



Studi Kasus

Rancangan Sistem Rekomendasi Topik Tugas Akhir dengan Naive Bayes Classifier (Studi Kasus Departemen Sistem Informasi, Universitas Andalas)

Hanif Izza Pratama^a, Siti Nur Aisah^a, Fajril Akbar^{a,*}

^aDepartemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas, Padang, 25163, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 10 Juni 2023

Revisi Akhir: 23 September 2025

Diterbitkan Online: 26 September 2025

KATA KUNCI

Rekomendasi Topik,
Tugas Akhir,
Machine Learning,
Naive Bayes Classifier

KORESPONDENSI

E-mail: ijab@it.unand.ac.id*

A B S T R A C T

Tugas akhir merupakan salah satu syarat yang harus dipenuhi oleh mahasiswa untuk menyelesaikan pendidikan di perguruan tinggi. Namun, seringkali mahasiswa mengalami kesulitan dalam menentukan topik yang sesuai. Oleh sebab itu, dilakukan pembangunan sebuah sistem web classification yang menjadi solusi efektif. Dalam penelitian ini digunakan metode klasifikasi Naive Bayes Classifier dalam pengembangan sistem web classification untuk mengklasifikasikan judul tugas akhir mahasiswa Departemen Sistem Informasi Universitas Andalas, data yang digunakan diperoleh dari nilai mata kuliah Struktur Data Algoritma (SDA), Sistem Informasi Geografis (SIG), Pemrograman Teknologi Bergerak (PTB), Pemrograman Web (PWeb), Sistem Informasi Manajemen (SIM), Enterprise Resource Planning (ERP), Business Intelligence (BI), dan Machine Learning (ML). Data diolah dengan membangun aplikasi yang menerapkan algoritma Naive Bayes menggunakan Python. Hasil keluaran sistem memiliki akurasi 83%. Hasil pengujian yang dilakukan dengan perbandingan data latih 80% dan data uji 20% menyatakan algoritma Naive Bayes memiliki performa dengan hasil Accuracy, Recall, Precision, F1-score dengan nilai 83%, 185%, 84% dan 84%. Penelitian ini dapat memberikan rekomendasi untuk topik tugas akhir bagi mahasiswa.

1. PENDAHULUAN

Departemen Sistem Informasi Universitas Andalas merupakan salah satu lembaga pendidikan tinggi yang menawarkan program studi dalam bidang sistem informasi. Tugas akhir merupakan komponen penting dari program studi ini, di mana mahasiswa diharapkan untuk melakukan penelitian yang mendalam dan menghasilkan kontribusi yang berarti dalam bidang sistem informasi. Tugas akhir merupakan salah satu syarat yang harus dipenuhi oleh mahasiswa untuk menyelesaikan pendidikan di perguruan tinggi. Ilmu yang diperoleh mahasiswa dituangkan ke dalam suatu penelitian yang nantinya akan menghasilkan keluaran berupa dokumen tugas akhir [1]. Dalam proses pengerjaan tugas akhir, mahasiswa membutuhkan dosen pembimbing sebagai media berkonsultasi [2]. Dosen pembimbing tugas akhir memiliki peran penting karena memiliki tanggung jawab untuk memastikan bahwa mahasiswa mampu

menyusun tugas akhir dengan baik hingga dapat menghasilkan tugas akhir yang berkualitas dan siap untuk diuji [3]. Untuk itu dibutuhkan dosen pembimbing yang tepat dengan bidang keahlian sesuai dengan kategori/ bidang ilmu topik tugas akhir yang diajukan mahasiswa dalam proposal tugas akhir. Hal ini bertujuan agar konsep dan perancangan tugas akhir yang dibuat dalam proposal dapat dikerjakan dan diwujudkan sesuai dengan tujuan yang direncanakan, dengan harapan tidak adanya kendala [4].

Dalam tugas akhir, pemilihan topik yang tepat sangat penting. Topik yang relevan dengan minat dan keahlian mahasiswa dapat memotivasi mereka untuk menghasilkan karya yang berkualitas tinggi. Namun, seringkali mahasiswa mengalami kesulitan dalam menentukan topik yang sesuai. Dosen pembimbing juga perlu memberikan panduan yang tepat untuk memastikan mahasiswa memilih topik yang relevan dan dapat diteliti dengan baik.

Untuk mengatasi kendala ini, dilakukan pembangunan sebuah sistem web *classification* yang menjadi solusi efektif. Dalam penelitian ini digunakan metode klasifikasi *Naive Bayes Classifier* dalam pengembangan sistem web *classification* untuk merekomendasikan topik tugas akhir mahasiswa Departemen Sistem Informasi Universitas Andalas. *Naive Bayes Classifier* (NBC) merupakan salah satu algoritma klasifikasi untuk mengklasifikasikan sekumpulan teks kedalam kelas yang ada. Algoritma ini memanfaatkan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya [5] [6].

Berdasarkan dari penelitian sebelumnya bahwasannya *Naive Bayes Classifier* mampu memberikan hasil yang lebih tinggi dibandingkan dengan metode lainnya [7], [8], [9], [10]. Seperti penelitian yang dilakukan oleh Pramudita, dkk hasil dari penelitiannya yaitu klasifikasi berita olahraga menggunakan *Naive Bayes Classifier* dengan enhanced confix stripping stemmer, didapat hasil dengan tingkat keakuratan 77% serta metode *Naive Bayes Classifier* mampu digunakan dalam klasifikasi berita olahraga dengan baik [11]. Selain itu, penelitian yang dilakukan oleh devita, dkk berfokus pada klasifikasi artikel bahasa indonesia dengan membandingkan kinerja 2 metode yaitu metode KNN dan NBC, penelitiannya menggunakan data jurnal sebagai data latih dan uji serta abstrak sebagai objeknya, didapatkan hasil bahwa metode *Naive Bayes* memiliki kinerja yang lebih baik dengan akurasi 70%, sedangkan metode KNN memiliki akurasi 40% [12].

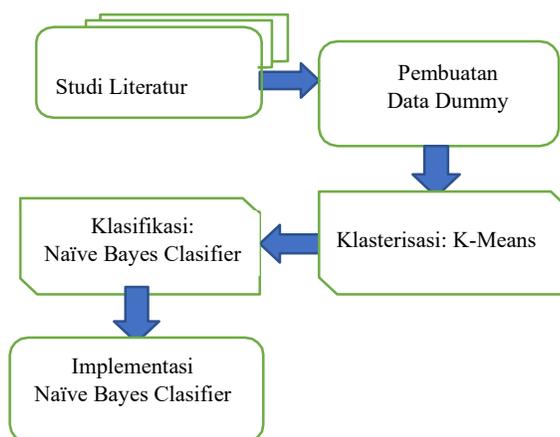
Sedangkan penelitian yang penulis lakukan yaitu web *clustering* untuk klasterisasi topik tugas akhir mahasiswa Departemen

Sistem Informasi Universitas Andalas menggunakan algoritma *Naive Bayes Classifier* yang mana sistem ini akan menerima input berupa nilai dari mata kuliah yang relevan, seperti Struktur Data Algoritma (SDA), Sistem Informasi Geografis (SIG), Pemrograman Teknologi Bergerak (PTB), Pemrograman Web (PWeb), Sistem Informasi Manajemen (SIM), *Enterprise Resource Planning* (ERP), *Business Intelligence* (BI), dan *Machine Learning* (ML). Nilai-nilai ini akan digunakan sebagai fitur dalam algoritma *Naive Bayes Classifier* untuk mengklasifikasikan topik tugas akhir ke dalam empat kelas yang telah ditentukan, yaitu ERP, Enterprise Architecture (EA), BI, dan GIS. Implementasi sistem web *classification* ini akan menggunakan bahasa *Python* dengan bantuan framework Streamlit.

Dengan adanya sistem web *classification* ini, diharapkan mahasiswa Departemen Sistem Informasi Universitas Andalas dapat dengan mudah menemukan topik tugas akhir yang sesuai dengan minat dan keahlian mereka. Selain itu, sistem ini juga akan membantu dosen pembimbing dalam memberikan panduan yang tepat kepada mahasiswa, mempercepat proses pengawasan tugas akhir, dan memastikan kualitas penyelesaian tugas akhir yang lebih baik.

2. METODE

Adapun metodologi yang diterapkan dalam melakukan penelitian ditunjukkan pada gambar 1.



Gambar 1. Metode penelitian

2.1 Studi Literatur

Pada tahap ini dilakukan pencarian beberapa literatur yang berkaitan dengan permasalahan dan bagaimana mencari solusi terbaik dari masalah tersebut. Studi literatur dilakukan untuk mendapatkan teori-teori tentang penelitian melalui jurnal, media online, buku atau penelitian lain yang sebelumnya berhubungan dengan penelitian yang akan dilakukan.

2.2 Pembuatan Data Dummy

Dataset yang penulis gunakan adalah data *dummy* yang terdiri dari 150 data, berupa kolom nama mahasiswa, nilai mata kuliah (Struktur Data Algoritma (SDA), Sistem Informasi Geografis (SIG), Pemrograman Teknologi Bergerak (PTB), Pemrograman Web (PWeb), Sistem Informasi Manajemen (SIM), *Enterprise Resource Planning* (ERP), *Business Intelligence* (BI), dan *Machine Learning* (ML)), dengan skala nilai dari rentang 0.00 -

4.00.

2.3 Clusterisasi

Klasterisasi adalah salah satu metode dalam analisis data yang digunakan untuk mengelompokkan objek-objek data ke dalam kelompok-kelompok yang serupa berdasarkan kemiripan karakteristiknya. Tujuan utama dari klasterisasi adalah untuk mengidentifikasi pola atau struktur yang ada dalam data tanpa adanya label atau klasifikasi sebelumnya.

Salah satu algoritma klasterisasi yang populer adalah K-means. K-means adalah metode klasterisasi yang berdasarkan pada pemisahan data ke dalam k kelompok yang didefinisikan sebelumnya. Algoritma ini bekerja dengan mencari pusat-pusat klaster yang optimal untuk mengurangi total varians dalam klaster-klaster yang terbentuk. Setiap objek data akan ditempatkan ke dalam klaster yang memiliki pusat terdekat dengannya berdasarkan jarak *Euklides* atau metrik lainnya. *K-means* adalah metode yang efisien dan mudah diimplementasikan, dan sering digunakan dalam berbagai bidang seperti pengelompokan konsumen, analisis citra, dan pemrosesan teks.

2.4 Klasifikasi

Klasifikasi merupakan kegiatan dalam mengukur objek data untuk dikelompokkan ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Klasifikasi *Naive Bayes* diasumsikan bahwa ada atau tidaknya suatu ciri tertentu dari sebuah kelas dan tidak memiliki hubungan dengan ciri dari kelas lainnya. *Naive Bayes* berpotensi cukup tinggi untuk mengklasifikasikan data karena kesederhanaannya [13]. Dalam klasifikasi, terdapat dua jenis pekerjaan utama yang dilakukan, yaitu pembangunan model sebagai prototype untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data baru agar dapat diketahui kelas mana objek data tersebut berdasarkan model yang sudah disimpan [14], [15], [16]

2.5 Implementasi Algoritma Naive Bayes Classifier

Algoritma NBC merupakan metode klasifikasi statistik yang didasarkan oleh teorema Bayes [17]. Setelah dataset dipersiapkan untuk menyeimbangkan antara jumlah data training dan testing maka data akan diberikan label berdasarkan *rule*. Selanjutnya, dilakukan estimasi probabilitas kelas dan probabilitas kondisional atribut pada set pelatihan. Probabilitas kelas mengacu pada kemungkinan seorang mahasiswa termasuk dalam salah satu dari empat fokus ilmu: ERP, EA, BI, atau GIS. Sementara itu, probabilitas kondisional atribut mengukur hubungan antara nilai atribut (nilai mata kuliah) dengan kelas (fokus ilmu).

Setelah itu, sistem web *classification* menggunakan metode *Naive Bayes* untuk mengklasifikasikan topik tugas akhir berdasarkan data nilai dari mahasiswa. Saat seorang mahasiswa mengunggah nilai mata kuliah yang diminta, sistem akan menghitung probabilitas untuk setiap kelas (fokus ilmu) berdasarkan atribut (nilai mata kuliah) yang diberikan. Mahasiswa akan diberikan rekomendasi tugas akhir berdasarkan kelas dengan probabilitas tertinggi.

3. HASIL & PEMBAHASAN

3.1. Pembuatan Data Dummy

Dalam beberapa situasi, sulit untuk mendapatkan data asli yang sesuai dengan kebutuhan analisis atau percobaan kita. Dalam hal seperti itu, penggunaan data dummy atau data sintetis sering digunakan sebagai alternatif. Data dummy adalah data buatan yang dibuat dengan tujuan untuk meniru atau merepresentasikan karakteristik data asli secara umum. Data dummy dapat digunakan untuk melengkapi atau memperluas dataset yang ada, menguji algoritma atau model baru, atau sebagai contoh dalam demonstrasi atau pelatihan. Meskipun data dummy tidak mewakili situasi yang sebenarnya, namun dengan kehati-hatian dan pemahaman yang baik tentang batasan dan kekurangannya, data dummy dapat memberikan solusi praktis dalam situasi-situasi di mana data asli tidak tersedia atau sulit diperoleh.

Data dummy yang dibuat kali ini berjumlah 150 data dengan 6 atribut (Struktur Data Algoritma (SDA), Sistem Informasi Geografis (GIS), Pemrograman Teknologi Bergerak (PTB), Pemrograman Web (PWeb), Sistem Informasi Manajemen (SIM), *Enterprise Resource Planning* (ERP), *Business Intelligence* (BI), dan *Machine Learning* (ML)). Data dibuat menggunakan Microsoft Excel dengan menerapkan Fungsi *RANDBETWEEN*. *RANDBETWEEN* adalah fungsi yang Mengembalikan angka bilangan bulat acak di antara angka-angka yang Anda tentukan. Bilangan bulat acak baru akan dikembalikan setiap kali lembar kerja dihitung. Penggunaan fungsi *RANDBETWEEN* dapat dilihat pada Gambar 2 dan hasil dari fungsi ini dapat dilihat pada Gambar 3.

```
=INDEX($O$4:$O$10,RANDBETWEEN(1,COUNTA($O$4:$O$10)),1)
A INDEX(array, row_num, [column_num])
N INDEX(reference, row_num, [column_num], [area_num])
```

Gambar 2. Fungsi *RANDBETWEEN*

	NIM	NAMA	SDA	GIS	PTB	PWEB	SIM	ERP	BI	APM
1	20192001	TALUQ HIDAYAT	2,75	2,5	3,5	4	2	3,75	3	3
2	20192003	DINA LATHIFUNNISA A	2,75	2,75	4	3,5	2	4	3,5	3
3	20192005	ANNISA FARAS	3,5	4	2	3,75	2,5	3,75	3	3,75
4	20192007	PRADHYA ROMEIO A	4	3	4	4	3	2,75	2,5	4
5	20192009	PRISKA SHIFA SALSABILLA	3	2	2,5	3,75	4	2	2,5	3,75
6	20192011	FADILA TFI AFFRANI	2,5	4	2	4	3,75	4	2	4
7	20192013	RITI SURALDO	3,5	4	3,75	2,5	3	3,75	3,75	3
8	20192015	HAGI SIRAJ SUKMANITA	2	2,5	2	3	3	2	3	4
9	20192017	IQBAL FITRAHULL RAMADHAN	3	2,5	3,5	2,75	2,5	3,5	3,5	3,75
10	20192019	DAENG FEBRINDO	2	2	4	4	2,75	3,5	2,5	2,75
11	20192021	AULIA DWI SHAVIGIA POHAN	3,5	2	3,5	3,5	2,5	2	2	2
12	20192023	HANIF IZZA PRATAMA	4	3	2,75	3,75	3,75	2	2,75	4
13	20192025	NADA SAFARINA	3,75	2	3,5	3,5	3	2	2,5	3,5
14	20192027	PUTRI WULANDARI	2	4	3,5	3	2,75	3,5	3	4
15	20192201	DZUL FAUZI	3	2	3,5	3,5	2	4	4	4
16	20192205	LUTHFIA HUMAIRA	4	3,75	2,5	2,75	2,75	2,5	3,75	3,5
17	20192307	IQBAL MANAZIL YUNI	3,5	2,5	2,75	4	2	3	2,75	2,5
18	20192309	ALIA MURRIKHAH	2	3	2,75	3,5	4	3,5	3	2
19	20192311	DWISUCI INSANI KARIMAH	2,75	3,5	2	3,75	3	3,75	2	2,5
20	20192313	HAHRID SURFOYA TAURIN	3,5	3	2,75	3,5	2,75	3	2	3,75
21	20192315	ANNISA LILFA	2,5	2,5	3	3,5	2,75	3,75	2,5	4
22	20192317	VALLEN ADHITYA PEKHSANA	2,5	3,75	3,5	3,5	3,75	2	4	4
23	20192319	ILHAM	2	4	3,5	3,75	3,5	2,75	4	2
24	20192321	MUHAMMAD REYHAN RIZALDI	2,75	2,75	3,5	3	3,75	2	2,75	2,75
25	20192323	BOBY DAIRMAWAN	3,75	2	2	2,75	2	4	3	2,5
26	20192325	AINI IZZATHY ISPENDI	3,75	3,75	3,75	2,5	2,5	2,75	3	4
27	20192327	DELICIA SYIFA MASHFIRA	3,5	3	2	3,5	2	3	3	2,5
28	20192329	KHALIL AMIR	2,5	3,75	2,5	3,75	3	2,5	2	2
29	20192331	SALSABILA RAMADHANI PUTRI	2,75	3,75	3,5	2	3,75	4	2,75	4
30	20192333	MUHAMMAD YUDHISTIRA	2	4	2,5	3	2	3,75	3,75	2,75
31	20192335	FARRIS MALIK SUHANDA	3,5	2,75	2,5	2	3,75	3,5	4	3,5
32	20192337	RAIDAN MALIK SANDIRA	4	4	3,75	3,75	3,5	3,75	2,5	2,75
33	20192339	NATHANARAHAYU	3,75	2,5	3	3,5	2,5	3	3,5	3,75
34	20192341	GITA PUTRI	4	3	3,75	2,5	3,5	2	3	4
35	20192343	SHEFILLA RAISYA R	2,5	2,5	3,75	2,5	2,75	3,75	2,75	4
36	20192345	MUHAMMAD REZA RIZKI	2	3,5	3,5	2,5	3	3,75	3,5	2,75
37	20192347	KEMAL MUHAMMAD HERD	3,75	2	2,5	2	3,5	2,75	4	2,75
38	20192349	PIKRI NAZIF KHAIRLINNAS	3	2,75	3,5	3,5	3	4	2,5	2,5
39	20192351	DAFFA RIZA MULIYA	2,5	3,75	3,5	2,75	4	2	3,5	3,5

Gambar 3. Data *Dummy* Hasil Fungsi *RANDBETWEEN*

3.2. Klaterisasi Data Menggunakan K-Means

Sebelum menggunakan data *dummy* yang belum memiliki target untuk proses klasifikasi, langkah penting yang perlu dilakukan adalah membuat target menggunakan algoritma klastering K-means. Algoritma K-means dapat digunakan untuk mengelompokkan data menjadi beberapa klaster berdasarkan kemiripan karakteristiknya. Penerapan algoritma K-means dapat menentukan pusat klaster dan menentukan objek data mana yang termasuk ke dalam setiap klaster, jumlah klaster yang dibuat sesuai dengan dijelaskan sebelumnya adalah empat. Setelah itu setiap klaster dapat digunakan sebagai kelas atau target yang akan digunakan dalam proses klasifikasi. Proses penerapan algoritma K-Means dapat dilihat pada gambar 4 sampai gambar 9.

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler

In [2]: dataset=pd.read_csv('dataset.csv')
dataset

Out[2]:
```

	SDA	GIS	PTB	PWEB	SIM	ERP	BI	APM
0	2.75	2.75	3.00	2.75	3.00	3.50	3.75	2.00
1	3.50	3.50	2.50	3.75	2.00	2.50	2.00	3.50
2	3.50	4.00	3.75	3.00	3.50	3.00	3.75	3.75
3	2.00	3.75	2.00	3.75	2.00	2.75	4.00	2.75
4	3.00	2.50	4.00	2.50	2.75	2.75	3.75	3.75

150 rows x 9 columns

Gambar 4. Import library dan dataset

```
In [5]: dataset_new = dataset.iloc[:, 0:8]
dataset_new.head()

Out[5]:
```

	SDA	GIS	PTB	PWEB	SIM	ERP	BI	APM
0	2.75	2.75	3.00	2.75	3.00	3.50	3.75	2.00
1	3.50	3.50	2.50	3.75	2.00	2.50	2.00	3.50
2	3.50	4.00	3.75	3.00	3.50	3.00	3.75	3.75
3	2.00	3.75	2.00	3.75	2.00	2.75	4.00	2.75
4	3.00	2.50	4.00	2.50	2.75	2.75	3.75	3.75

Gambar 5. Proses seleksi data yang akan digunakan untuk proses klasterisasi

```
In [7]: array_x = np.array(dataset_new)
array_x

Out[7]: array([[2.75, 2.75, 3. , ..., 3.5, 3.75, 2. ],
[3.5, 3.5, 2.5, ..., 2.5, 2. , 3.5 ],
[3.5, 4. , 3.75, ..., 3. , 3.75, 3.75],
...,
[2.5, 3.5, 3. , ..., 2.75, 4. , 2.75],
[3. , 2.75, 3.5, ..., 2.75, 2.5, 2. ],
[2.75, 2. , 3.75, ..., 3.5, 2.75, 2. ]])
```

Gambar 6. Konversi dataset ke dalam bentuk array

Pada Gambar 6 dan 7 dilakukan konversi dataset yang telah diseleksi menjadi bentuk array dan dilakukan proses *Scaling*. Selanjutnya data hasil proses *Scaling* dilakukan klasterisasi seperti pada Gambar 8 dan hasil proses klasterisasi dapat dilihat pada Gambar 9.

```
In [8]: scaler_fit = MinMaxScaler()
scaled_x = scaler_fit.fit_transform(array_x)
scaled_x

Out[8]: array([[0.375, 0.375, 0.5, ..., 0.75, 0.875, 0. ],
[0.75, 0.75, 0.25, ..., 0.25, 0. , 0.75 ],
[0.75, 1. , 0.875, ..., 0.5, 0.875, 0.875],
...,
[0.25, 0.75, 0.5, ..., 0.375, 1. , 0.375],
[0.5, 0.375, 0.75, ..., 0.375, 0.25, 0. ],
[0.375, 0. , 0.875, ..., 0.75, 0.375, 0. ]])
```

Gambar 7. Proses scaling data sebelum proses Klasterisasi

```
In [12]: dataset["kluster"] = kmeans.labels_

In [13]: print(kmeans.labels_)

[2 1 3 0 0 2 1 2 2 0 2 0 1 2 2 0 1 1 1 0 3 1 3 1 3 2 0 1 3 1 3 0 2 2 1 2 2
1 1 2 2 2 1 1 2 0 1 0 2 1 0 2 0 2 1 2 3 3 1 2 2 1 1 0 2 0 3 0 0 1 1 2 2 1
2 3 0 3 2 3 3 1 2 2 2 3 0 0 0 0 3 1 3 0 2 3 1 2 2 1 3 0 0 1 2 3 3 0 3
1 0 3 3 1 3 3 1 3 3 2 0 0 0 2 0 3 0 1 3 0 3 0 1 1 1 2 1 3 1 3 0 0 1 3 1 0
2 2]
```

Gambar 8. Proses Klasterisasi menggunakan K-Means

```
kmeans = KMeans(n_clusters = 4, random_state=123)
kmeans.fit(scaled_x)

C:\Users\hanif\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of 'n_init' will change from 10 to 'auto' in 1.4. Set the value of 'n_init' explicitly to suppress the warning
  warnings.warn(
C:\Users\hanif\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(

KMeans
KMeans(n_clusters=4, random_state=123)

print(kmeans.cluster_centers_)

[[0.40540541 0.4527027 0.56756757 0.62162162 0.37162162 0.29854054
0.86480486 0.74662162]
[0.75961538 0.63461538 0.30769231 0.49038462 0.49679487 0.625
0.40705128 0.76692564]
[0.465625 0.40625 0.625 0.628125 0.5375 0.33125
0.340625 0.18125 ]
[0.39705882 0.57352941 0.8125 0.39705882 0.79044118 0.73529412
0.59550824 0.76470586]]
```

Gambar 9. Hasil Klusterisasi

Setelah proses K-means selesai, kelas target yang dihasilkan berbentuk angka yang mewakili klaster-klaster yang terbentuk. Namun, untuk keperluan analisis dan interpretasi yang lebih mudah, langkah selanjutnya adalah mengkonversi angka-angka tersebut menjadi kelas yang diinginkan. Dalam kasus ini, kelas yang diinginkan meliputi ERP, EA, BI, dan SIG. Konversi dilakukan dengan cara menghubungkan angka-angka yang ada dengan kelas-kelas yang relevan. Misalnya, jika angka 1 diwakili oleh kelas ERP, angka 2 oleh EA, angka 3 oleh BI, dan angka 4 oleh SIG. Dengan demikian, hasil akhir dari proses ini akan memberikan interpretasi yang lebih jelas dan dapat dengan mudah diidentifikasi sebagai salah satu dari kelas ERP, EA, BI, atau SIG.

3.3. Pembuatan Model dan Prediksi Data Menggunakan NBC

Sebelum mengimplementasikan *machine learning*, langkah yang penting adalah membangun model terhadap dataset yang dimiliki. Model ini akan menjadi dasar untuk melakukan prediksi atau klasifikasi pada data baru yang masuk. Untuk membangun model, beberapa tahapan perlu dilakukan. Pertama, kita perlu melakukan pemrosesan data seperti membersihkan data, menangani nilai yang hilang, atau melakukan normalisasi. Selanjutnya, kita akan membagi dataset menjadi data pelatihan (training data) dan data pengujian (testing data) untuk menguji kinerja model. Setelah itu, kita memilih algoritma yang diinginkan, pada kasus ini algoritma yang digunakan adalah *naive bayes*. Selanjutnya, kita melatih

model dengan menggunakan data pelatihan dan melakukan evaluasi untuk memastikan kinerja model yang optimal. Setelah model dibangun dan diuji dengan baik, maka pengimplementasikan *machine learning* untuk melakukan prediksi atau klasifikasi pada data baru yang masuk dapat dilakukan. Proses pembuatan model dan prediksi data dapat dilihat pada Gambar 10 sampai 14.

```
In [1]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report

In [2]: df = pd.read_csv('datacluster.csv')
df

Out[2]:
```

	SDA	GIS	PTB	PWEB	SIM	ERP	BI	APM	Kluster
0	2.0	3.75	2.00	3.75	2.00	2.75	4.00	2.75	Business Intelligence
1	3.0	2.50	4.00	2.50	2.75	2.75	3.75	3.75	Business Intelligence
2	2.5	3.75	3.75	3.75	2.75	2.00	3.75	4.00	Business Intelligence
3	2.0	3.00	2.00	3.75	3.50	2.00	4.00	4.00	Business Intelligence
4	3.0	2.75	3.50	4.00	4.00	2.50	3.75	4.00	Business Intelligence
...
145	3.5	2.50	4.00	2.75	3.50	3.50	2.00	3.75	Enterprise Resource Planning
146	3.5	2.75	3.75	2.00	3.50	4.00	4.00	3.75	Enterprise Resource Planning
147	2.0	3.00	4.00	4.00	4.00	3.75	2.50	3.50	Enterprise Resource Planning
148	2.0	3.75	3.50	2.00	3.75	2.50	3.75	4.00	Enterprise Resource Planning
149	2.5	3.00	4.00	4.00	2.75	4.00	2.75	3.50	Enterprise Resource Planning

150 rows x 9 columns

Gambar 10. Import library dan dataset

```
In [6]: y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
Business Intelligence	0.86	0.67	0.75	9
Enterprise Architecture	1.00	1.00	1.00	6
Enterprise Resource Planning	0.80	1.00	0.89	8
Sistem Informasi Geografis	0.71	0.71	0.71	7
accuracy			0.83	30
macro avg	0.84	0.85	0.84	30
weighted avg	0.84	0.83	0.83	30

Gambar 14. Evaluasi model menggunakan *Confussion Matrix*

Penerapan metode *Naive Bayes* pada *dataset* yang terdiri dari 150 data dengan pembagian data *testing* sebesar 20% dan data *training* sebesar 80% menghasilkan akurasi sebesar 83% berdasarkan *confusion matrix*. *Confusion matrix* adalah sebuah matriks yang digunakan untuk mengukur performa model klasifikasi dengan membandingkan hasil prediksi dengan label sebenarnya. Dalam kasus ini, dengan menggunakan *Naive Bayes*, model berhasil mengklasifikasikan data dengan tingkat akurasi sebesar 83%. *Confusion matrix* memberikan gambaran lebih jelas tentang performa model dengan menyajikan jumlah data yang terklasifikasikan dengan benar (*true positives* dan *true negatives*) serta data yang salah terklasifikasikan (*false positives* dan *false negatives*). Dengan hasil akurasi yang cukup tinggi dan informasi dari *confusion matrix*, kita dapat mengevaluasi kinerja model dan membuat penyesuaian jika diperlukan untuk meningkatkan kemampuan prediksi atau klasifikasi pada data yang belum terlihat sebelumnya.

3.4. Implementasi Model Berbasis Website

Pengimplementasian model dilakukan dengan cara menerapkan model yang telah dibuat sebelumnya ke dalam Streamlit. Streamlit adalah sebuah framework open-source yang digunakan untuk membangun antarmuka pengguna interaktif untuk aplikasi data dan machine learning. Kode yang digunakan kurang lebih

```
In [3]: X = df.drop(columns="Kluster")
y = df.Kluster
```

Gambar 11. Proses seleksi data yang akan digunakan untuk proses klasifikasi

```
In [4]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train.shape, X_test.shape, y_train.shape, y_test.shape

Out[4]: ((120, 8), (30, 8), (120,), (30,))
```

Gambar 12. Pembagian data *training* dan data *testing*

```
In [5]: clf = GaussianNB()
clf.fit(X_train, y_train)

Out[5]: GaussianNB()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

Gambar 13. Proses Klasifikasi menggunakan *Naive Bayes*

sama dengan yang digunakan sebelumnya pada model, hanya saya ada beberapa tambahan kode yang berfungsi untuk menampilkan rekomendasi dosen pembimbing dan juga topik-topik tugas akhir yang relevan dengan topik yang disarankan. Kode dan interface dari sistem yang dibangun dapat dilihat pada Gambar 15 sampai 17.

```
1 #import library
2 import streamlit as st
3 import pandas as pd
4 import webbrowser as wb
5 from sklearn.naive_bayes import GaussianNB
6 from PIL import Image
7
```

Gambar 15. Kode Import library

```
#header website
st.write("""
# Web Klasifikasi Topik Tugas Akhir Mahasiswa DST UNAND
Aplikasi Berbasis Web untuk Mengklasifikasikan **Topik Tugas Akhir** Mahasiswa Departemen Sistem Informasi UNAND
""")

#image website
img = Image.open('topik.png')
st.image(img, width=1000)
```

Gambar 16. Kode header dan gambar header

```

18 #sidebar
19 st.sidebar.header('Inputkan Nilai Anda')
20 #inputan data tester
21 def input_user():
22     sda = st.sidebar.slider('Nilai Struktur Data & Algoritma', 2.0, 4.0, 2.5)
23     gis = st.sidebar.slider('Nilai Sistem Informasi Geografis', 2.0, 4.0, 2.5)
24     ptb = st.sidebar.slider('Nilai Pemrograman Teknologi Bergerak', 2.0, 4.0, 2.5)
25     pweb = st.sidebar.slider('Nilai Pemrograman Web', 2.0, 4.0, 2.5)
26     sim = st.sidebar.slider('Nilai Sistem Informasi Manajemen', 2.0, 4.0, 2.5)
27     erp = st.sidebar.slider('Nilai Enterprise Resource Planning', 2.0, 4.0, 2.5)
28     bi = st.sidebar.slider('Nilai Business Intelligence', 2.0, 4.0, 2.5)
29     apm = st.sidebar.slider('Nilai Aplikasi Pembelajaran Mesin', 2.0, 4.0, 2.5)
30     data = {'SDA': sda,
31            'GIS': gis,
32            'PTB': ptb,
33            'PWEB': pweb,
34            'SIM': sim,
35            'ERP': erp,
36            'BI': bi,
37            'APM': apm}
38     fitur = pd.DataFrame(data, index=[0])
39     return fitur
40 df = input_user()
    
```

Gambar 17. Kode inputan data

```

46 #open data train
47 dataset=pd.read_csv('dataset.csv')
48
49 #declare feature and target (train&test)
50 x = dataset.values[:,0:8]
51 y = dataset.values[:,8]
52
53 #NBC
54 model = GaussianNB()
55 model.fit(x,y)
    
```

Gambar 18. Kode import dataset dan algoritma naive bayes

Pada Gambar 18 dapat dilihat kode untuk import dataset dan menjalankan fungsi untuk algoritma naive bayes. Hasil rekomendasi dosen dan topik tugas akhir ditampilkan dengan menggunakan kode seperti yang terlihat pada gambar 19, gambar 20 dan gambar 21. Tampilan aplikasi web dari sistem rekomendasi topik tugas akhir ini dapat dilihat pada Gambar 22.

```

57 #predict
58 prediksi = model.predict(df)
59 prediksi_proba = model.predict_proba(df)
60
61 #showpredict
62 st.subheader('Rekomendasi Topik Tugas Akhir')
63 st.write(prediksi)
64
    
```

Gambar 19. Kode Predict dan menampilkan rekomendasi

```

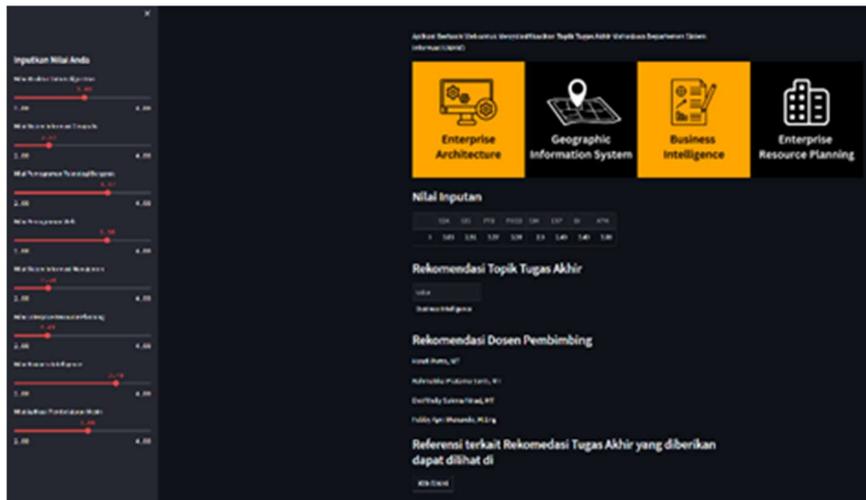
65 #dosen
66 if (prediksi=="Business Intelligence"):
67     st.subheader('Rekomendasi Dosen Pembimbing')
68     st.write("""Hasdi Putra, MT """)
69     st.write("""Rahmatika Pratama Santi, MT""")
70     st.write("""Dwi Welly Sukma Nirad, MT """)
71     st.write("""Febby Apri Wenando, M.Eng """)
72 elif (prediksi=="Sistem Informasi Geografis"):
73     st.subheader('Rekomendasi Dosen Pembimbing')
74     st.write("""Haris Suryamen, M.Sc""")
75     st.write("""Ully Mega Wahyuni, M.Kom""")
76     st.write("""Hafizah Hanim, M.Kom""")
77     st.write("""Fajril Akbar, M.Sc""")
78 elif (prediksi=="Enterprise Architecture"):
79     st.subheader('Rekomendasi Dosen Pembimbing')
80     st.write("""Prof. Surya Afnarius. Ph.D""")
81     st.write("""Husnil Kamil, MT """)
82     st.write("""Aina Hubby Aziira, M.Eng """)
83     st.write("""Adi Arga Arifnur, M.Kom """)
84 else :
85     st.subheader('Rekomendasi Dosen Pembimbing')
86     st.write("""Ricky Akbar, M.Kom """)
87     st.write("""Jepril Rahmadoni, M.Kom """)
88     st.write("""Afriyanti Dwi Kartika, MT""")
    
```

Gambar 20. Kode menampilkan rekomendasi dosen

```

#Referensi
url = 'http://scholar.unand.ac.id/'
st.subheader('Referensi terkait Rekomedasi Tugas Akhir yang diberikan dapat dilihat di')
if st.button('Klik Disini'):
    wb.open_new_tab(url)
    
```

Gambar 21. Kode menampilkan rekomendasi topik tugas akhir yang relevan



Gambar 22. Interface Sistem Rekomendasi Topik Tugas Akhir untuk Mahasiswa DSI Unand

Setelah berhasil mengembangkan aplikasi menggunakan Streamlit, langkah selanjutnya adalah meng-hosting aplikasi tersebut di Streamlit Share. Streamlit Share adalah platform yang disediakan oleh Streamlit untuk memudahkan pengguna dalam menyebarkan dan berbagi aplikasi Streamlit secara online. Dengan meng-hosting aplikasi di Streamlit Share, aplikasi yang telah dibuat sebelumnya dapat diakses oleh pengguna lain melalui

web. Hal ini memungkinkan untuk menyebarkan sistem ini ke masyarakat luas.

4. KESIMPULAN

Berdasarkan pembahasan yang telah dilakukan maka dapat diambil kesimpulan yaitu algoritma *Naive Bayes Classifier* mampu melakukan proses klasifikasi tugas akhir secara otomatis. Algoritma *Naive Bayes Classifier* memiliki kinerja yang baik dalam klasifikasi tugas akhir ditunjukkan dengan hasil pengujian berdasarkan parameter nilai mahasiswa pada mata kuliah Struktur Data Algoritma (SDA), Sistem Informasi Geografis (SIG), Pemrograman Teknologi Bergerak (PTB), Pemrograman Web (PWeb), Sistem Informasi Manajemen (SIM), *Enterprise Resource Planning* (ERP), *Business Intelligence* (BI), dan *Machine Learning* (ML) menyatakan algoritma *Naive Bayes* memiliki performa dengan hasil Accuracy, Recall, Precision, F1-score dengan nilai 83%, 185%, 84% dan 84%. Penelitian ini dapat memberikan rekomendasi untuk topik tugas akhir bagi mahasiswa.

DAFTAR PUSTAKA

- [1] D. O. Yusra and Y. Vitriani, "Perbandingan Klasifikasi Tugas Akhir Mahasiswa Jurusan Teknik Informatika Menggunakan Metode *Naive Bayes Classifier* dan *K-Nearest Neighbor*," *J. Sains, Teknol. dan Ind*, vol. 14, no. 1, pp. 79–85, 2016.
- [2] P. E. Mas'udia, "Klasifikasi Tugas Akhir Untuk Menentukan Dosen Pembimbing Menggunakan *Naive Bayes Classifier* (NBC)," *Politeknik Negeri Malang*, 2015.
- [3] R. M. Hariyati, "Survey kinerja dosen pembimbing skripsi dan kualitas skripsi mahasiswa akuntansi stie malangkucewara," *Jurnal Dinamika Akuntansi*, vol. 4, no. 2, 2012.
- [4] P. Kasih, "Integrasi Kategori Skripsi Dan Keahlian Dosen Dalam *Naive Bayes* Untuk Pemilihan Dosen Pembimbing," *Nusantara of Engineering (NOE)*, vol. 3, no. 2, pp. 34–42, 2016.
- [5] N. SA and F. Ismail, "Automatic classifications of Malay proverbs using *Naive Bayesian* algorithm," 2008.
- [6] T. Angelya, A. Rahman, and I. Pradesan, "Implementasi Sistem Klasifikasi Tim Kerja Menggunakan Algoritma *Naive Bayes* (Studi Kasus: PT. MNL)," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, no. 2, pp. 86–94, 2023.
- [7] A. F. Hidayatullah and M. R. Ma'arif, "Penerapan Text Mining dalam Klasifikasi Judul Skripsi," presented at the Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2016.
- [8] M. Ikhsan, "Penerapan Text Mining Pada Sistem Rekomendasi Pembimbing Skripsi Mahasiswa Menggunakan Algoritma *Naive Bayes Classifier* di Program Studi Ilmu Komputer UIN Sumatera Utara Medan," *The Indonesian Journal of Computer Science*, vol. 12, no. 6, 2023.
- [9] S. Sukriadi, I. Ismail, and A. M. Andzar, "Penerapan text mining dalam klasifikasi judul skripsi yang diusulkan mahasiswa menggunakan metode *naive Bayes*," *Jurnal Ilmiah Sistem Informasi Dan Teknik Informatika (JISTI)*, vol. 6, no. 2, pp. 184–196, 2023.
- [10] D. A. U. Zaza, E. Umar, and F. E. O. Sanga, "Penerapan Text Mining Dalam Klasifikasi Judul Skripsi Mahasiswa Pada Universitas Stella Maris Sumba Menggunakan Metode *Naive Bayes*," *Journal Of Informatics And Business*, vol. 2, no. 3, pp. 327–337, 2024.
- [11] Y. D. Pramudita, S. S. Putro, and N. Makhmud, "Klasifikasi Berita Olahraga Menggunakan Metode *Naive*

Bayes dengan *Enhanced Confix Stripping Stemmer*," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 3, pp. 269–276, 2018.

- [12] R. N. Devita, H. W. Herwanto, and A. P. Wibawa, "Perbandingan kinerja metode *naive bayes* dan *k-nearest neighbor* untuk klasifikasi artikel berbahasa indonesia," *J. Teknol. Inf. dan Ilmu Komput*, vol. 5, no. 4, 2018.
- [13] E. Prasetyo, "Data mining konsep dan aplikasi menggunakan matlab," *Yogyakarta: Andi*, vol. 1, 2012.
- [14] A. Basuki, "Customer Classification using Learning Vector Quantization Neural Network," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 8, no. 5, pp. 131–135, 2016.
- [15] L. Zuo and J. Guo, "Customer classification of discrete data concerning customer assets based on data mining," presented at the 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), IEEE, 2019, pp. 352–355.
- [16] T. K. Das, "A customer classification prediction model based on machine learning techniques," presented at the 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), IEEE, 2015, pp. 321–326.
- [17] S. Ting, W. Ip, and A. H. Tsang, "Is *Naive Bayes* a good classifier for document classification," *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37–46, 2011.

BIODATA PENULIS



Hanif Izza Pratama
Mahasiswa Jurusan Sistem Informasi
Universitas Andalas angkatan tahun 2020.
Berasal dari Kota Solok, Sumatera Barat,
Indonesia.



Siti Nur Aisah
Mahasiswa Jurusan Sistem Informasi
Universitas Andalas angkatan tahun 2020.
Berasal dari Pasaman Barat, Sumatera Barat,
Indonesia.