

Perbandingan Kinerja Metode Naive Bayes dan Support Vector Machine dalam Analisis Soal

Hidayatunnisa¹, Kusrini², Kusnawi³

^{1,2,3}Teknik Informatika, Universitas Amikom Yogyakarta

¹hidayatunnisa@students.amikom.ac.id, ²kusrini@amikom.ac.id, ³khusnawi@amikom.ac.id

Abstract

In conducting item analysis the process of collecting, summarizing, and using information from student answers is carried out to make decisions about each assessment. The purpose of the assessment is to improve student learning outcomes and provide information to students about their strengths and weaknesses in certain subjects that have been studied. Therefore, this study will discuss the comparative analysis of the performance levels of the Naïve Bayes classification algorithm and the SVM algorithm. Data mining methods for classification can be used to help increase speed and accuracy in analyzing item items so that the types of questions that are accepted, revised, and rejected will be obtained. Comparison of the performance of the Naive Bayes and SVM algorithms aims to measure the level of accuracy and length of processing time (execution time) of each algorithm to get the best algorithm to be applied in assisting the process of analyzing the items. The data used in the study were 55 with the results of students' answers to biology questions using training data and testing data of 80:20. By using the Python programming language tools, and after testing using cross validation so that the average yield of the Support Vector Machine algorithm is 97% while the Naïve Bayes algorithm is 95%, so it is explained that the performance of the Support Vector Machine algorithm is better than the Naïve Bayes, and it can be concluded that the SVM algorithm with K-fold Cross Validation of 10 or 10 times the trial stage is superior to Naïve Bayes.

Keywords: naïve bayes, SVM, cross validation, classifications.

Abstrak

Dalam melakukan analisis butir soal yang dilakukan proses pengumpulan, peringkasan, dan penggunaan informasi dari jawaban siswa untuk membuat keputusan tentang setiap penilaian. Tujuan dari penilaian adalah untuk meningkatkan hasil belajar siswa serta memberikan informasi kepada siswa tentang kelebihan dan kekurangannya dalam mata pelajaran tertentu yang telah dipelajari. Oleh karena itu, dalam penelitian ini akan membahas tentang analisis perbandingan tingkat kinerja algoritma klasifikasi *Naïve Bayes* dan algoritma SVM. Metode data mining untuk klasifikasi dapat digunakan untuk membantu meningkatkan kecepatan dan ketepatan dalam menganalisis butir soal sehingga akan didapatkan jenis soal yang diterima, direvisi, dan ditolak. Perbandingan kinerja algoritma *Naïve Bayes* dan SVM bertujuan untuk mengukur tingkat akurasi dan lama waktu proses (execution time) dari masing-masing algoritma untuk mendapatkan algoritma terbaik yang akan diterapkan dalam membantu proses analisis butir soal. Data yang digunakan dalam penelitian sebanyak 55 dengan hasil jawaban siswa pada soal biologi dengan menggunakan data training dan data testing 80:20. Dengan menggunakan alat bantu bahasa pemrograman *python*, dan setelah dilakukan pengujian menggunakan *cross validation* sehingga rata-rata yang dihasilkan dari algoritma *Support Vector Machine* adalah 97% sedangkan algoritma *Naïve Bayes* adalah 95%, sehingga dipaparkan bahwa kinerja algoritma *Support Vector Machine* lebih baik dibandingkan dengan algoritma *Naïve Bayes*. dapat disimpulkan bahwa algoritma SVM dengan *K-fold Cross Validation* sebesar 10 atau 10 kali tahap percobaan lebih unggul dibandingkan dengan *Naïve Bayes*.

Kata kunci: *naïve bayes, SVM, cross validation, klasifikasi.*

©This work is licensed under a Creative Commons Attribution - ShareAlike 4.0 International License

1. Pendahuluan

Analisis butir soal adalah pengkajian pertanyaan-pertanyaan tes agar diperoleh perangkat pertanyaan yang memiliki kualitas yang memadai. Meningkatnya kebutuhan penilaian berbasis sekolah, guru tentunya harus memiliki pengetahuan dalam menganalisis butir soal agar bisa meningkatkan kualitas soal yang ada [1]. Dalam melakukan analisis butir soal yang harus dilakukan proses pengumpulan, peringkasan, dan penggunaan informasi dari jawaban siswa untuk membuat keputusan tentang setiap penilaian [2].

Tujuan dari penilaian adalah untuk meningkatkan hasil belajar siswa serta memberikan informasi kepada siswa

tentang kelebihan dan kekurangannya dalam mata pelajaran tertentu yang telah di pelajari.

Hal ini mengkhawatirkan karena soal pilihan ganda yang dikembangkan sendiri dan yang diambil dari internet dapat memberikan kualitas yang buruk pada soal tersebut dengan itu perlu dilakukannya analisis butir soal (Nitko) [3].

Dengan melakukan analisis butir soal juga, guru tentunya dapat melihat dan menilai bahwa soal yang diberikan merupakan soal berkualitas atau tidak, serta guru dapat menentukan apakah ini waktu terbaik untuk beralih topik ke topik lain atau topik yang diajarkan memerlukan elaborasi dan pengajaran lebih lanjut [3]. Guru dapat melihat kekuatan siswa sekaligus

kelemahan siswa terhadap topik tersebut. Oleh karena itu, seorang guru dengan mudah mengatasi kesulitan siswa [4].

Peraturan Pemerintah Nomor 19 Tahun 2005 tentang Standar Nasional Pendidikan, Pasal 64 ayat (1) menyatakan bahwa pentingnya melakukan analisis butir soal guna mendapatkan kualitas butir soal yang baik. Penilaian hasil belajar oleh guru dilakukan secara berkesinambungan untuk memantau proses, kemajuan, dan perbaikan hasil dalam bentuk ulangan harian, ulangan tengah semester, ulangan akhir semester, dan ulangan kenaikan kelas. Salah satu tugas utama guru sebagai tenaga profesional adalah menilai dan mengevaluasi peserta didik. Soal esai, soal pilihan ganda, tes lisan, dan tes praktek sering digunakan sebagai alat penilaian di sekolah.

Keuntungan memiliki soal pilihan ganda adalah memberi guru manfaat untuk mencakup bidang konten yang luas dan mengidentifikasi siswa yang memiliki kualifikasi tinggi maupun yang memiliki kualifikasi rendah. Namun dalam membuat soal pilhan ganda untuk siswa tidaklah mudah, karena dalam membuat soal pilihan ganda juga kian menyita waktu guru (Tarrant), selain itu, studi oleh (Tarrant) [2], menemukan bahwa tidak banyak guru yang memiliki keterampilan dan pengetahuan dalam mengembangkan soal pilihan ganda. Biasanya, guru mengembangkan dari keterampilan sendiri dalam menyusun soal, maupun mereka hanya mengambil bank soal-soal pilihan ganda yang ada di buku referensi atau *browsing* di internet.

Menurut kriteria suatu soal harus sesuai dengan indicator soal, pokok soal harus dirumuskan secara jelas dan tegas, pokok soal jangan memberikan kearah yang benar, dan penilaian kualitas soal tersebut dengan mencari tingkat kesukaran dan daya pembeda. Butri soal yang baik atau berkualitas adalah soal yang tidak terlalu mudah dan terlalu sukar, sehingga dapat diartikan sebah soal yang harus memiliki tingkat kesukaran yang sedang atau cukup.

Dalam penelitian ini, peneliti akan menjelajahi kinerja dari algoritma *Naive Bayes* dan algoritma *Support Vector Machine* (SVM) dengan penggunaan dataset hasil dari ulangan harian siswa kelas ix pada mata pelajaran biologi dalam melakukan analisis butir soal. Peneliti akan menganalisis hubungan kinerja yang relatif unggul dari kedua algoritma tersebut berdasarkan tingkat kesukaran dan daya pembeda.

Banyak metode yang bisa dilakukan untuk menganalisis kualitas butir soal. Salah satu metode yang digunakan adalah teknik data *mining*. Tapi tidak semua algoritma data *mining* memiliki kinerja yang baik dalam mengklasifikasi analisis butir soal. Oleh karena itu, dalam penelitian ini akan membahas tentang analisis perbandingan tingkat kinerja algoritma klasifikasi *Naive Bayes Classifier* dan algoritma *Support Vector Machine* (SVM).

2. Metode Penelitian

Dalam penelitian ini, jenis penelitian yang digunakan adalah penelitian yang bersifat kuantitatif dengan penelaahan butir soal yang berdasarkan data butir soal yang bersangkutan. Penelitian ini memberikan gambaran yang digunakan untuk mengetahui tingkat kinerja dala penggunaan algoritma *Naive Bayes* dan *Support Vector Machine* dalam menganalisis kualitas butir soal dan variabel manakah yang berpengaruh kinerja model terhadap performa kedua algoritma tersebut.

Dalam penelitian ini, variabel yang akan digunakan berjumlah dua variabel yaitu tingkat kesukaran dan daya pembeda. Data yang akan diujikan didapat dari data hasil jawaban ulangan harian siswa kelas ix pada mata pelajaran biologi tahun ajaran 2022/2023. Data yang digunakan sebanyak 30 butir soal dengan jumlah siswa sebanyak 25 siswa.

Dalam perhitungan tingkat kesukaran dan daya pembeda memiliki konsep perhitungan dari jawaban hasil ulangan hari siswa dengan rumus yang telah ada, yang kemudian di konversikan kedalam format *.xlxs*. Hasil dari pengumpulan dan perhitungan dataset tingkat kesukaran dan daya pembeda, kemudian data kemudian dilakukan konversi kedua data menjadi data numerik dengan melakukan penghitungan jumlah tingkat kesukaran dan data pembeda berdasarkan algoritma *Naive Bayes* dan *Support Vector Machine*. Data yang telah dikonversi kedalam angka tersebut kemudian dimasukkan kedalam tabel dataset seperti terlihat pada tabel 1 dibawah, dengan data variabel atau kolom yang terdiri dari nomor, tingkat kesukaran dan daya pembeda.

Tabel 1. Dataset Tingkat Kesukaran dan Daya Pembeda

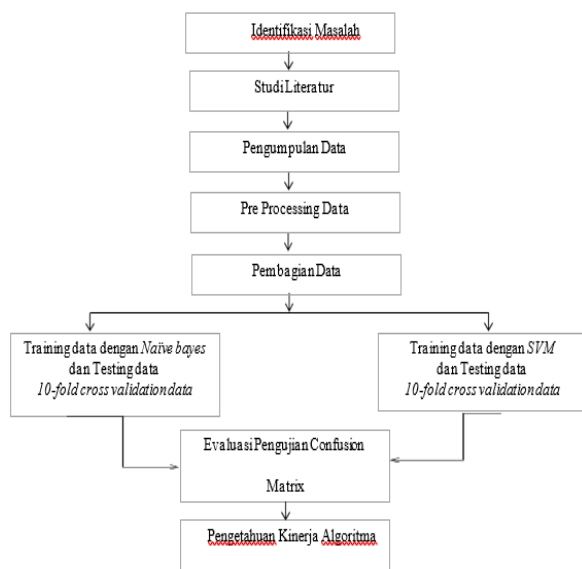
Tanggal	Kesukaran	TK	Pembeda	DP	Klasifikasi
1	0.85	0	0.3	1	0
2	0.4	1	0.6	1	1
3	0.65	1	0.3	1	1
4	0.6	1	0.2	0	0
5	0.2	0	0.5	1	0
::	::	::	::	::	::
::	::	::	::	::	::
6	0.8	0	0.1	0	0
7	0.6	1	0.6	1	1
8	0.85	0	0.1	0	0
9	0.5	1	0.6	1	1
10	0.5	1	-0.1	2	1

Dari Tabel 1 diatas data berisi hasil perhitungan tingkat kesukaran dan daya pembeda dan klasifikasi tiap variabel dan klasifikasi terhadap 2 variabel tersebut. Tingkat kesukaran soal adalah peluang untuk menjawab benar suatu soal pada tingkat kemampuan tertentu yang biasanya dinyatakan dalam bentuk indeks. Indeks tingkat kesukaran ini pada umumnya

dinyatakan dalam bentuk proporsi yang besarnya berkisar 0,00-1,00 [7], sedangkan daya pembeda adalah suatu tes yang bertujuan untuk mengetahui perbedaan kemampuan peserta didik. Semakin tinggi indeks yang dimiliki oleh butir soal, maka semakin baik butir soal karena memiliki daya untuk membedakan kemampuan peserta didik yang pandai dan kurang pandai [8], dimana indeks daya pembeda berkisar dari -1,00 sampai dengan +1. Dalam melakukan analisis butir soal tujuannya adalah dengan melihat kelayakan soal tersebut, dimana klasifikasi analisis soal tersebut dikonversikan kedalam format numerik, dimana yang bernilai 0 adalah soal diterima, bernilai 1 adalah soal direvisi, dan bernilai 2 soal ditolak.

2.1 Alur Penelitian

Tahapan penelitian yang dilakukan dalam pengujian data pada perbandingan kinerja metode *Naïve Bayes* dan metode *Support Vector Machine* pada analisis kualitas butir soal dapat dilihat pada gambar 1 dibawah ini :



Gambar 1. Diagram Alur Penelitian

Alur penelitian merupakan langkah-langkah yang dilakukan dalam melakukan penelitian. Pada Gambar 1 diatas penelitian dimulai dari melakukan identifikasi masalah, mencari studi literatur sebagai acuan penelitian, pengumpulan data, pembagian data, sampai dengan pengukuran tiap algoritma yang digunakan, sehingga hasil dari penelitian ini mengukur kinerja dari algoritma *Naïve Bayes* dan algoritma *Support Vector Machine*.

Dengan mengumpulkan data, yaitu hasil dari ulangan siswa kelas SMP yang dimana terdiri dari tingkat kesukaran dan daya pembeda, dari data yang didapatkan akan diolah dengan algoritma yang dipilih, sehingga dapat melihat perbandingan kinerja masing-masing algoritma dengan mempertimbangkan hasil akurasi, presisi, dan recall dalam setiap pengujian algoritma dengan menggunakan *k-fold cross validation* sebanyak 10.

2.2 Naïve Bayes

Algoritma *Naïve Bayes* merupakan metode klasifikasi yang sederhana namun didasarkan pada probabilitas [9]. Prinsip dasar dari algoritma *Naïve Bayes* adalah menghitung himpunan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari data yang diberikan. Algoritma ini menggunakan teorema Bayes dalam membangun model klasifikasi. Oleh karena itu, untuk memprediksi kelas suatu objek, kita menentukan probabilitas keanggotaannya. Algoritma *Naïve Bayes* juga mampu mengatasi dataset dengan banyak atribut [10]. Persamaan (1) dari *Naïve Bayes* dapat dinyatakan sebagai berikut:

$$P(H|E) = \frac{p(E|H)XP(H)}{P(E)} \quad (1)$$

Keterangan :

$P(H|E)$: Probabilitas akhir bersyarat (*conditional probability*) suatu hipotesis H terjadi jika diberikan bukti (*evidence*) E terjadi.

$P(E|H)$: Probabilitas sebuah bukti E terjadi akan mempengaruhi hipotesis H

$P(H)$: Probabilitas awal (priori) hipotesis H terjadi tanpa memandang bukti apapun.

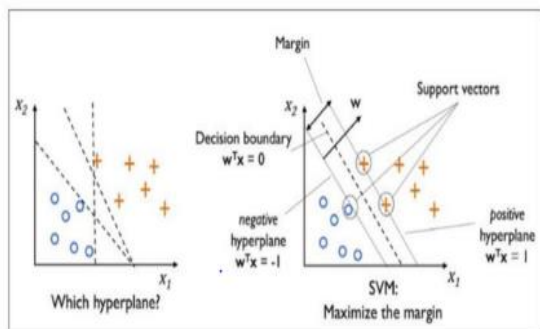
$P(E)$: Probabilitas awal (priori) bukti E terjadi tanpa memandang hipotesis/bukti yang lain.

Pada persamaan 1 diatas X_scaled merupakan variabel input sedangkan X merupakan matrix dataset yang akan dinormalisasi, X_{min} merupakan nilai minimum dari setiap kolom dalam matrix dataset, dan X_{max} merupakan nilai maksimum dari setiap kolom dalam matrix dataset [11].

2.3 Support Vector Machine (SVM)

SVM merupakan salah satu metode klasifikasi yang telah banyak dikembangkan dan diaplikasikan saat ini. Metode ini berasal dari teori pembelajaran statistika yang menjanjikan dan memberikan hasil yang lebih baik jika dibandingkan dengan metode lainnya. SVM adalah sistem pembelajaran yang menggunakan fungsi linier dalam ruang fitur berdimensi tinggi. Metode ini dilatih menggunakan algoritma pembelajaran yang didasarkan pada teori optimisasi dan menerapkan bias pembelajaran yang berasal dari teori pembelajaran statistika [11]. Vapnik, Guyon, dan Boser adalah pengembang utama SVM. Metode ini pertama kali diperkenalkan sekitar tahun 1992 dalam Workshop Tahunan Teori Pembelajaran Komputasi[12]. SVM merupakan mesin pembelajaran yang mencari hyperplane terbaik untuk memisahkan dua kelas dalam ruang input. *Hyperplane* pemisah terbaik dihasilkan dengan mengoptimalkan margin, yaitu jarak antara hyperplane dan titik-titik terdekat dari masing-masing kelas. Posisi *hyperplane* yang optimal tercapai saat berada di tengah, membagi antara kelas negatif dan kelas positif [13]. Prinsip dasar SVM adalah sebagai classifier linear, tetapi kemudian dikembangkan untuk

menangani masalah *non-linear* dengan menggunakan trik kernel dalam ruang fitur berdimensi tinggi [14].



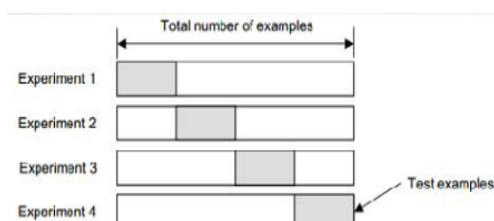
Gambar 2. *Hyperplane* yang Memisahkan 2 Kelas Positif (+1) dan Kelas Negatif (-1)

Hyperplane yang ditemukan oleh SVM diilustrasikan seperti pada Gambar 2. Posisinya berada di tengah tengah antara kedua kelas, artinya jarak antara *hyperplane* dan objek data berbeda dengan jarak kelas yang berdekatan (terluar) yang ditandai dengan bulatan kosong dan positif. Dalam SVM, objek data terluar yang paling dekat dengan *hyperplane* disebut support vector. Objek yang disebut vektor pendukung adalah yang paling sulit untuk diklasifikasi karena posisinya yang hampir Tumpang tindih dengan kelas lain.

Mengingat sifatnya yang kritis, hanya vektor dukungan ini yang diperhitungkan untuk menemukan *hyperplane* paling optimal oleh SVM *Hyperplane* adalah garis pemisah terbaik antara dua kelas. Untuk mencari *hyperplane* dapat dilakukan dengan mencari margin *hyperplane* 11 dan mencari titik maksimalnya. Margin adalah jarak antara data terdekat antara dua kelas yang berbeda, yang disebut support vector. Garis solid pada Gambar 1-b menunjukkan *hyperplane* terbaik, karena terletak tepat di antara kedua kelas, sedangkan support vector diwakili oleh titik merah dan kuning di dalam lingkaran hitam. *Hyperplane* klasifikasi linier SVM dilambangkan: $f(x) : w \cdot x + b = 0$ Dari persamaan di atas, kita mendapatkan pertidaksamaan kelas +1 (negatif) $w \cdot x + b \leq -1$ Pertidaksamaan kelas -1: $w \cdot x + b \geq 1$.

2.4 K-Fold Cross Validation

K-fold cross validation adalah Teknik untuk mengestimasi performansi dari model pelatihan yang telah dibangun [15]. Metode ini membagi data *training* dan *testing* sebanyak k bagian data. Fungsi dari *k-fold cross validation* adalah agar tidak ada terjadi *overlapping* pada data *testing*.



Gambar 3. Ilustrasi K-Fold Cross Validation

Cross validation juga memungkinkan kami untuk memperoleh estimasi yang lebih stabil tentang kinerja model. Dengan melakukan beberapa iterasi *cross validation*, peneliti dapat menghitung rata-rata dan deviasi standar dari metrik evaluasi yang dihasilkan. Ini memberikan kami pemahaman yang lebih baik tentang sejauh mana model kami dapat digeneralisasi dan mengukur tingkat kepastian dari hasil evaluasi yang kami peroleh. Secara keseluruhan, penerapan *cross validation* merupakan pendekatan yang penting dalam pengembangan dan evaluasi model analisis kualitas butir soal. Melalui penggunaan metode ini, dapat menghasilkan hasil yang lebih objektif dan reliabel, serta memperoleh pemahaman yang lebih baik tentang kinerja algoritma secara keseluruhan.

2.5 Akurasi, Presesi, dan Recall

Akurasi dapat didefinisikan sebagai tingkat kedekatan antara lain prediksi dengan nilai actual [16]. Presesi menunjukkan tingkat ketepatan atau ketelitian dalam pengklasifikasian. Sedangkan *recall* berfungsi untuk mengukur proporsi positif actual yang benar diidentifikasi. Untuk mengukur akurasi, presesi, dan *recall* biasanya menggunakan *confusion matrix*. *Confusion matrix* adalah sebuah metode untuk mengevaluasi kinerja model klasifikasi [17]. Ini adalah tabel persegi yang digunakan untuk menggambarkan performa prediksi model pada set data uji, di mana setiap kolom dari matriks mewakili prediksi dari kelas tertentu, dan setiap baris mewakili instansi aktual dari kelas tersebut.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<p>TP (True Positive)</p>	<p>FP (False Positive) <small>Type I Error</small></p>
	0 (Negative)	<p>FN (False Negative) <small>Type II Error</small></p>	<p>TN (True Negative)</p>

Gambar 4. Bentuk *Confusion Matrix* dari Dua Kelas

Pada Tabel 1 *True Positive* (TP) Ini terjadi ketika model memprediksi *instance* sebagai positif dan benar-benar positif. Dakonteks klasifikasi dua kelas, ini berarti *instance* diklasifikasikan dengan benar sebagai kelas positif. *True Negative* (TN) Ini terjadi ketika model memprediksi *instance* sebagai negatif dan benar-benar negatif. Dalam konekes klasifikasi dua kelas, ini berarti *instance* diklasifikasikan dengan benar sebagai kelas negatif. *False Positive* (FP) Ini terjadi ketika model memprediksi *instance* sebagai positif tetapi sebenarnya *negative*. Berikut formulasi untuk menghitung akurasi, presesi, dan recall pada

pembentukan model klasifikasi ditunjukkan pada Persamaan (2), Persamaan (3), dan Persamaan (4) [18].

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2)$$

$$\text{Presesi} = \frac{TP}{TP+FP} \times 100\% \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

3. Hasil dan Pembahasan

3.1 Pengolahan Data

Pengolahan data yang dilakukan melalui eksperimen terhadap hasil dari jawaban siswa, hingga untuk melihat perbandingan mana yang lebih baik dalam mengukur kinerja algoritma *Naïve Bayes* dan algoritma *Support Vector Machine*. Dataset yang telah dikumpulkan untuk penelitian ini terdiri dari dua variabel yaitu tingkat kesukaran dan daya pembeda. Dataset tersebut diunggah ke *google drive* untuk bisa diakses oleh *google collab*. Pada pengujian penelitian membandingkan algoritma *Support Vector Machine* (SVM) dan *Naïve Bayes* dengan dataset yang berjumlah 55 data hasil ulangan harian siswa, dengan pembagian data *training* sebanyak 80 % dan data *testing* sebanyak 20% dan menggunakan *10-fold cross validation*. Tujuan dari penelitian ini adalah untuk mengukur tingkat kinerja dari masing-masing algoritma dengan mempertimbangkan akurasi, presesi, dan *recall* dari setiap pengujian algoritma.

```
[2] import warnings
     warnings.filterwarnings('ignore')

[4] data = '/content/data2.csv'
     df = pd.read_csv(data)

[ ] df.shape

(30, 6)
```

Gambar 5. Algoritma Import Dataset

Pada Gambar 5 tersebut menjelaskan bahwa `df=pd.read_csv(r"/content/data2.csv")` ini bertujuan untuk membaca file dataset yang bernama "data2.csv" dan mengimpor data ke dalam program *Python* menggunakan *library*. Langkah selanjutnya yang dilakukan ialah dengan melakukan *runtime* untuk menampilkan dataset yang akan digunakan

No	Kesukaran	TK	Pembeda	DP	Klasifikasi
0	1	0.85	0	0.3	1
1	2	0.40	1	0.6	1
2	3	0.65	1	0.3	1
3	4	0.60	1	0.2	0
4	5	0.20	0	0.5	1

Gambar 6. Dataset Naïve Bayes

No	Kesukaran	TK	Pembeda	DP	Klasifikasi
17	18	0.57	1	0.3	1
2	3	0.65	1	0.3	1
16	17	1.10	2	1.0	2
11	12	0.55	1	0.1	0
15	15	0.55	1	0.1	0

Gambar 7. Dataset SVM

Pada Gambar 6 dan 7 tampilan dari dataset. Dataset yang digunakan dalam penelitian ini terdiri dari 55 data yang dikumpulkan untuk melakukan perbandingan kinerja metode *Naive Bayes* dan SVM dalam analisis butir soal. Setiap data dalam dataset ini memiliki atribut "tingkat kesukaran" yang menggambarkan sejauh mana suatu butir soal dianggap sulit. Skala yang digunakan untuk mengukur tingkat kesulitan didasarkan pada penilaian ahli. Selain itu, dataset juga mencakup atribut "daya pembeda" yang mengukur sejauh mana suatu butir soal dapat membedakan antara siswa yang memiliki pemahaman yang baik tentang materi dan siswa yang tidak, dan juga "klasifikasi" untuk mengukur kinerja soal apakah diterima, direvisi, dan ditolak, dengan keterangan bahwa diterima bernilai 1, direvisi bernilai 0, dan ditolak bernilai 2. Metode yang digunakan untuk menentukan daya pembeda adalah dengan menguji validitas butir soal.

Dataset ini disajikan dalam format file CSV dengan kolom-kolom tambahan yang mencakup informasi lain terkait butir soal. Sebagai contoh, beberapa baris data diberikan untuk memberikan gambaran tentang jenis data yang ada. Dataset ini digunakan untuk melatih dan menguji model klasifikasi menggunakan metode *Naive Bayes* dan SVM, dan kemudian digunakan untuk mengevaluasi kualitas kinerja dari kedua metode tersebut.

3.2 Pengujian

Pengujian Pada tahap ini dilakukan pengujian data dengan dataset yang telah diunggah ke *google drive*. kemudian diunggah ke *google collab* dilakukan *preprocessing* data supaya data siap untuk digunakan dan dimasukkan kedalam model.

```
df.info() # All features in integer form.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---          -
0   No               30 non-null    int64
1   Kesukaran       30 non-null    float64
2   TK              30 non-null    int64
3   Pembeda        30 non-null    float64
4   DP              30 non-null    int64
5   Klasifikasi     30 non-null    int64
dtypes: float64(2), int64(4)
memory usage: 1.5 KB
```

```
df.describe() # There may be outlier in glucose.
```

Gambar 8. Algoritma Preprocessing Data SVM

```
[ ] X_train.shape, X_test.shape
```

```
((24, 5), (6, 5))
```

```
[ ] cols = X_train.columns
```

```
[ ] from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

```
[ ] X_train = pd.DataFrame(X_train, columns=[cols])
```

```
[ ] X_test = pd.DataFrame(X_test, columns=[cols])
```

Gambar 9. Algoritma Preprocessing Data Naive Bayes

Setelah data dimuat, langkah berikutnya adalah membersihkan data dari nilai yang hilang, duplikat, atau tidak valid. Hal ini melibatkan penghapusan baris atau kolom yang memiliki nilai yang hilang, penggantian nilai yang hilang dengan nilai yang sesuai, dan penghapusan duplikat. Proses ini bertujuan untuk memastikan bahwa data yang digunakan adalah lengkap dan valid.

Selanjutnya, data dapat diubah atau diformat sesuai dengan kebutuhan analisis atau pemodelan yang akan dilakukan. Hal ini dapat meliputi transformasi data, seperti normalisasi atau standarisasi, encoding variabel kategorikal menjadi bentuk numerik, atau pengubahan skala data untuk menghindari bias. Selain itu, dalam proses preprocessing, fitur-fitur yang tidak relevan atau tidak diperlukan dapat dihapus untuk mengurangi dimensi data dan meningkatkan efisiensi analisis.

Dengan melakukan langkah-langkah preprocessing yang tepat, data dapat dipersiapkan dengan baik sebelum dilakukan analisis atau pemodelan. Hal ini membantu memastikan bahwa data yang digunakan adalah bersih, valid, dan dalam format yang sesuai untuk tujuan analisis yang diinginkan.

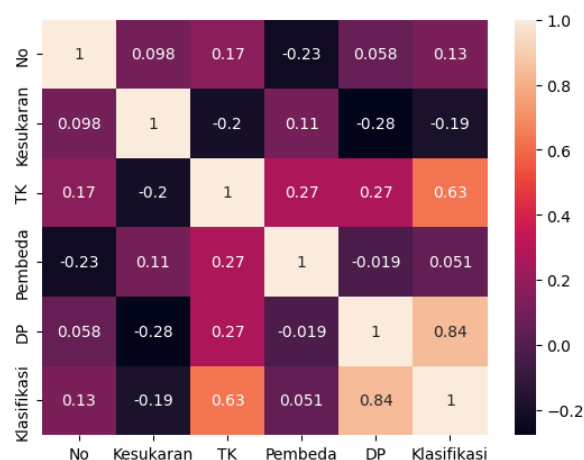
	No	Kesukaran	TK	Pembeda	DP
count	2.400000e+01	2.400000e+01	2.400000e+01	24.000000	2.400000e+01
mean	7.401487e-17	-8.326673e-17	-5.551115e-17	0.000000	-2.775558e-17
std	1.021508e+00	1.021508e+00	1.021508e+00	1.021508	1.021508e+00
min	-1.662351e+00	-2.208358e+00	-1.315499e+00	-1.846372	-1.260252e+00
25%	-8.487534e-01	-6.525677e-01	-1.315499e+00	-0.738549	-1.260252e+00
50%	2.511105e-02	-3.414096e-01	5.416763e-01	0.000000	4.200840e-01
75%	7.784425e-01	9.680475e-01	5.416763e-01	0.461593	4.200840e-01
max	1.833107e+00	2.459014e+00	2.398852e+00	2.584921	2.100420e+00

Gambar 10. Preprocessing Data Naive Bayes

	No	Kesukaran	TK	Pembeda	DP	Klasifikasi
count	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000
mean	15.466667	0.625667	0.766667	0.266667	0.800000	0.700000
std	8.803343	0.176766	0.504007	0.269525	0.664364	0.749713
min	1.000000	0.200000	0.000000	-0.200000	0.000000	0.000000
25%	8.250000	0.512500	0.250000	0.100000	0.000000	0.000000
50%	15.000000	0.600000	1.000000	0.300000	1.000000	1.000000
75%	22.750000	0.737500	1.000000	0.375000	1.000000	1.000000
max	30.000000	1.100000	2.000000	1.000000	2.000000	2.000000

Gambar 11. Preprocessing Data SVM

Pada Gambar 10 dan 11 dilakukan *preprocessing* data, dengan melakukan preprocessing data yang tepat, sehingga dapat memastikan bahwa data yang digunakan dalam analisis atau pembelajaran mesin memiliki kualitas yang baik, relevan, dan sesuai dengan kebutuhan model atau algoritma yang akan diterapkan oleh penelitian, yaitu penggunaan algoritma *Support Vector Machine* dan algoritma *Naive Bayes*.



Gambar 12. Heatmap data

Heatmap adalah teknik visualisasi grafis yang menggunakan warna untuk menunjukkan intensitas atau nilai dalam dataset. Dalam konteks ini, heatmap dapat digunakan untuk memvisualisasikan matriks korelasi antara variabel tingkat kesukaran dan daya pembeda. Dengan menggunakan heatmap, kita dapat melihat hubungan atau pola antara kedua variabel ini dengan cara yang lebih intuitif dan mudah dipahami. Misalnya, dalam heatmap, dapat mengamati pola warna yang berbeda untuk mengidentifikasi korelasi positif atau negatif antara tingkat kesukaran dan daya pembeda. Warna yang lebih terang dapat menunjukkan korelasi yang lebih kuat, sementara warna yang lebih gelap menunjukkan korelasi yang lebih lemah. Dengan menggunakan *heatmap*, dapat menggambarkan secara visual hubungan antara variabel-variabel ini, yang dapat memberikan wawasan yang berharga dalam analisis butir soal.

Dari data hasil pengujian algoritma *Support Vector Machine* (Kernel linear) dengan *Naïve Bayes* dengan menggunakan variabel tingkat kesukaran, dan daya pembeda dipaparkan bahwa tingkat performansi akurasi, presesi, *recall* untuk ujian pengujian pertama memiliki tingkat performansi sebesar 100%. Dalam penelitian ini, peneliti juga menggunakan *10-fold cross validation*, kedua algoritma ini menggunakan iterasi sebanyak 10 kali menggunakan *cross validation* untuk melihat lebih jelas lagi tingkat kinerja dari kedua algoritma yang digunakan, setelah dilakukan pengujian kedua menggunakan *cross validation* sehingga rata-rata yang dihasilkan dari algoritma *Support Vector Machine* adalah 97% sedangkan algoritma *Naïve Bayes* adalah 95%, sehingga dipaparkan bahwa kinerja algoritma *Support Vector Machine* lebih baik dibandingkan dengan algoritma *Naïve Bayes*.

Dengan demikian dalam melakukan pengujian *Naive Bayes* memberikan kinerja yang baik dalam penelitian di mana terdapat asumsi independensi antara fitur-fitur dalam dataset. *Naive Bayes* tetap memberikan hasil yang memuaskan dan efektif. SVM, di sisi lain, tidak mengasumsikan independensi fitur, tetapi dapat menjadi kurang efisien dan sulit diterapkan pada dataset dengan dimensi yang tinggi.

Selain itu, *Naive Bayes* memiliki keunggulan dalam mengatasi masalah *overfitting*. Metode ini cenderung memiliki kemampuan yang baik dalam melakukan generalisasi terhadap data baru, meskipun memiliki asumsi yang sederhana. SVM cenderung memiliki kecenderungan untuk *overfitting*, terutama ketika jumlah fitur meningkat atau ketika terdapat data yang tidak linier.

4. Kesimpulan

1. Penelitian dilakukan dengan melakukan perbandingan kinerja algoritma *Support Vector Machine* dan algoritma *Naïve Bayes*.

2. Data yang digunakan dalam penelitian ini adalah hasil ulangan harian siswa SMP kelas IX dengan cakupan data sebanyak 55 data dengan variabel tingkat kesukaran dan daya pembeda.

3. Tingkat kinerja yang diukur dengan akurasi, presesi, *recall*, dan *cross validation* sebanyak 10 kali, algoritma *Support Vector Machine* memiliki tingkat kinerja yang lebih besar nilainya dari pada algoritma *Naïve Bayes*.

Daftar Rujukan

- [1] Mania, Sitti, et al., 2020. Analisis butir soal ujian akhir sekolah. *Al asma: Journal of Islamic Education* 2(2), 274-284.
- [2] Mania, S., Fitriani, F., Majid, A. F., Ichiana, N. N., & Abrar, A. I. P., 2020. Analisis butir soal ujian akhir sekolah. *Al asma: Journal of Islamic Education*, 2(2), 274-284.
- [3] Dewi P, Elvia R, Elvisnawati E., 2021. Pengembangan Butir Soal Hots Untuk Menguji Kemampuan Berpikir Tingkat Tinggi Siswa Di Ma Negeri 2 Kota Bengkulu. *ALOTROP*. Aug 5(2),141-8.
- [4] Aiken Jr, Lewis R., 1976. Update on attitudes and other affective variables in learning mathematics. *Review of educational research* 46.2, 293-311.
- [5] Nalattissifa, Hiya, et al., 2021. Perbandingan Kinerja Algoritma Klasifikasi Naive Bayes, Support Vector Machine (SVM), dan Random Forest untuk Prediksi Ketidakhadiran di Tempat Kerja. *Jurnal Informatika Universitas Pamulang* 5.4, 578-584.
- [6] Fikri, Mujaddid Izzul, Trifebi Shina Sabrila, and Yufis Azhar., 2020. Perbandingan metode naïve bayes dan support vector machine pada analisis sentimen twitter. *SMATIKA Jurnal: STIKI Informatika Jurnal* 10(2), 71-76.
- [7] Y. Li., H. Cao., 2018. Prediction for Tourism Flow based Nurhalimah, Sri, et al., 2022. Hubungan Antara Validitas Item Dengan Daya Pembeda Dan Tingkat Kesukaran Soal Pilihan Ganda PAS. *Natural Science Education Research* 4.3, 249-257.
- [8] Wijaya, Tommy Tanu, et al., 2022. An Analysis of the Online Final Examination Items in the Mathematics Course Using the Rasch Measurement Model. *Jurnal Educative: Journal of Educational Studies* 7.2, 153-172.
- [9] Reddy, Eguturi Manjith Kumar, et al., 2022. Introduction to Naive Bayes and a Review on Its Subtypes with Applications. *Bayesian Reasoning and Gaussian Processes for Machine Learning Applications*, 1-14.
- [10] Reddy EM., Gurralla A., Hasitha VB., Kumar KV. 2022. Introduction to Naive Bayes and a Review on Its Subtypes with Applications. *Bayesian Reasoning and Gaussian Processes for Machine Learning Applications*. Apr , 19:1-4.
- [11] Sun, Zhen., João Santos., and Elsa Caetano. 2022. Vision and support vector machine-based train classification using weigh-in-motion data. *Journal of Bridge Engineering* 27(6), 06022001.
- [12] Sun, Z., Santos, J., & Caetano, E. 2022. Vision and support vector machine-based train classification using weigh-in-motion data. *Journal of Bridge Engineering*, 27(6), 06022001.
- [13] Sun Z., Santos J., Caetano E. 2022. Vision and support vector machine-based train classification using weigh-in-motion data. *Journal of Bridge Engineering*. 2022 Jun 1;27(6), 06022001.
- [14] Sonmez, Mesut Ersin, et al. 2022. Convolutional neural network-Support vector machine based approach for classification of cyanobacteria and chlorophyta microalgae groups." *Algal Research* 61, 102568.
- [15] Supeli, Muhammad Fudhail Ferio, and Setiaji Setiaji. 2023. Klasifikasi Sentimen Positif Dan Negatif Pada Aplikasi Vidio

-
- Dengan Algoritma K-Nearest Neighbor." *Indonesian Journal*[17] *Computer Science* 2.(1), 7-15.
- [16] Riandari, Fristi, et al. 2023. Classification of Book Types Using the Support Vector Machine (SVM) Method." *Jurnal Mantik* 6.(1), 43-49.
- Riandari F., Sihotang HT., Tarigan T., Rafli M., 2022. Classification of Book Types Using the Support Vector Machine (SVM) Method. *Jurnal Mantik* 6(1), 43-9.