



Perbandingan Algoritma *Support Vector Machine* dan *AdaBoost* Dalam Memprediksi Waktu Kelulusan Mahasiswa

Hary Sabita*¹, Sherli Trisnawati²

^{1,2}Institut Informatika dan Bisnis Darmajaya; Jl ZA Pagar Alam N0 93 Gedong Meneng Kec Rajabasa Bandar Lampung, Lampung 35141, (0721)787214

³Jurusan Teknik Informatika, Institut Informatika dan Bisnis Darmajaya, Lampung

*Email Penulis Korespondensi: hary.sabita@darmajaya.ac.id

Abstrak

Fokus penelitian ini adalah untuk membuat sebuah model machine learning dengan menggunakan dua metode yang berbeda. Metode yang dikembangkan akan menghasilkan model yang nantinya akan dapat memprediksi waktu kelulusan mahasiswa. Metode yang di gunakan adalah klasifikasi dengan membandingkan dua algoritma, yaitu Support Vector Machine (SVM) dan AdaBoost. Penelitian ini menggunakan 660 data internal di prodi Teknik Informatika IIB Darmajaya dan dianalisa dengan menggunakan bahasa pemrograman python. Hasil akhir dari penelitian ini adalah akurasi sebesar 0,73 untuk algoritma AdaBoost dan 0,62 untuk SVM. Hasil penelitian ini sangat berguna untuk memprediksi waktu kelulusan mahasiswa berdasarkan 6 variabel independent, yaitu; pendidikan orang tua, pendapatan orang tua, jalur pendidikan, kegiatan internal dan eksternal yang diikuti serta ipk mahasiswa.

Kata kunci—*Machine Learning, Support Vector Machine, AdaBoost, Python, Waktu Kelulusan*

Abstract

The focus of reaseach build a machine learning using different two models. These methods will produce models that able to predict graduation of student. The Classification is used as a method with compare the two algorithms, they were Supoort Vector Machine (SVM) and AdaBoost. This research used 660 internal datas at Informatics Engineering of IIB Darmajaya and its analyse used the language programming of python. The result of this research was an accuracy model, it was 0,73 for AdaBoost model and 0,62 for SVM model. This result is useful to predict the graduation time of student with according to 6 variables, they were; education of parents, income of parent, the way of education, internal student activities, external student activities and GPA of student

Keywords—*3-5 Machine Learning, Support Vector Machine, AdaBoost, Python, Graduation Time*

1. PENDAHULUAN

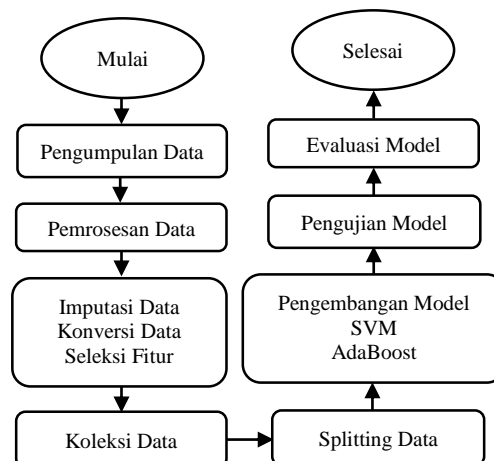
Salah satu aset terpenting didalam sebuah perguruan tinggi adalah mahasiswa [1]. Jumlah mahasiswa aktif yang terdaftar dipangkalan data PDDIKTI ada sekitar 9.534.695 [1]. Perguruan tinggi memiliki tanggung jawab penting yaitu bagaimana mengelola mahasiswa aktif tersebut sehingga bisa lulus tepat waktu [2]. Salah satu indikator keberhasilan sebuah perguruan tinggi adalah jumlah lulusan tidak berbeda jauh dengan jumlah saat mahasiswa masuk. BAN PT sebagai salah satu badan yang menilai akreditasi sebuah prodi maupu perguruan tinggi, juga menjadikan ini sebagai salah satu aspek penting didalam melakukan penilaian [1]. Tidak hanya itu, kelulusan tepat waktu juga akan menguntungkan mahasiswa, dari segi biaya dan juga waktu perkuliahan.

Melihat kondisi yang ada, penelitian ini fokus pada pengembangan model machine learning untuk memprediksi waktu kelulusan mahasiswa [3] [2]. Dengan menggunakan metode klasifikasi, penelitian ini akan membandingkan dua algoritma yaitu *Support Vector Machine* (SVM) dan *AdaBoost* [4] [5]. Teknik machine learning digunakan karena data yang akan dianalisa cukup banyak [4] [6]. Prodi Teknik Informatika pada IIB Darmajaya dijadikan sumber data didalam penelitian ini. Penelitian ini menggunakan 6 variabel independet, yaitu pendidikan orang tua, pendapatan orang tua, jalur pendidikan, kegiatan internal dan eksternal yang diikuti serta ipk mahasiswa [3].

Penelitian ini merupakan kelanjutan serta pengembangan dari penelitian sebelumnya yang pernah dilakukan, diantaranya adalah pada tahun 2022, sebuah artikel yang dibuat oleh Amirudin & Ishak pada *Jambura Journal of Electrical and Electronics Engineering* dengan judul *Implementasi Seleksi Fitur Klasifikasi Waktu Kelulusan Mahasiswa Menggunakan Correlation Matrix with Heatmap*. Metode yang digunakan adalah klasifikasi dengan algoritma *Supprot Vector Machine* (SVM) dan algoritma *Logistic Regression* dan *Decission Tree* [3]. Penelitian ini sangat sejalan dengan penelitian sebelumnya yaitu fokus pada melakukan prediksi waktu kelulusan mahasiswa. Perbedaan dengan penelitian sebelumnya adalah algoritma yang digunakan. Pada penelitian ini, algoritma SVM di pilih karena pada beberapa penelitian sebelumnya algoritma ini memiliki nilai akurasi yang paling tinggi. Sementara, algoritma yang kedua yaitu *AdaBoost*, dipilih karenanan memiliki metode yang berbeda dengan algoritma tunggal lainnya (SVM, LR atau DT) yaitu adanya perlakuan *boosting* [7] [8] [9].

2. METODE PENELITIAN

Penelitian ini dibagi ke dalam tiga tahapan. Tahapan tersebut adalah; (1). Pengumpulan data, (2). Pembersihan data, dan (3). Pengembangan dan evaluasi model. Berikut ini adalah gambar 1 yang akan menggambarkan langkah-langkah penelitiannya.



Gambar 1. Langkah Penelitian

Gambar 1 diatas adalah langkah penelitian yang dilakukan. Langkah pertama akan dimulai dari pengumpulan data (*rawdata*). Setelah *rawdata* terkumpul, selanjutnya dilakukan proses pembersihan data. Proses ini akan dimulai dari mengisi data yang kosong (*imputasi data*), mengkonversi data dan melakukan seleksi fitur. Akhir dari proses ini adalah sebuah dataset yang akan digunakan untuk membangun model. Sebelum model dibangun, maka data akan dibagi terlebih dahulu (*splitting data*). Selanjutnya model yang telah terbentuk untuk masing-masing algoritma akan di uji dan di evaluasi.

2.1 Pengumpulan Data

Proses pertama pada penelitian ini adalah melakukan pengumpulan data. Pengumpulan data dilakukan di Prodi Teknik Informatika IIB Darmajaya, Lampung. Data diambil dalam *range* waktu lulusan 2015 – 2019. Berikut ini tabel 1, informasi detail dari data yang dikumpulkan. Tabel 1 ini menjelaskan tentang nama variabel serta keterangannya. Variabel ini berjumlah 21. 20 variabel nantinya akan digunakan untuk variabel *independent* dan 1 kolom akan digunakan sebagai variabel *dependent*.

Tabel 1. Informasi Data Penelitian

No	Nama Kolom	Keterangan
1.	nama	Nama mahasiswa
2.	jk	Jenis Kelamin
3.	status_mahasiswa	Status Mahasiswa
4.	tah_masuk	Tahun Masuk
6.	pendidikan_ortu	Pendidikan Akhir Orang Tua
7.	pendapatan_ortu	Pendapatan Orang Tua
8.	jalur_pendidikan	Jalur pendidikan yang ditempuh selama kuliah
9.	keg_intern	Status keaktifan dikegiatan internal kampus
10.	keg_ekstern	Status keaktifan dikegiatan eksternal kampus
11.	umur	Usia alumni
12.	status_pernikahan	Status pernikahan
13.	ips_1 – ips_8	Nilai Indeks Prestasi Semester alumni dari semester 1 – 8
14.	ipk	Indeks Prestasi Kumulatif alumni
15.	status_lulus	Target data

2.2 Pemrosesan Data

Pada penelitian ini, ada 3 tahapan yang dilakukan dalam pemrosesan data, yaitu; (1) Imputasi data, (2) Konversi data, serta (3) Seleksi fitur. Semua proses ini dilakukan secara berurut, mulai dari poin 1 hingga 3. Berikut ini masing-masing penjelasannya;

2.2.1 Imputasi data

Aktifitas yang dilakukan pada proses ini adalah mengisi data yang masih kosong. Data yang dikumpulkan terdiri dari 2 kategori, yaitu data numerik dan kategori. Berikut gambar 2, potongan *snippet script program* yang menampilkan kondisi data, tipe dan jumlah kosongnya.

jml_data_kosong		data.dtypes	
No	0	No	int64
Nama	0	Nama	object
jk	0	jk	object
status_mahasiswa	0	status_mahasiswa	object
tah_masuk	0	tah_masuk	int64
pendidikan_ortu	20	pendidikan_ortu	object
pendapatan_ortu	0	pendapatan_ortu	int64
jalur_pendidikan	0	jalur_pendidikan	object
keg_intern	0	keg_intern	object
keg_ekstern	0	keg_ekstern	object
umur	0	umur	int64
status_pernikahan	0	status_pernikahan	object
ips_1	0	ips_1	float64
ips_2	2	ips_2	float64
ips_3	0	ips_3	float64
ips_4	0	ips_4	float64
ips_5	0	ips_5	float64
ips_6	0	ips_6	float64
ips_7	0	ips_7	float64
ips_8	0	ips_8	float64
ipk	0	ipk	float64
status_lulus	0	status_lulus	object

Gambar 2. Tipe dan Jumlah Data Kosong

Gambar 2 diatas menjelaskan tentang jumlah nilai data yang kosong pada setiap variabel serta tipe datanya. Terlihat bahwa variabel 'pendidikan_ortu' memiliki 20 data yang masih kosong, sementara variabel 'ips_2' memiliki 2 data kosong. Teknik yang dilakukan untuk mengisi data kosong pada tipe data numerik (yaitu; pada kolom 'ips_2') yang berjumlah 2, adalah dengan menggunakan nilai rata-rata (*mean*) pada kolom tersebut. Berikut gambar 3, yaitu potongan *script* yang digunakan untuk menghasilkan nilai rata-ratanya. *Script* ini akan menghasilkan nilai pada data yang kosong. Nilai ini berasal dari rata-rata pada kolom tersebut.

```
imputasi_ips_2.iloc[[0,12],[13]]
```

ips_2	
0	2.87997
12	2.87997

Gambar 3. Nilai Rata-Rata yang Dihasilkan

Sementara untuk mengisi data kosong pada tipe data kategorial (yaitu; pada pendidikan_ortu) yang berjumlah 20, adalah dengan menggunakan data yang sering muncul *modus* pada kolom tersebut. Berikut ini gambar 4, yaitu potongan *script* untuk menghasilkan nilai tersebut. Gambar 4 ini menjelaskan tentang nilai modulus yang dihasilkan untuk mengisi data kosong pada kolom tersebut. Nilai modulus yang diperoleh adalah 'SMA'.

```
po_1 = data['pendidikan_ortu']
po_2 = pd.DataFrame(po_1)

from sklearn.impute import SimpleImputer
imp_pend_ortu = SimpleImputer(strategy='most_frequent')
imp2_pend_ortu = imp_pend_ortu.fit_transform(po_2)
imp2_pend_ortu_2 = pd.DataFrame(imp2_pend_ortu)
imp2 = imp2_pend_ortu_2.iloc[[25,27,47,49,162,164,184,186,299,301,321,323,436,438,458,460,573,575,595,597],:]
imp2.rename(index=str, columns={0:'pendidikan_ortu'})
```

pendidikan_ortu			
25	SMA	321	SMA
27	SMA	323	SMA
47	SMA	436	SMA
49	SMA	438	SMA
162	SMA	458	SMA
164	SMA	460	SMA
184	SMA	573	SMA
186	SMA	575	SMA
299	SMA	595	SMA
301	SMA	597	SMA

Gambar 4. Nilai Modus yang Dihasilkan

2.2.2 Konversi Data

Untuk mempermudah dalam menganalisa data, semua data harus dalam bentuk numerik. Pada data penelitian ini, ada beberapa kolom yang masih bernilai kategorial. Berikut ini tabel 2 yang berikis perubahan konversi data tersebut.

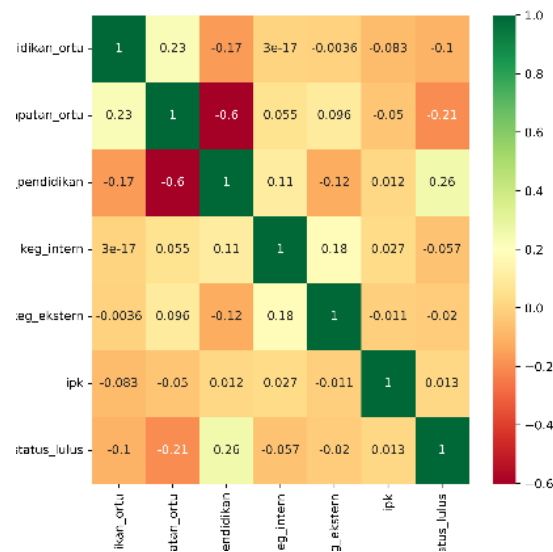
Tabel 2. Perubahan Data Kategorial ke Numerik

No	Nama Kolom	Data Awal	Perubahan Data
1.	jenis_kelamin	Perempuan & Laki-laki	Perempuan =0, Laki-laki =1
2.	pendidikan_ortu	SMA & Sarjana	SMA = 0, Sarjana = 1
3.	jalur_pendidikan	Mandiri, Yayasan & Bidik Misi	Mandiri = 0, Yayasan = 1, Bidik Misi = 2
4.	keg_intern	Tidak, Ya	Tidak = 0, Ya = 1
5.	keg_ekstern	Tidak, Ya	Tidak = 0, Ya = 1
6.	status_lulus	Tidak Tepat, Tepat	Tidak Tepat = 0, Tepat = 1

Tabel 2 diatas menjelaskan tentang konversi data kategorial. Variabel yang masih memiliki nilai kategorial akan dikonversi menjasi numerik. Nilai numerik yang diberikan dalam range 0, 1 dan 2.

2.2.3 Seleksi Fitur

Penelitian ini hanya menggunakan 6 variabel data independent dari 20 data variabel yang ada. Proses yang dilakukan pada tahapan ini adalah dengan menggunakan korelasi matrik *heatmap*. Hasil korelasi antara 6 variabel dengan variabel target ('status_lulus') tersebut adalah sebagai berikut.



Gambar 5. Korelasi 6 Variabel Independent

Penjelasan detail dari gambar 5 diatas, bisa dilihat pada tabel 3 berikut;

Tabel 3. Nilai Korelasi 6 Variabel Independent

variabel	status_lulus
pendidikan_ortu	-0,1
pendapatan_ortu	-0,21
jalur_pendidikan	0,26
keg_intern	-0,057
keg_ekstern	-0,02
ipk	0,013

Pada tabel 3 diatas, terlihat bahwa variabel jalur pendidikan yang memiliki nilai korelasi tinggi yaitu sebesar 0,26.

2.3 Koleksi Data

Seleksi data adalah data akhir yang akan digunakan untuk dianalisa. Data ini selanjutnya yang akan digunakan sebagai dataset yang akan digunakan untuk membangun model. Berikut ini gambar 6, yaitu potongan *script* yang menampilkan variabel serta isi dari dataset.

	pendidikan_ortu	pendapatan_ortu	jalur_pendidikan	keg_intern	keg_ekstern	ipk	status_lulus
0	1.0	7000000	0	1	0	2.75	1
1	0.0	4750000	1	1	1	3.39	1
2	1.0	5000000	0	1	0	2.81	0
3	0.0	3000000	2	1	0	2.82	0
4	0.0	2750000	1	1	0	3.09	1
...
655	0.0	5500000	0	0	1	3.02	0
656	0.0	5000000	0	1	1	3.34	0
657	0.0	5000000	0	1	0	3.63	1
658	0.0	5000000	0	1	1	3.37	0
659	0.0	6000000	0	1	1	2.93	0

660 rows × 7 columns

Gambar 6. Data Akhir untuk Dianalisa

Gambar 6 diatas adalah kondisi data akhir data yang telah di lakukan *preprocessing*. Variabel yang digunakan berjumlah 7. Variabel yang akan digunakan sebagai independent berjumlah 6, sementara variabel yang digunakan sebagai dependent berjumlah 1. Dataset tersebut selanjutnya akan digunakan untuk membangun model.

2.4 Splitting Data

Data yang akan dianalisa dipisah menjadi 2 bagian, yaitu data latih (*training*) 70% (462 data) dan data uji (*testing*) 30% (198). Komposisi persentase berdasarkan pembagian data pada umumnya yang dilakukan oleh penelitian sebelumnya. Berikut gambar 7, yaitu potongan *script* yang memperlihatkan hasil dari proses tersebut.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

X_train					
	pendidikan_ortu	pendapatan_ortu	jalur_pendidikan	keg_intern	keg_ekstern
317	1	5000000	0	0	0
172	1	5000000	0	1	0
97	0	5000000	0	0	1
391	0	7500000	0	1	1
640	1	5000000	0	1	0
...
9	1	4500000	2	1	1
359	1	7500000	0	1	1
192	0	5000000	0	0	0
629	0	3500000	1	0	0
559	0	8000000	0	1	1
462 rows × 5 columns					
X_test					
	pendidikan_ortu	pendapatan_ortu	jalur_pendidikan	keg_intern	keg_ekstern
541	0	5000000	0	1	0
103	0	5000000	0	1	1
14	0	2500000	1	0	0
526	0	5000000	0	0	1
298	1	5000000	0	1	1
...
455	0	3600000	1	1	0
132	0	5000000	0	1	0
173	0	6000000	0	1	0
178	0	4700000	1	1	0
610	0	3000000	1	1	0
198 rows × 5 columns					

Gambar 7. Jumlah Data Latih dan Uji

Gambar 7 diatas adalah komposisi jumlah data latih (*training*) dan data uji (*testing*). Data latih dan data uji dipilih secara acak. Jumlah data sesuai dengan komposisi persentase yang telah ditentukan.

2.5 Pengembangan Model

Model dibangun setelah dataset selesai dibentuk. Pengembangan model dilakukan melalui beberapa tahapan/ proses. Berikut ini adalah tahapan dan proses yang dilakukan pada pengembangan model *machine learning*.

2.5.1 Distribusi Nilai

Proses pertama adalah melakukan distribusi nilai. Distribusi nilai dilakukan agar komposisi data bisa sesuai pada saat dilakukan pemodelan. Berikut ini gambar 8, yaitu potongan *script* program yang menampilkan hasil distribusi nilai.


```

from sklearn.preprocessing import StandardScaler

std_X = StandardScaler()
X_train_svm = std_X.fit_transform(X_train)
X_test_svm = std_X.transform(X_test)

X_train_svm

array([[ 1.6881943 , -0.05084724, -0.42309507, -1.52438217, -0.8698586 ],
       [ 1.6881943 , -0.05084724, -0.42309507,  0.65600347, -0.8698586 ],
       [-0.59234888, -0.05084724, -0.42309507, -1.52438217,  1.14961213],
       ...,
       [-0.59234888, -0.05084724, -0.42309507, -1.52438217, -0.8698586 ],
       [-0.59234888, -1.23728275,  1.87655108, -1.52438217, -0.8698586 ],
       [-0.59234888,  2.32202379, -0.42309507,  0.65600347,  1.14961213]])

std_X = StandardScaler()
X_train_adaboost = std_X.fit_transform(X_train)
X_test_adaboost = std_X.transform(X_test)

X_train_adaboost

array([[ 1.6881943 , -0.05084724, -0.42309507, -1.52438217, -0.8698586 ],
       [ 1.6881943 , -0.05084724, -0.42309507,  0.65600347, -0.8698586 ],
       [-0.59234888, -0.05084724, -0.42309507, -1.52438217,  1.14961213],
       ...,
       [-0.59234888, -0.05084724, -0.42309507, -1.52438217, -0.8698586 ],
       [-0.59234888, -1.23728275,  1.87655108, -1.52438217, -0.8698586 ],
       [-0.59234888,  2.32202379, -0.42309507,  0.65600347,  1.14961213]])

```

Gambar 8. Distribusi Data Model SVM dan AdaBoost

Gambar 8 adalah hasil dari pendistribusian nilai. Proses pertama adalah melakukan *import library* dan dan men-skala nilai untuk masing-masing model. Pada proses berikutnya adalah menampilkan nilai X_{train} untuk masing-masing model.

2.5.1.2 Fitting Model

Proses berikutnya adalah melakukan *fitting* model. Nilai yang telah terdistribusi dimasukkan ke dalam fungsi *SVM Classifier* untuk model SVM dan *AdaBoostClassifier* untuk model AdaBoost. Berikut ini gambar 9, yaitu hasil akhir proses *fitting* model.

<pre> from sklearn.svm import SVC klasifikasi_svm = SVC(kernel='linear', random_state=0) klasifikasi_svm.fit(X_train_svm, y_train) </pre>	<pre> from sklearn.ensemble import AdaBoostClassifier klasifikasi_adaboost = AdaBoostClassifier(n_estimators=9) klasifikasi_adaboost.fit(X_train_adaboost, y_train) </pre>
<div style="border: 1px solid black; padding: 2px; width: fit-content;"> <div style="background-color: #e6f2ff; padding: 2px; margin-bottom: 2px;">SVC</div> <div style="border: 1px dashed black; padding: 2px;">SVC(kernel='linear', random_state=0)</div> </div>	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> <div style="background-color: #e6f2ff; padding: 2px; margin-bottom: 2px;">AdaBoostClassifier</div> <div style="border: 1px dashed black; padding: 2px;">AdaBoostClassifier(n_estimators=9)</div> </div>

Gambar 9. Fitting SVM Classifier dan AdaBoostClassifier

Pada gambar 9 diatas, menunjukkan *library* yang di *import* untuk masing-masing model. Model SVM meng-*import* SVC untuk proses *fitting*. Sementara untuk model *AdaBoost* menggunakan *AdaBoostClassifier* dalam proses *fitting*. Setiap model akan melakukan *fitting* terhadap data latih (X_{train}) dan data latih prediksi (y_{train}).

2.5.1.3 Prediksi Data Uji

Proses berikutnya adalah melakukan prediksi data uji. Masing-masing model yang telah terbentuk akan diuji hasil prediksinya. Proses ini melibatkan *library predict*. *Library* ini akan melakukan prediksi terhadap variabel X_{test} pada masing-masing model.

```

y_prediksi_svm = klasifikasi_svm.predict(X_test_svm)

y_prediksi_svm

array([[0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1])

y_prediksi_adaboost = klasifikasi_adaboost.predict(X_test_adaboost)

y_prediksi_adaboost

array([[0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])

```

Gambar 10. Prediksi Data Uji Model SVM dan AdaBoost

Pada gambar 10 diatas memperlihatkan hasil prediksi masing-masing model. Nilai yang dihasilkan dalam bentuk *array* numerik 0 dan 1.

2.5.1.4 Melakukan Perbandingan $y_{prediksi_svm}$ dengan y_{test}

Proses ini menggunakan *confusion metrics*. Proses ini akan menghasilkan nilai *true positive* (TP), *true negative* (TN), *false positive* (FP) serta *false negative* (FN). Hasil dari proses ini adalah seperti pada gambar 11 berikut ini.

```

from sklearn.metrics import confusion_matrix

cm_svm = confusion_matrix(y_test, y_prediksi_svm)
cm_svm

array([[112, 19],
       [ 42, 25]])

cm_adaboost = confusion_matrix(y_test, y_prediksi_adaboost)
cm_adaboost

array([[126,  5],
       [ 48, 19]])

```

Gambar 11. Confusion Metrics Model SVM dan AdaBoost

Gambar 11 diatas menunjukkan nilai *confusion metrics* untuk masing-masing model. *Confusion metrics* yang dihasilkan dalam format *array*. Proses ini menggunakan *library confusion metrics* dengan melakukan perbandingan antara data uji (y_{test}) dan data prediksi masing-masing model.

Secara lengkap, nilai pada gambar 11 diatas dapat ditunjukkan seperti tabel berikut ini.

Tabel 4. Tabel Confusion Metrics

Predicted Values	Actual Values	
	Positive (1)	Negative (0)
	Positive (1)	TP

	Negative (0)	FN	TN
--	--------------	----	----

2.6 Pengujian Model

Teknik yang digunakan pada pengujian ini adalah dengan menggunakan *library accuracy_score*. Proses ini dilakukan dengan menggunakan *library accuracy_score*. Hasil dari proses ini adalah seperti yang terlihat pada gambar 12 dibawah ini.

```
from sklearn.metrics import accuracy_score

print(accuracy_score(y_test, y_prediksi_svm))    print(accuracy_score(y_test, y_prediksi_adaboost))
0.6919191919191919                               0.7323232323232324
```

Gambar 12. Hasil Pengujian Model SVM dan AdaBoost

Pada gambar 12 diatas memperlihatkan nilai pengujian untuk masing-masing model. Proses ini membandingkan data uji (*y_test*) dengan data prediksi masing-masing model.

2.7 Evaluasi Model

Proses terakhir yang dilakukan adalah dengan melakukan evaluasi model. Proses ini menggunakan *performance metrics*. Berikut ini adalah formula yang digunakan untuk menghasilkan setiap nilai pada *performance metrics*.

$$\textit{precision} = \frac{TP}{TP+FP} \dots\dots\dots (1)$$

$$\textit{recall} = \frac{TP}{TP+FN} \dots\dots\dots (2)$$

$$\textit{f1-score} = \frac{2 \times \textit{recall} \times \textit{precision}}{\textit{recall} + \textit{precision}} \dots\dots\dots (3)$$

$$\textit{accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \dots\dots\dots (4)$$

Teknik ini dilakukan setelah diperoleh nilai *confusion metrics* sebelumnya pada masing – masing model, berikut ini gambar 13, yaitu nilai dari *performance metrics* untuk masing-masing model.

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_prediksi_svm))    print(classification_report(y_test, y_prediksi_adaboost))
```

	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.73	0.85	0.79	131	0	0.72	0.96	0.83	131
1	0.57	0.37	0.45	67	1	0.79	0.28	0.42	67
accuracy			0.69	198	accuracy			0.73	198
macro avg	0.65	0.61	0.62	198	macro avg	0.76	0.62	0.62	198
weighted avg	0.67	0.69	0.67	198	weighted avg	0.75	0.73	0.69	198

Gambar 13. Hasil Evaluasi Model SVM dan AdaBoost

Pada gambar 13 diatas, menunjukkan nilai *peformance metrics* yang dihasilkan setiap model. Nilai ini terdiri dari *precision*, *recall*, *f1-score* serta *accuracy*.

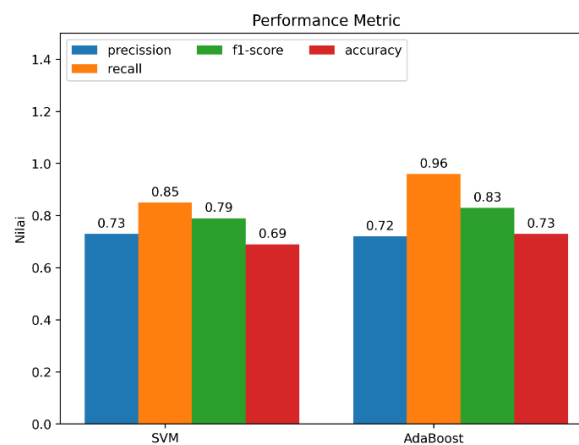
3. HASIL DAN PEMBAHASAN

Model yang telah dihasilkan memiliki informasi data seperti tabel 5 berikut;

Tabel 5. Perbandingan Model SVM dan AdaBoost

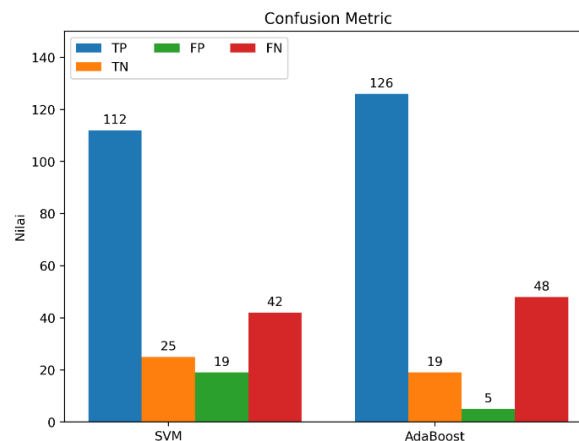
Model	Performance Metric				Confusion Metric			
	precision	recall	f1-score	accuracy	TP	TN	FP	FN
SVM	0,73	0,85	0,79	0,69	112	25	19	42
Adaboost	0,72	0,96	0,83	0,73	126	19	5	48

Berdasarkan tabel 5 diatas, tampak bahwa nilai *precision* atau presisi dari model SVM lebih besar dibandingkan dengan AdaBoost, hal ini dikarenakan jumlah TP dan TN model SVM lebih besar dibandingkan dengan AdaBoost. Berikut ini adalah gambar 14, grafik perbandingan *performance metric* antara kedua model yang dibangun.



Gambar 14. Grafik Perbandingan Performance Metric Model SVM dan AdaBoost

Pada gambar 14 diatas terlihat perbandingan kedua model perdasarkan nilai *performance metric* yang dihasilkan. Akurasi model AdaBoost 0,73 (73%), nilai yang dihasilkan lebih besar dibandingkan dengan SVM yaitu 0,69 (69%). Model AdaBoost tidak hanya menghasilkan nilai akurasi yang tinggi, namun semua nilai yang dihasilkan lebih tinggi dibandingkan dengan model SVM. Masing-masing nilai kedua model juga memiliki selisih yang cukup tinggi, yaitu 0,01 untuk nilai *precision*, 0,11 untuk *recall*, 0,04 untuk *f1-score* dan 0,04 untuk *accuracy*. Berikut ini adalah gambar 15, yaitu perbandingan *confusion metric* kedua model.



Gambar 15. Perbandingan Confusion Metric Model SVM dan AdaBoost

Pada gambar 15 diatas, terlihat bahwa nilai TP model *AdaBoost* yaitu 126. Kondisi ini menunjukkan bahwa model *AdaBoost* mampu memprediksi 126 nilai positif dari data uji yang diberikan. Sementara model SVM hanya memprediksi nilai positif sebanyak 112.

Jumlah dataset juga mempengaruhi performa pembentukan kedua model tersebut. Data yang terkumpul belum cukup banyak sehingga *performance metrics* menghasilkan nilai dibawah standar yaitu 0,85. Tidak hanya jumlah dataset yang mempengaruhi, namun *preprocessing data* juga sangat berpengaruh. Data numerik hanya terbagi menjadi 2 kategori yaitu 0 dan 1. Kedua kategori ini dirasa belum cukup untuk menghasilkan performa model yang maksimal dan spesifik. Nilai *float* lebih baik digunakan dalam variasi yang banyak, sehingga model bisa melakukan proses latih (*training*) lebih spesifik.

4. KESIMPULAN

Penelitian ini memberikan kesimpulan sebagai berikut.

1. Model algoritma *AdaBoost* memiliki nilai akurasi yang lebih baik dari pada SVM, dengan selisih 0,04.
2. Variabel independent yang digunakan belum cukup baik relasinya terhadap variabel dependent. Nilai korelasi masih jauh dibawah standar yaitu 0,5 – 0,7. Sementara pada penelitian ini hanya 2 variabel yang memiliki nilai diatas 0,2.
3. Komposisi nilai numerik belum cukup baik untuk dilakukan analisa. Komposisi sebaiknya dalam format *float* atau desimal dan dalam variasi yang cukup banyak.
4. Model yang dihasilkan belum cukup baik untuk melakukan prediksi, karena nilai akurasi yang dihasilkan masih dibawah 80%.

5. SARAN

Penelitian selanjutnya sebaiknya menggunakan dataset yang lebih banyak, disarankan lebih dari 10.000 data. Proses imputasi data, sebaiknya dilakukan dengan menggunakan beberapa teknik, tujuannya untuk memberikan nilai beda dari penelitian ini. Teknik rata-rata (*mean*) baik, namun ada teknik lain yang juga lebih baik, agar model yang dihasilkan nantinya akan memiliki nilai akurasi diatas 80%.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tim Redaksi Jurnal Teknika Politeknik Negeri Sriwijaya yang telah memberi kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

DAFTAR PUSTAKA

- [1] B. PT, "Akreditasi Perguruan Tinggi: Instrumen Pemantauan dan Evaluasi Peringkat Akreditasi Perguruan," in *Akreditasi Perguruan Tinggi: Instrumen Pemantauan dan Evaluasi Peringkat Akreditasi Perguruan*, 2022.
- [2] A. I. M. M. Martanto M, "Prediksi Tingkat Kelulusan Mahasiswa Menggunakan Machine Learning dengan Teknik Deep Learning," *Jurnal Informatika: Jurnal Pengembangan IT*, pp. 191-194, 2019.
- [3] I. R. Amirudin, "Implementasi Seleksi Fitur Klasifikasi Waktu Kelulusan Mahasiswa Menggunakan Correlation Matrix with Heatmap," *Jambura Journal of Electrical and Electronics Engineering*, pp. 169-174, 2022.
- [4] A. R. S. M. S. H. N. M. A. Suhaimi N M, "Review on Predicting Students' Graduation Time Using Machine Learning Algorithms," *International Journal of Modern Education and Computer Science*, vol. 11, no. 7, pp. 1-13, 2019.

-
- [5] N. S. F. U. N. F. Y. Utami T M, "Tide Prediction in Prigi Beach using Support Vector Regression (SVR) Method," *Scientific Journal of Informatics*, vol. 8, no. 2, pp. 194-201, 2021.
- [6] W. D. H. M. D. D. Wiyono S, "Comparative Study of KNN, SVM and Decision Tree Algorithm for Student's Performance Prediction," *International Journal of Computing Science and Applied Mathematics*, vol. 6, no. 2, p. 50, 2020.
- [7] H. T. A. I. Amrieh E.A, "Mining Educational Data to Predict Student's Academic Performance using Ensemble Methods," *International Journal of Database Theory and Application*, vol. 9, no. 8, pp. 119-136, 2016.
- [8] A. N. A. A. Bakri R, "Machine Learning Algorithms with Parameter Tuning to Predict Students Graduation on time: A Case Study in Higher Education," *Journal of Applied Science Engineering Technology and Education*, pp. 259-265, 2022.
- [9] U. D. Kesumawati A, "Predicting Patterns of Student Graduation Rates Using Naive Bayes Classification and Support Vector Machine," in *AIP Conference Proceedings*, 2018.
- [10] S. S. Z. M. Riansyah R, "Improved Accuracy In Data Mining Decission Tree Classification Using Adaptive Boosting (Adaboost)," *Jurnal dan Penelitian Teknik Informatika*, pp. 617-622, 2023.
- [11] S. S. Z. M. Rismayati M, "Ensemble Implementation for Predicting student Graduation with Classification Algorithm," *International Journal of Engineering and Computer Science Applications*, vol. 1, no. 1, pp. 35-42, 2022.
- [12] M. G. Wanjau S K, "Improving Student Enrollment Prediction Using Ensemble Classifiers," *International Journal of Computer Applications Technology and Research*, vol. 7, no. 3, pp. 122-128, 2018.
- [13] D. K. S. T. C. Teoh C, "Ensemble Learning Techniques for Predicting Student Performance on Video Based Learning," *International Journal of Information and Education Technology*, vol. 12, no. 8, pp. 741-745, 2022.
- [14] W. M. A. D. P. S. Setiyani L, "Analisis Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Data Mining Naive Bayes," *Factor Exacta*, vol. 13, no. 1, pp. 35-43, 2020.
- [15] S. P. P. J. Sagun M, "Modelling Student Dropout using AdaBoost and Survival Analysis," *Philippine Engineering Journal*, vol. 42, no. 2, pp. 75-98, 2021.