JURNAL TEKNOLOGI TERPADU VOL. 11 NO. 2 OKTOBER 2023 ISSN: 2338 - 6649

Received: Juli 2023	Accepted: Agustus 2023	Published: Oktober 2023

Perancangan Klasifikasi Algoritma *Naive Bayes* Pada Data Pemilihan Jurusan Siswa

Juan Fakhri[1*], Aswan Supriyadi Sunge[2], Ahmad Trmudi Zy[3]
[1,2,3]Universitas Pelita Bangsa*

**Email: juanfakhri02@gmail.com*

Abstract

This research focuses on the development of the Naive Bayes algorithm for the purpose of classification in predicting students' choice of majors. The study evaluates the impact of class imbalance on the classification model's performance and discovers unintended data oversampling. Experiments are conducted using the SMOTE technique to address the class imbalance issue. Student major selection data from SMA Negeri 2 Cikarang Selatan is employed to identify class imbalance. This study compares the classification outcomes before and after the implementation of the SMOTE technique. The analysis results indicate that significant class imbalance exists in the student major selection data before employing the SMOTE technique. This imbalance negatively affects the performance of the classification model, particularly in recognizing minority classes. However, after applying the SMOTE technique, the class imbalance is effectively reduced, leading to a significant improvement in the classification model's performance. Prior to using the SMOTE technique, the classification model's accuracy was 85.19%, and the F1 Score was 33.3%. However, after applying the SMOTE technique, the accuracy increased to 85.71%, and the F1 Score improved to 92%. In conclusion, the application of the SMOTE technique successfully mitigates class imbalance and enhances the classification model's performance in predicting students' choice of majors. The Naive Bayes method can be utilized as an effective alternative for predicting students' major selections at SMA Negeri 2 Cikarang Selatan after applying the SMOTE technique.

Keywords: Major selection, classification, Naive Bayes, SMOTE

Abstrak

Penelitian ini berfokus pada pengembangan algoritma Naive Bayes untuk keperluan klasifikasi dalam konteks memprediksi pemilihan jurusan siswa. Mengevaluasi dampak ketidak seimbangan kelas terhadap kinerja model klasifikasi, serta menemukan kelebihan sampel data secara tidak sengaja, melakukan eksperimen dengan teknik *SMOTE* untuk mengatasi ketidak seimbangan kelas tersebut. Data pemilihan jurusan siswa dari SMA Negeri 2 Cikarang Selatan digunakan untuk mengidentifikasi ketidak seimbangan kelas. Penelitian ini membandingkan hasil klasifikasi sebelum dan setelah penerapan teknik *SMOTE*. Hasil analisis menunjukkan bahwa sebelum menggunakan teknik *SMOTE*, terdapat ketidak seimbangan kelas yang signifikan dalam data pemilihan jurusan siswa. Ketidak seimbangan ini memiliki dampak negatif terhadap kinerja model klasifikasi, terutama dalam mengenali kelas minoritas. Namun, setelah penerapan teknik *SMOTE*, ketidak seimbangan kelas berhasil dikurangi dan kinerja model klasifikasi mengalami peningkatan yang signifikan. Sebelum penerapan teknik *SMOTE*, akurasi model klasifikasi adalah 85,19% dan *F1 Score* adalah 33,3%. Namun, setelah penerapan teknik *SMOTE*, akurasi meningkat menjadi 85,71% dan *F1 Score* meningkat menjadi 92%. Dengan demikian, dapat disimpulkan bahwa penerapan teknik *SMOTE* berhasil mengurangi ketidak seimbangan kelas dan meningkatkan kinerja model klasifikasi dalam memprediksi pemilihan jurusan siswa. Metode *Naive Bayes* dapat digunakan sebagai alternatif yang efektif dalam memprediksi penjurusan siswa di SMA Negeri 2 Cikarang Selatan setelah menerapkan teknik *SMOTE*.

Kata kunci: Pemilihan Jurusan, klasifikasi, *Naive Bayes*, *SMOTE*

1. Pendahuluan

Pendidikan dapat dilakukan di berbagai tingkatan, seperti pendidikan anak usia dini, pendidikan dasar (seperti sekolah dasar dan sekolah menengah pertama), pendidikan menengah (seperti sekolah menengah atas atau sekolah menengah kejuruan), dan pendidikan tinggi (seperti perguruan tinggi atau universitas). Selain itu, pendidikan juga dapat berlangsung di luar lingkungan formal seperti melalui pendidikan non-formal dan informal [1].

Sekolah Menengah Atas (SMA) di Indonesia termasuk dalam Sistem Pendidikan Nasional, yang diatur secara sentralistik oleh pemerintah pusat dan berlaku di seluruh negeri. Sistem pendidikan di Indonesia terus kehilangan makna pendidikan secara empirik dan perlu memperbaiki pendidikan karakter sebagai bagian terpenting dari pengembangan sumber daya manusia. Keberhasilan pendidikan dapat dicapai oleh guru yang memiliki sifat positif, kecerdasaan emosional yang stabil, dan keahlian dalam materi pelajaran dan disiplin [2].

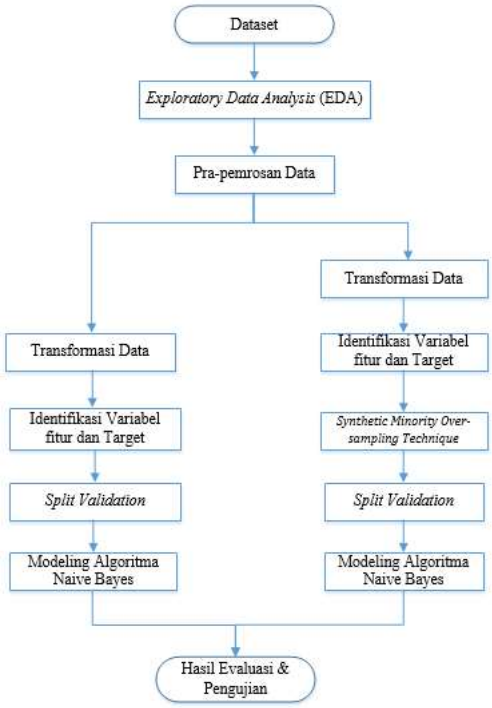
Jurusan atau program studi yang akan diambil oleh siswa pada jenjang pendidikan tertentu ditentukan berdasarkan kemampuan, minat, dan potensi mereka. Pada awal kurikulum baru, seperti pada kelas X SMA, dilakukan untuk memaksimalkan potensi siswa dalam bidang yang sesuai dengan kemampuan dan minatnya, sehingga mereka dapat mencapai Sekolah kesulitan menghitung semua kriteria siswa karena kesulitan mengklasifikasikan jurusan siswa, dan menghitung nilai setiap tes yang sudah dilakukan sebelumnya juga menghabiskan waktu [3].

Dalam era informasi dan teknologi yang semakin maju, data mengenai preferensi siswa terkait pemilihan jurusan tersedia dalam jumlah yang besar. Data ini mencakup faktor-faktor seperti minat pribadi, nilai akademik, aktivitas ekstrakurikuler, dan pendapat dari guru atau orang tua. Namun, membutuhkan waktu yang lebih lama jika proses dan analisis data dilakukan secara manual.

2. Metoda Penelitian

2.1. Tahapan Penelitian

Langkah-langkah penelitian adalah perancangan tahapan atau proses yang dilakukan oleh peneliti untuk menjalankan penelitian. Berikut ini adalah urutan langkah-langkah yang dilakukan oleh peneliti.



Gambar 1. Tahapan Penelitian

Metode yang diterapkan adalah *Naive Bayes*. Sumber data yang digunakan merupakan data sekunder. Data penjurusan diperoleh dari SMA Negeri 2 Cikarang Selatan yang terletak di Jalan Utama Jl. Raya Perum Bumi Cikarang Makmur, Sukadami, Cikarang Selatan, Kabupaten Bekasi, Jawa Barat 17530.

Data yang di dapatkan adalah data sekunder dalam bentuk excel yang teridiri dari 136 field. Field tersebut berisikian NISN, Nama Lengkap, Jenis Kelamin, B.Indo, B.ING, MTK, IPA, IPS, Minat, Impian dan Class. Kemudian, data tersebut diolah menjadi dataset seperti berikut ini:

NAMA LENGKAP	Jenis Kelamin	B, Indo	B, ING	MTK	IPA	IPS	Minat	Impian	Class
DEVANI ARLENIA	Perempuan	91	94	97	89	95	IPS	Kuliah	IPS
Revina Shakila Adhar	Perempuan	89	75	75	78	80	IPS	Bekerja	IPS
PITRI ANI	Perempuan	88	84	88	88	81	IPS	Bekerja	MIPA
NAIFAH ATHALIA SIAHAAN	Perempuan	88	80	88	85	85	IPS	Kuliah	IPS
SYAHRIA JAUZA JUNITA	Perempuan	88	76	77	78	78	IPS	Bekerja	IPS
Merlin maharani	Perempuan	87	84	87	87	87	IPS	Kuliah	IPS
DEWI WULANDARI RAMADANI	Perempuan	86	81	84	83	85	IPS	Bekerja	IPS
ZULPADI	Laki-Laki	85	84	85	86	78	MIPA	Kuliah	MIPA
RAHMA ZAKIATUNNISA	Perempuan	85	83	85	86	79	IPS	Bekerja	IPS
Alexsa Riznanda Bais	Laki-Laki	85	83	90	84	87	IPS	Bekerja	IPS

Gambar 2.Tabel Dataset

2.2. Algoritma *Naive Bayes Classifier*

Salah satu algoritma klasifikasi yang paling banyak digunakan dalam penggalian data dan teks adalah *Naive Bayes* Classifier. Algoritma ini didasarkan pada teorema Bayes bahwa setiap kegiatan memberikan kontribusi yang sama penting atau saling bebas untuk memilih kelas tertentu. Dalam analisis sentimen, *Naive Bayes* Classifier sering digunakan untuk membagi teks atau dokumen ke dalam kategori positif, negatif, atau netral [4].

Algoritma *Naive Bayes* membuat asumsi bahwa setiap atribut dalam dataset saling bebas, meskipun sebenarnya atribut dapat saling terkait. Namun, asumsi ini memudahkan perhitungan probabilitas dalam algoritma tersebut[5].

Naive Bayes terkenal karena sifatnya yang sederhana, efisien, dan kemampuannya dalam menghadapi dataset yang besar. Metode ini sering digunakan dalam berbagai aplikasi, seperti klasifikasi teks, filtrasi spam, analisis sentimen, dan diagnosis penyakit [6].

2.3. Data Mining

Data mining dapat disebut sebagai penambangan data atau penemuan pengetahuan dalam basis data. Hal ini berarti data mining adalah proses analisis data pengamatan yang dilakukan untuk menemukan hubungan yang tidak terduga serta untuk menyusun data dengan cara baru yang dapat dipahami dan bermanfaat bagi pemilik data [7].

Data mining sering disebut sebagai proses yang memanfaatkan teknik statistik, matematika, kecerdasan buatan, dan pembelajaran mesin untuk mengekstraksi dan mengidentifikasi informasi yang berguna serta pengetahuan terkait dari beragam basis data besar [8].

2.4. *Machine Learning*

Pembelajaran mesin (*machine learning*) merupakan bagian dari kecerdasan buatan yang melibatkan pengembangan algoritma dan model yang memungkinkan komputer untuk belajar dan membuat prediksi atau keputusan tanpa perlu diprogram secara eksplisit. Ini adalah proses di mana mesin dapat belajar secara otomatis dari data, mengidentifikasi pola, dan meningkatkan kinerjanya seiring waktu. Algoritma pembelajaran mesin menggunakan teknik statistik untuk menganalisis dan menginterpretasi data, sehingga memungkinkan mereka untuk membuat prediksi atau mengambil tindakan berdasarkan pola dan tren dalam data tersebut.

Pembelajaran mesin terdiri dari tiga jenis, yaitu pembelajaran terpantau (supervised learning), pembelajaran tak terpantau (unsupervised learning), dan pembelajaran semi-terpantau (semi-supervised learning) [9].

2.5. Bahasa Pemograman Python

Pada penelitian ini menggunakan bahasa pemgogrman python. Python adalah sebuah bahasa pemrograman yang sangat populer yang memiliki banyak keuntungan dalam mendukung pemrograman berorientasi objek dan dapat berjalan pada berbagai platform sistem operasi seperti komputer pribadi, Macintosh, dan UNIX. Beberapa keunggulan dari bahasa pemrograman Python adalah sebagai berikut [10]:

- Program dapat dikembangkan dengan cepat dengan sedikit kode yang diperlukan.
- Python mendukung multiplatform, artinya dapat digunakan di berbagai sistem operasi.
- Bahasa pemrograman Python relatif mudah dipelajari.
- Python memiliki sistem pengelolaan memori otomatis.
- Python adalah bahasa pemrograman berorientasi objek.

2.6. *Exploratory Data Analysis*

Exploratory Data Analysis (EDA) digunakan untuk mengeksplorasi data secara visual dan deskriptif dengan tujuan untuk memahami karakteristik data, menemukan pola, dan mengidentifikasi anomali atau outlier dalam data. *Exploratory Data Analysis* (EDA) digunakan sebagai tahap awal dalam analisis data sebelum dilakukan pemodelan dengan metode klasifikasi data mining [11].

.

2.7. Teknik *SMOTE*

SMOTE (*Synthetic Minority Over-sampling Technique*) merupakan teknik *oversampling* yang bertujuan untuk menyeimbangkan ketidak seimbangan antara kelas-kelas dalam dataset. Sampel-sampel baru ini kemudian ditambahkan ke dalam data pelatihan, sehingga pengklasifikasi dapat dilatih dengan menggunakan data yang telah diperluas. Algoritma *SMOTE* umumnya memiliki tingkat akurasi yang lebih baik jika dibandingkan dengan pendekatan *oversampling* yang umum digunakan [12].

Proses *SMOTE* dimulai dengan menghitung jarak antara data kelas minoritas; kemudian, Anda menemukan persentase *SMOTE*; kemudian, Anda menemukan jumlah k terdekat; dan akhirnya, Anda membuat data sintesis [13].

3. Hasil Penelitian

3.1. *Import Libray* dan *Dataset*

Library-library yang umum digunakan untuk analisis dan pembelajaran mesin, seperti pandas, numpy, dan sklearn, akan digunakan dalam tahap pertama ini. Selain itu, dataset yang relevan juga perlu dimuat agar dapat digunakan dalam proses pengolahan dan pembelajaran. Berikut adalah kode program dan hasil untuk langkah ini:

```
import pandas as pd
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import precision_score, recall_score, f1_score
from imblearn.over_sampling import SMOTE
from sklearn import preprocessing

# Baca file CSV dengan Pandas
df = pd.read_csv('dataset.csv')
df.head()
```

Gambar 3. *Import Libray* dan *Dataset*

3.2. *Pre-processing Data*

Pada tahap ke dua, kita perlu melakukan pra-pemrosesan data untuk mempersiapkan dataset sebelum digunakan dalam klasifikasi *Naive Bayes* kita harus membersihkan data mungkin mengandung noise atau nilai yang tidak valid.

Data yang dikumpulkan mungkin mengandung noise atau nilai yang tidak valid. Oleh karena itu, langkah ini melibatkan penngecekan data apabila data tidak valid maka dilakukan pembersihan data agar nilai yang tidak valid tidak mempengaruhi kualitas prediksi. Berikut kode program dan hasil untuk langkah ini:

```
# Fungsi reusable pribadi saya untuk mendeteksi data yang hilang
def missing_value_describe(data):
# Periksa nilai yang hilang dalam data
  missing_value_stats = (data.isnull().sum() / len(data)*100)
  missing_value_col_count = sum(missing_value_stats > 0)
  missing_value_stats = missing_value_stats.sort_values(ascending=False)[:missing_value_col_count]
  print("Jumlah kolom dengan nilai yang hilang:", missing_value_col_count)
  if missing_value_col_count != 0:
      # Mencetak nama kolom dengan persentase nilai yang hilang
      print("\nPersentase hilang (menurun):")
      print(missing_value_stats)
  else:
      print("Tidak ada data yang hilang!!!")
missing_value_describe(df)

Jumlah kolom dengan nilai yang hilang: 0
Tidak ada data yang hilang!!!
```

Gambar 4. Pemeriksaan *missing value*

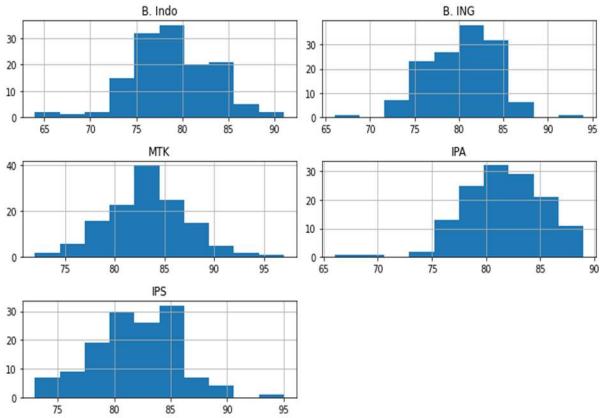
Pada gambar 4 terlihat bahwa tidak ada data yang hilang. Data ini siap untuk melakukan *Exploratory Data Analysis.*

3.3. *Exploratory Data Analysis*

Setelah melakukan pemrosesan data, Tahap ke tiga melakukan *Exploratory Data Analysis* (EDA) untuk memahami dataset. EDA membantu kita mengidentifikasi pola, hubungan, dan karakteristik penting dalam data sebelum melakukan proses pemodelan.

Kita akan memvisualisasikanya dengan setiap variabel numerik, Matriks korelasi, dua variabel numerik dan variabel kategorikal.

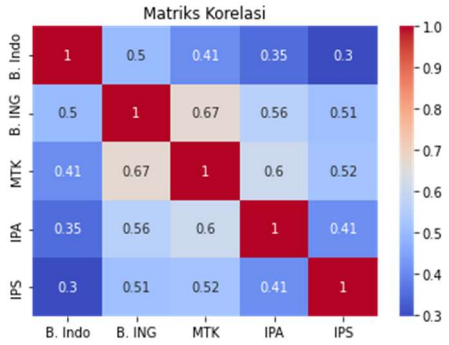
- Setiap variabel numerik



Gambar 5. Histogram untuk setiap variabel numerik

Pada histogram gambar 5 Histogram menampilkan pola distribusi ke kiri, yang berarti frekuensi kemunculan nilai data lebih tinggi pada nilai-nilai yang lebih rendah. Histogram dengan distribusi ke kiri akan memiliki puncak yang lebih rendah dan ekor yang panjang di sisi kiri.

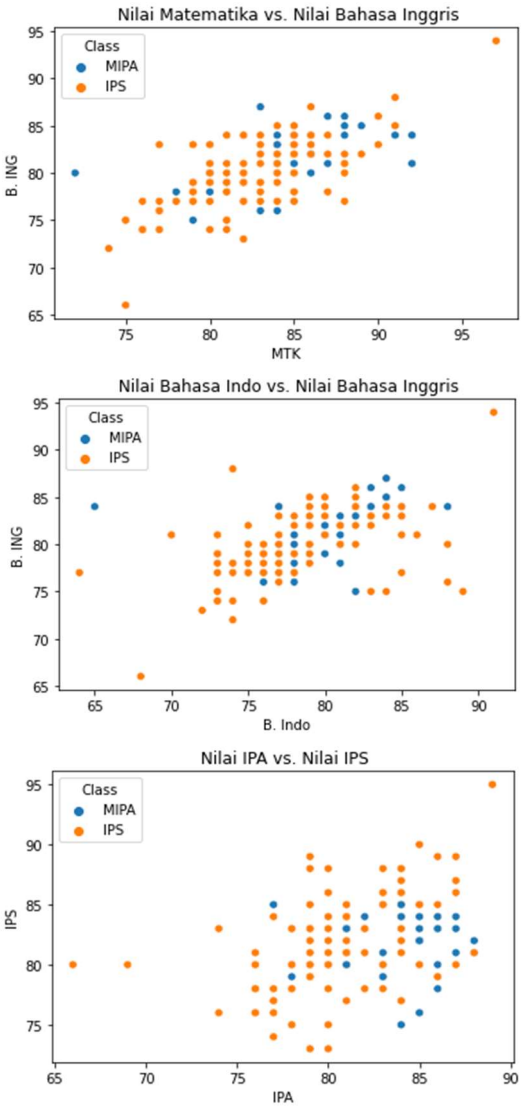
- Matriks korelasi



Gambar 6. Matriks korelasi

Pada Matriks korelasi ini hubungan antara variabel-variabel dalam dataset tidak cukup baik karena banyak korelasi < 0,6 menunjukan tidak adanya korelasi antara dua variabel [14].

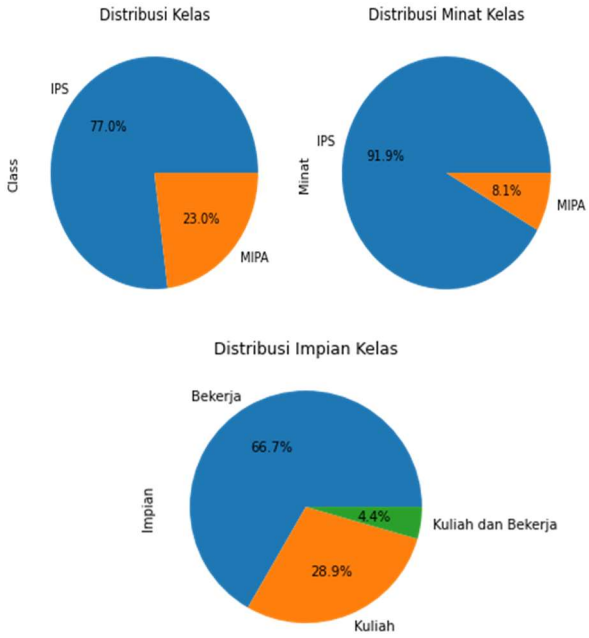
- Dua variabel numerik



Gambar 7. Dua variabel numerik

Pada gambar 7 hubungan antara dua variabel dieksplorasi dengan baik oleh plot, dikarenakan memiliki variasi nilai yang memadai, sehingga variasi yang ada dalam populasi yang diteliti dapat digambarkan dengan baik

- Setiap variabel kategorikal



Gambar 8. Chart pie variabel kategorikal

Pada gambar 8 distribusi kelas target terlihat bahwa data terdapat ketidak seimbangan dimana pada kelas IPS dengan presentase 77,0% sedangkan kelas MIPA presentase rendah 23,0%. Maka kita dapat simpulkan bahwa dataset *oversampling*. Kita akan melakukan teknik *SMOTE* (*Synthetic Minority Over-sampling Technique*) teknik ini menghasilkan sampel sintetis baru dengan mempertimbangkan tetangga terdekat dari setiap sampel dalam kelas minoritas. Sampel sintetis ini dibuat dengan melakukan interpolasi antara sampel-sampel yang ada.

3.4. Transformasi Data

Tahap ke empat, lakukan transformasi data pada kolom minat, impian dan kelas dikarenakan variabel kolom tersebut mempunyai nilai kategori yang nantinya akan di ubah ke numeric. Untuk melakukan transformasi data library yang diperlukan yaitu sklearn prep*ROC*essing.

```
# Transformasi data.
label_encoder = preprocessing.LabelEncoder()
df['Minat']= label_encoder.fit_transform(df['Minat'])
df['Impian']= label_encoder.fit_transform(df['Impian'])
df['Class']= label_encoder.fit_transform(df['Class'])
```

Gambar 9. Transformasi data

Pada gambar 9 Hasil transformasi data variable minat dan *class* adalah MIPA = 1 dan IPS = 0 sedangan pada variable Impian adalah bekerja = 1, kuliah = 0 dan kuliah dan bekerja = 2. Berikut gambar table hasil transformasi data:

Tabel 1. Hasil transformasi data

Atribut	Subset	Nilai
Minat	IPS	0
	MIPA	1
Impian	Kuliah	0
	Bekerja	1
	Kuliah dan Bekerja	2
Class	IPS	0
	MIPA	1

3.5. Identifikasi Variabel fitur dan Target

Pada tahap ke lima, identifikasi variabel fitur dan Target yang akan diprediksi. Misalnya, dalam dataset kita, variabel targetnya(y) adalah "kelas" yang dapat bernilai MIPA atau IPS. Dan variabel fitunya(X) adalah B.Indo, B. ING, MTK, IPA, IPS, Minat, dan Impian. Berikut kode program dan hasil variabel fitur dan target:

```
features = ["B. Indo","B. ING","MTK","IPA","IPS",'Minat','Impian']
X = df[features]
y = df["Class"]
```

	B. Indo	B. ING	MTK	IPA	IPS	Minat	Impian
0	85	84	85	86	78	1	1
1	85	81	83	86	79	0	0
2	85	77	83	81	82	0	0
3	74	78	87	84	86	0	1
4	83	75	81	79	73	0	0
...	...	...	...	...	...	...	...
130	79	80	80	81	84	0	0
131	78	79	80	81	84	0	0
132	78	76	84	81	80	0	1
133	78	77	85	78	83	0	2
134	82	84	86	66	80	0	2

135 rows × 7 columns

Variable fitur(X)

	Class
0	1
1	0
2	0
3	0
4	0
...	...
130	0
131	0
132	1
133	0
134	0

135 rows × 1 co

Variable target(y)

Gambar 10. Identifikasi Variabel fitur dan Target

3.6. *Split Validation*

Tahap ke enam, setalah transformasi dan identifikasi variabel fitur &Target bagi data menjadi data pelatihan dan data pengujian. Data pelatihan digunakan untuk melatih model, sedangkan data pengujian digunakan untuk menguji kinerja model yang telah dilatih. Pada penelitian ini akan dibagi dalam perbandingan 80:20 [15].

```
# Pembagian data
X_train, X_test, y_train, y_test = train_test_split(
    X_oversampled, y_oversampled, test_size=0.2, random_state=41)
```

Gambar 11. Pembagian Data

3.7. Pemodelan *Naive Bayes*

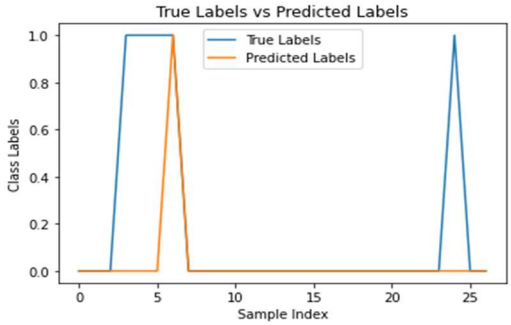
Pada tahap terakhir, setelah melakukan pembagian data langkah selanjutnya adalah membangun model klasifikasi menggunakan algoritma *Naive Bayes*. Ketika melakukan *Exploratory Data Analysis* penelitian ini terdapat *oversampling* maka akan dilakukan perbandingan antara data asli tanpa *oversampling* dengan model data setelah dilakukan *oversampling*. Tujuannya adalah untuk untuk membuat perbandingan dengan model yang menggunakan data setelah dilakukan *oversampling* dan melihat apakah *oversampling* dapat meningkatkan kinerja model dalam mengklasifikasikan data.

- Data asli tanpa *oversampling*

Pada tahapan ini membangun model klasifikasi menggunakan data asli tanpa melakukan *oversampling* dengan algoritma *Naive Bayes*. Berikut kode program dan visualisasi hasil prediksi:

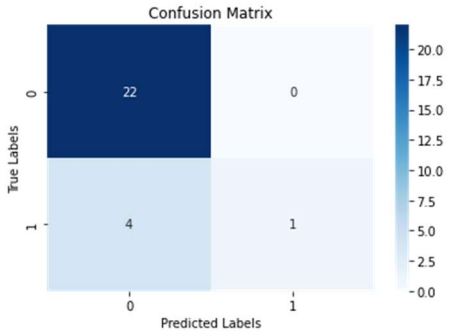
```
# Impelentasi algoritma naive bayes
nb = GaussianNB()
nb.fit(X_train, y_train)
predictions = nb.predict(X_test)
```

Gambar 12. Kode Program algoritma *Naive Bayes*



Gambar 13. Visualisasi hasil prediksi

Setelah kita melihat visualisasi pada gambar 13 dapat disimpulkan bahwa hasil model kurang dikarenakan model tersebut memiliki kinerja yang lebih rendah dalam mengklasifikasikan data kelas 1. Kemudian dilakukan validasi serta pengukuran keakuratan hasil yang dicapai oleh model menggunakan pembelajaran mesin (*machine learning*) menggunakan bahasa pemograman python dengan melihat *confussion matrix*, dapat dilihat pada gambar 14 berikut :



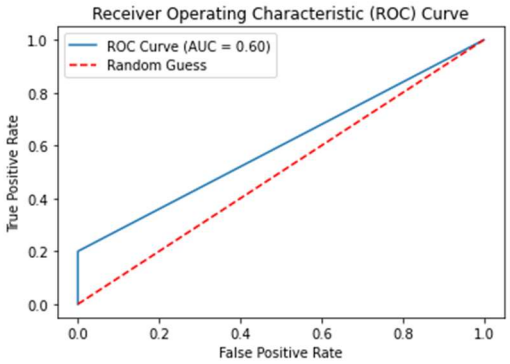
Gambar 14. Confussion matrix

Dari confussion matrix diatas didapat nilai: *true positive* (tp) sebanyak 22 record, *false positive* (fp) sebanyak 4 record, *true negative* (tn) sebanyak 0 record, *false negative* (fn) sebanyak 1 record. *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Berdasarkan analisis menggunakan pembelajaran mesin (*machine learning*) menggunakan bahasa pemograman python dengan pengukuran *Naive Bayes* didapatkan hasil dengan tingkat:

Tabel 2. Hasil *Accuracy*, Precision dan Recall

Accuracy	*Precission*	*Recall*	*F1 Score*
85,19%	100%	20%	33,3%

Untuk mengetahui evaluasi kinerja secara lebih menyeluruh, Berikut hasil *ROC* dan *AUC*:



Gambar 15. Plot *ROC* dan *AUC*

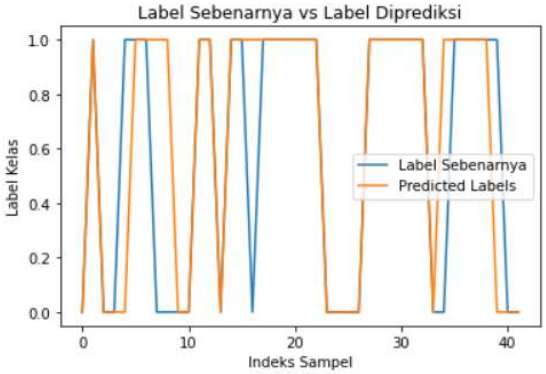
Keakuratan *AUC* dikatakan sempurna ketika nilai *AUC* mencapai 1,00 dan keakuratan menjadi buruk jika nilai *AUC* berada di bawah 0,500. Pada gambar 15 bernilai 0.60 nilai ini menunjukkan bahwa model memiliki kemampuan yang sedang untuk membedakan antara kelas positif dan negatif. masuk dalam tingkat diagnosa Poor Classification.

- Data setelah dilakukan *oversampling*

Setelah mempelajari data asli tanpa *oversampling*, Kita akan melakukan *oversampling* pada dataset menggunakan teknik *SMOTE* (*Synthetic Minority Over-sampling Technique*). Berikut kode program dan visualisasi hasil prediksi:

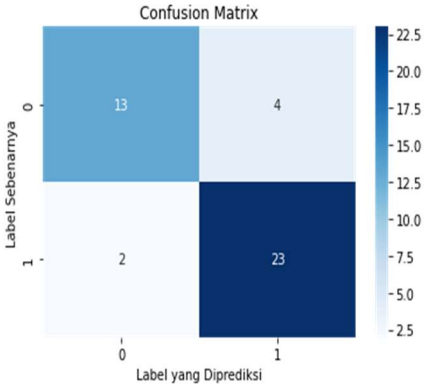
```
# Masukan kembali dataset untuk melakukan teknik SMOTE
df = pd.read_csv('dataset.csv')
# Lakukan transformasi data kembali
label_encoder = preprocessing.LabelEncoder()
df['Minat']= label_encoder.fit_transform(df['Minat'])
df['Impian']= label_encoder.fit_transform(df['Impian'])
df['Class']= label_encoder.fit_transform(df['Class'])
# Lakukan kembali Identifikasi Variabel fitur dan Target
features = ["B. Indo","B. ING","MTK","IPA","IPS",'Minat','Impian']
X = df[features]
y = df["Class"]
# Melakukan teknik SMOTE
smote = SMOTE()
X_oversampled, y_oversampled = smote.fit_resample(X, y)
# Setalah teknik SMOTE Lakukan pembagian data
X_train, X_test, y_train, y_test = train_test_split(
    X_oversampled, y_oversampled, test_size=0.2, random_state=41)
# Lakukan impelentasi algoritma naive bayes kembali untuk melihat hasil prediksi
nb = GaussianNB()
nb.fit(X_train, y_train)
predictions = nb.predict(X_test)
```

Gambar 16. Kode program *Synthetic Minority Over-sampling Technique* dengan algoritma *Naive Bayes*



Gambar 17. Visualisasi hasil prediksi *Synthetic Minority Over-sampling Technique*

Kita dapat melihat visualisasi pada gambar 17 dapat disimpulkan bahwa hasil model sudah bagus karena mampu memiliki kinerja yang cukup baik dalam mengklasifikasikan data kelas 1 dan kelas 0. Kemudian dilakukan validasi serta pengukuran keakuratan hasil yang dicapai oleh model dengan melihat *confussion matrix*, dapat dilihat pada gambar 18 berikut:



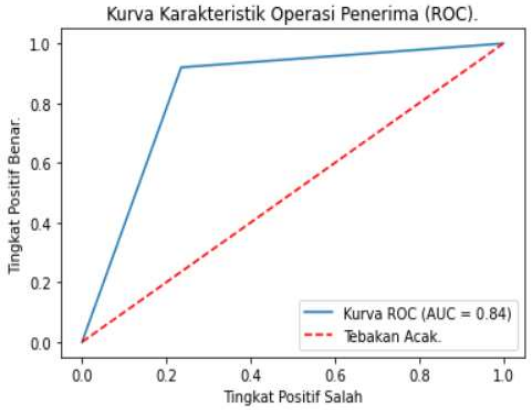
Gambar 18. Confusion matrix dari *Synthetic Minority Over-sampling Technique*

Dari confussion matrix diatas didapat nilai: *true positive* (tp) sebanyak 13 record, *false positive* (fp) sebanyak 2 record, *true negative* (tn) sebanyak 4 record, *false negative* (fn) sebanyak 23 record. *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Hasil analisis yang dilakukan menggunakan pembelajaran mesin (*machine mearning*) menggunakan bahasa pemograman Python dengan pengukuran *Naive Bayes* ditemukan dalam tingkat:

Tabel 3. Hasil *Accuracy*, Precision dan Recall dari *Synthetic Minority Over-sampling Technique*

Accuracy	***Precission***	***Recall***	***F1 Score***
85,71%	85%	92%	88,46%

Berikut Hasil *ROC* dan *AUC* berikut menunjukkan evaluasi kinerja teknik over-sampling sintetika minoritas:



Gambar 19. Plot *ROC* dan *AUC* dari *Synthetic Minority Over-sampling Technique*

Akurasi *AUC* gambar 19 bernilai 0.84 nilai ini cukup tinggi, menunjukkan bahwa model yang telah dilakukan *Synthetic Minority Over-sampling Technique* memiliki kinerja yang lebih baik dalam membedakan antara kelas positif dan negatif. itu berarti hasil klasifikasi penelitian ini masuk ke dalam tingkat diagnosa Good Classification

4. Kesimpulan

Hasil pengujian menunjukkan bahwa model klasifikasi *Naive Bayes* dengan data sebelum *oversampling* memiliki nilai precision sebesar 1.0, recall sebesar 0.20, dan *F1 Score* sebesar 0.33 dan data setelah dilakukan *oversampling* memiliki nilai precision sebesar 0.85, recall sebesar 0.92, dan *F1 Score* sebesar 0.88. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengklasifikasikan data pemilihan jurusan siswa.

Ketidak seimbangan kelas dalam data pemilihan jurusan siswa memiliki pengaruh signifikan terhadap kinerja model klasifikasi. Ketidaks eimbangan ini menyebabkan model cenderung memprediksi dengan baik untuk kelas mayoritas, namun memiliki kinerja yang lebih rendah untuk kelas minoritas. Pada penelitian ini, dilakukan pengujian menggunakan teknik *SMOTE* untuk menyeimbangkan jumlah sampel antara kelas MIPA terhadap kelas IPS.

Dengan menggunakan data mining, langkah demi langkah pengolahan data dengan metode *Naive Bayes* menghasilkan nilai probabilitas untuk setiap kriteria untuk kelas yang berbeda. Nilai probabilitas dari kriteria ini dapat dioptimalkan untuk mempercepat pemilihan jurusan.

5. Saran

Pada penelitian selanjutnya diharapkan untuk membuat versi aplikasi atau web yang lebih mudah digunakan dan diakses untuk pengembangan sistem yang akan datang.

Penting untuk menangani ketidak seimbangan kelas dalam data pemilihan jurusan siswa. Dalam penelitian ini, penggunaan teknik *SMOTE* terbukti efektif

dalam menyeimbangkan jumlah sampel antara kelas MIPA dan kelas IPS. Oleh karena itu, direkomendasikan untuk menerapkan teknik *SMOTE* atau teknik seimbang lainnya untuk mengatasi masalah ketidak seimbangan kelas dalam model klasifikasi.

Melakukan penelitian lanjutan untuk mengidentifikasi faktor-faktor lain yang dapat mempengaruhi pemilihan jurusan siswa. Mengintegrasikan variabel-variabel seperti minat, potensi akademik, dan preferensi individu dapat membantu memperbaiki model klasifikasi dan meningkatkan prediksi pemilihan jurusan.

6. Daftar Pustaka

[1] M. S. Prof. Dr. Hamid Darmadi, M.PD., *PENGANTAR PENDIDIKAN ERA GLOBALISASI: Konsep Dasar,Teori, Strategi dan Implementasi dalam Pendidikan Globalisasi*. An1mage, 2019.

[2] N. Afifah, "Sistem pendidikan di indonesia," no. April, 2020.

[3] B. A. Rahmadi and Mufti, "Sistem Penjurusan IPA/IPS Menggunakan Algoritma K-Nearest Neighbor Pada SMA Muhammadiyah 13 Jakarta," *Semin. Nas. Sains Teknol. Inf.*, pp. 300–305, 2019, [Online]. Available: http://prosiding.seminar-id.com/index.php/sensasi/issue/archivePage%7C300

[4] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional," *J. Tekno Kompak*, vol. 15, no. 1, p. 131, 2021, doi: 10.33365/jtk.v15i1.744.

[5] H. Putri, A. I. Purnamasari, A. R. Dikananda, O. Nurdiawan, and S. Anwar, "Penerima Manfaat Bantuan Non Tunai Kartu Keluarga Sejahtera Menggunakan Metode NAÏVE BAYES dan KNN," *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 331–337, 2021, doi: 10.47065/bits.v3i3.1093.

[6] N. Salmi and Z. Rustam, "Naïve Bayes Classifier Models for Predicting the Colon Cancer," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 546, no. 5, 2019, doi: 10.1088/1757-899X/546/5/052068.

[7] T. R. Rivanthio, M. Ramdhani, and A. Sahi, "Penerapan Teknik Clustering Data Mining untuk Memprediksi Kesesuaian Jurusan Siswa (Studi Kasus SMA PGRI 1 Subang)," vol. 13, no. 2, pp. 125–131, 2020, doi: 10.30998/faktorexacta.v13i2.6588.

[8] Y. N. Lubis and H. Winata, "Data Mining Untuk Memprediksi Data Pengunjung dengan Menggunakan Algoritma Simple Moving Average," vol. 21, no. 2, pp. 50–59, 2022.

[9] F. D. Telaumbanua, P. Hulu, T. Z. Nadeak, R. R. Lumbantong, and A. Dharma, "Penggunaan Machine Learning," *J. Teknol. dan Ilmu Komput.*, vol. 3, no. 1, pp. 57–64, 2019.

[10] V. S. Ginting, K. Kusrini, and E. Taufiq, "Implementasi Algoritma C4.5 untuk Memprediksi Keterlambatan Pembayaran Sumbangan Pembangunan Pendidikan Sekolah Menggunakan Python," *Inspir. J. Teknol. Inf. dan Komun.*, vol. 10, no. 1, pp. 36–44, 2020, doi: 10.35585/inspir.v10i1.2535.

[11] E. D. Wahyuni, A. A. Arifiyanti, and M. Kustyani, "Exploratory Data Analysis dalam Konteks Klasifikasi Data Mining," *Pros. Nas. Rekayasa Teknol. Ind. dan Inf. XIV Tahun 2019*, vol. 2019, no. November, pp. 263–269, 2019, [Online]. Available: http://journal.itny.ac.id/index.php/ReTII

[12] R. Fahlapi *et al.*, "Analisa sentimen vaksinasi covid-19 dengan metode support vector machine dan naïve bayes berbasis teknik smote," *J. Inform. Kaputama*, vol. 6, no. 1, pp. 57–64, 2022.

[13] S. Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti, A. Nikmatul Kasanah, U. Pujianto, T. Elektro, F. Teknik, and U. Negeri Malang, "Terakreditasi SINTA Peringkat 2 Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *Masa Berlaku Mulai*, vol. 1, no. 3, pp. 196–201, 2017.

[14] D. Anggaini, A. Senen, and H. S. Dini, "Proyeksi Kebutuhan Energi Secara Microspasial Berdasarkan Penentuan Variabel Independen Dengan Metode Kolmogorov-Smirnov," *Kilat*, vol. 10, no. 2, pp. 349–358, 2021, doi: 10.33322/kilat.v10i2.1401.

[15] R. B. Widodo, W. Swastika, and ..., "Studi Pemrosesan Data Pengenalan Gestur Tangan Menggunakan Metode Knn," *... Innov. ...*, no. Ciastech, pp. 277–286, 2021, [Online]. Available: http://publishing-widyagama.ac.id/ejournal-v2/index.php/ciastech/article/view/3320.