij^2}}$$

$P\ (X|tepat\ waktu)$
$= P(jk = L|tepat\ wkt)x\ \ P(jurusan = ipa|tepat\ wkt)\ P(selisih$
$= -1|tepat\ waktu)x\ P(mtk$
$= -0.52173913\ |tepat\ waktu)\ xP(b.ing$
$= 0.529411765|tepat\ waktu)\ x\ P(b.indo$
$= 0.411764706|tepat\ waktu)\ x\ P(tepat\ wkt)$

$P\ (X|tdk\ tepat)$
$= P(jk = L|tdk\ tepat)\ \ x\ \ P(jurusan = ipa|tdk\ wkt)\ x\ P(selisih$
$= -1|tdk\ tepat)x\ \ P(mtk$
$= -0.52173913|tdk\ tepat)\ xP(b.ing$
$= 0.529411765|tdk\ tepat)\ x\ P(b.indo$
$= 0.411764706\ |tdk\ tepat)\ x\ P(tdk\ tepat)$

Karena probabilitas tidak tepat waktu (0.003916261) lebih besar dari pada probabilitas tepat waktu (0.001142526), maka data/*record*/fitur pertama diklasifikasikan lulus tidak tepat waktu. Dengan cara yang sama dilakukan perhitungan untuk data/*record*/fitur kedua dan seterusnya.

Tabel 6 Tabel Rekapitulasi Perhitungan Naïve Bayes

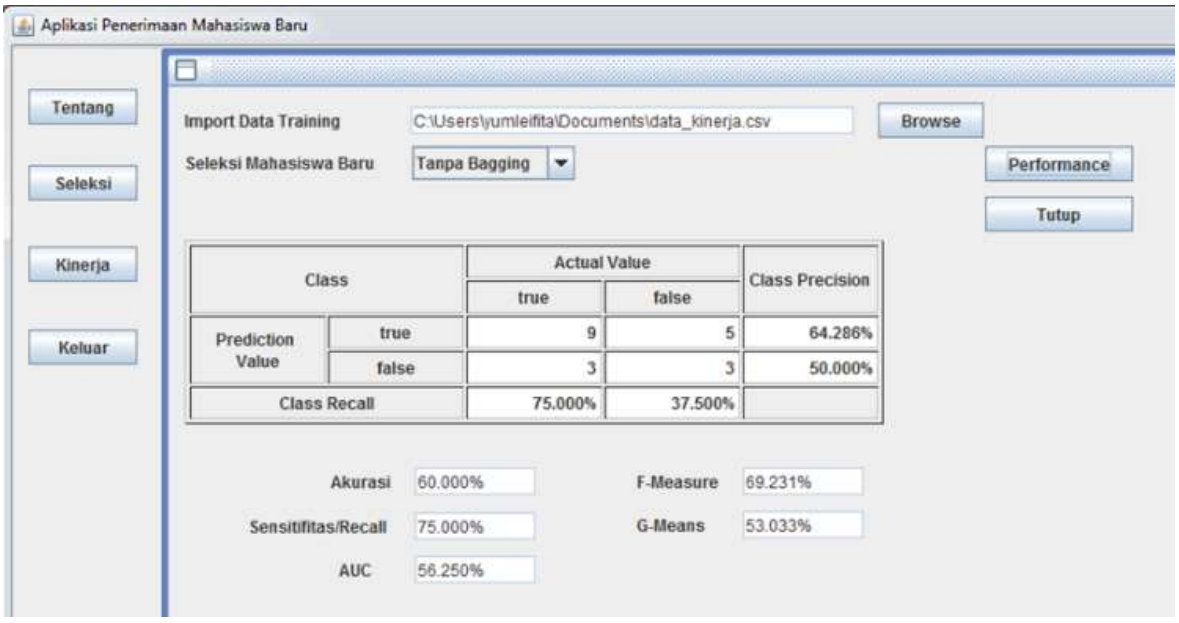
Split	Aktual	Prediksi
1	Y	Y
	Y	Y
2	Y	Y
	Y	Y
3	N	Y
	Y	Y
4	N	Y
	N	Y
5	Y	Y
	Y	Y
6	Y	Y
	Y	Y
7	N	Y
	Y	N
8	Y	Y
	N	Y
9	Y	N
	N	N
10	N	N
	N	N

Berdasarkan hasil validasi Tabel 6 dibuat tabel confusion matrix seperti Tabel 7, dan dilakukan perhitungan kinerja model.

Tabel 7 Hasil Klasifikasi Perhitungan Manual Dengan 10-Fold Cross Validation

Class		Actual		Class Precision
		Tepat Waktu	Tidak Tepat	
Prediction	Tepat Waktu	9 (TP)	5 (FP)	64,286 %
	Tidak Tepat	3 (FN)	3 (TN)	50,000 %
Class Recall		75,00 %	37,50 %	

Untuk menguji aplikasi yang telah dibuat, hasil perhitungan di atas dibandingkan dengan hasil perhitungan dari aplikasi. Hasil perhitungan dari aplikasi dengan menggunakan metode Naïve bayes seperti Gambar 3.



Gambar 3 Hasil Perhitungan Kinerja Model dengan Naïve Bayes

5 Kesimpulan

Berdasarkan penulisan dan penelitian yang telah diuraikan, maka dapat dibuat beberapa kesimpulan yaitu:

a. Teknik *data mining* dapat menyeleksi mahasiswa baru dengan efektif, karena dapat memprediksi ketepatan waktu kelulusan berdasarkan data-data calon mahasiswa.
b. Boosting tidak dapat bekerja dengan baik untuk algoritma klasifikasi Naïve Bayes, dikarenakan *Naïve Bayes* adalah pengklasifikasian yang stabil dengan bias yang kuat (Ting & Zheng, 2003). Adacost adalah varian lain dari Adaboost, AdaCost dapat memperbaiki teknik boosting (Sun, Kamel, Wong, & Wang, 2007) maka perlu penelitian lanjutan pada teknik adacost sebagai perbaikan dari algoritma klasifikasi Naïve Bayes.

Referensi

Arti, Y. (2009). Penentuan Tingkat Keberhasilan Mahasiswa Tingkat I IPB Menggunakan Induksi Pohon Keputusan dan Bayesian Classifier. *IPB journal*, 1-37.

Baizal, Z. A., Bijaksana, M. A., & Nasihati, I. R. (2009). Penggunaan Metode Bagging Dengan Menerapkan Data Balancing Pada Churn Prediction Untuk Perusahaan Telekomunikasi. *Aplikasi Teknologi Komunikasi*, 134-139.

BAN-PT. (2011). *Akreditasi Institusi Perguruan Tinggi - Buku II Standar dan Presedur.* Jakarta.

Bustami. (2013). Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi. *TECHSI:Jurnal Penelitian Teknik Informatika*, 128-146.

Hastuti, K. (2012). Analisis Komparasi Algoritma Klasifikasi Data Mining untuk Prediksi Mahasiswa Non Aktif. *Prosiding Semantik* (pp. 241-249). Semarang: Universitas Dian Nuswantoro.

Kusrini, & Luthfi, E. T. (2009). *Algoritma Data Mining.* Yogyakarta: Andi Publisher.

Mujib, R., Suyono, H., & Sarosa, M. (2013). Penerapan Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *Jurnal EECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems), 7*(1), 59-64.

Mulyati, S., Yulianti, Y., & Saifudin, A. (2017). Penerapan Resampling dan Adaboost untuk Penanganan Masalah Ketidakseimbangan Kelas Berbasis Naïve Bayes pada Prediksi Churn Pelanggan. *Jurnal Informatika Universitas Pamulang, 2*(4), 190-199.

Nuha, M. U., Arieshanti, I., & Purwananto, Y. (2012). Pengembangan Perangkat Lunak Prediktor Kebangkrutan Menggunakan Metode Bagging Nearest Neighbor Support Vector Machine. *Jurnal Teknik POMITS, 1*(1), 1-6.

Saifudin, A. (2018). Metode Data Mining untuk Seleksi Calon Mahasiswa pada Penerimaan Mahasiswa Baru di Universitas Pamulang. *Jurnal Teknologi, 10*(1), 25-36.

Saifudin, A., & Wahono, R. S. (2015). Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. *Journal of Software Engineering, 1*(1), 28-37.

Salim, Y. (2012). Penerapan Algoritma Naive Bayes untuk Penentuan Status Turn-Over Pegawai. *Media Sains, 4*(2), 196-205.

Sun, Y., Kamel, M. S., Wong, A. K., & Wang, Y. (2007). AdaCost : Misclassification Cost-Sensitive Boosting. *Pattern Recognition 40*, 3358-3378.

Tan, P.-N., Steinbach, M., & Kumar, V. (2014). *Introduction to Data Mining.* Essex: Pearson Education Limited.

Ting, K. M., & Zheng, Z. (2003). A Study Of AdaBoost With Naive Bayesian Classifiers : Weakness and Improvement. *Computational Intelligence, Volume 19, Number 2*, 186-199.

Turban, E., Aronson, J. E., & Liang, T. P. (2007). *Decision Support Systems and Intelligent Systems* (7 ed.). Yogyakarta: Andi Publisher.

Wicaksono, S. A., Oranova S, D., & Sawosri. (2010). Pembangunan Model Prediksi Defect Menggunakan Metode Ensemble Decision Tree dan Cost Sensitive Learning. *Jurnal EECCIS Vol.IV No.1*, 1-7.

Wirayuda, T. A., Hidayat, D., & Shaufiah. (2010). Analisis Dan Implementasi Metode Bootstrap Aggregating (Bagging) Pada Model Artificial Neural Network Dengan Studi Kasus Klasifikasi Penanganan Tindak Lanjut Pasien Unit Gawat Darurat. *posiding ITT*, 1-9.

Yulianti. (2018). Metode Data mining Untuk prediksi Churn Pelanggan. *Jurnal ICT Akademi Telkom Jakarta, 9*(16), 46-52.

Zhang, H., & Su, J. (2006). Learning Probabilistic Decision Trees For AUC. *Pattern Recognition Letters 27*, 892-899.