
**PERBANDINGAN KINERJA ALGORITMA MACHINE LEARNING
DALAM MEMPREDIKSI TINGKAT STRES MAHASISWA
BERDASARKAN FAKTOR AKADEMIK
DAN NON-AKADEMIK**

**Ahmad Jihad Al Fayed¹, Surya Darma², Muhammad Hizbul Aqsha³,
Surya Maruli P Pardede⁴, Muhammad Amin⁵**

^{1,3,4,5}Universitas Pembangunan Panca Budi Medan, Sumatera Utara

²Universitas Potensi Utama Medan, Sumatera Utara

e-mail: ¹jihadahmad000@gmail.com, ²suryadarma766@gmail.com,

³hizbul2000@gmail.com, ⁴sp.trainer.belajar.id@gmail.com, ⁵mhdamin9977@gmail.com

Abstract: *Stress among students is a growing phenomenon due to high academic demands, changes in the social environment, and various non-academic pressures faced during their studies. Stress that is not managed properly can have a negative impact on students' mental health, motivation to study, and academic achievement. Therefore, an approach is needed that can identify and predict students' stress levels objectively and based on data. This study aims to analyze and compare the performance of several machine learning algorithms in predicting student stress levels based on academic and non-academic factors. The dataset used in this study is Student Stress Factors, which includes various variables such as Sleep Quality, Academic Achievement, Study Load, Frequency of Headaches, Extracurricular Activities, Level of Social Support, Screen Time, etc. The algorithms applied are Support Vector Machine and Naive Bayes. This research is expected to contribute to the development of a decision support system for early detection of student stress levels, as well as serve as a reference for educational institutions in designing strategies for the prevention and management of mental health issues in higher education environments.*

Keywords: *Machine Learning, Support Vector Machine, Naive Bayes, Stress, Student*

Abstrak: Stres pada mahasiswa merupakan fenomena yang semakin meningkat seiring dengan tuntutan akademik yang tinggi, perubahan lingkungan sosial, serta berbagai tekanan non-akademik yang dihadapi selama masa studi. Kondisi stres yang tidak dikelola dengan baik dapat berdampak negatif terhadap kesehatan mental, motivasi belajar, serta capaian akademik mahasiswa. Oleh karena itu, diperlukan suatu pendekatan yang mampu mengidentifikasi dan memprediksi tingkat stres mahasiswa secara objektif dan berbasis data. Penelitian ini bertujuan untuk menganalisis serta membandingkan kinerja algoritma machine learning dalam memprediksi tingkat stres mahasiswa berdasarkan faktor akademik dan non-akademik. Dataset yang digunakan pada penelitian ini adalah Student Stress Factors, yang mencakup berbagai variabel seperti Kualitas Tidur, Prestasi Akademik, Beban Studi, Frekuensi Sakit Kepala, Kegiatan Ekstrakurikuler, Tingkat Dukungan Sosial, Jam Waktu Layar, dll. Algoritma yang diterapkan yaitu Support Vector Machine dan Naive Bayes dengan akurasi tertinggi dihasilkan oleh Algoritma SVM dengan akurasi 85% sedangkan NV memiliki akurasi 83%. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem pendukung keputusan untuk deteksi dini tingkat stres mahasiswa, serta menjadi referensi bagi institusi pendidikan dalam merancang strategi pencegahan dan penanganan masalah kesehatan mental di lingkungan perguruan tinggi.

Kata kunci: Machine Learning, Support Vector Machine, Naive Bayes, Stres, Mahasiswa

PENDAHULUAN

Mahasiswa merupakan kelompok yang berada pada fase transisi penting dalam kehidupan, di mana mereka dituntut untuk beradaptasi dengan lingkungan akademik, sosial, dan personal secara bersamaan (Rizki et al., 2025). Berbagai tuntutan tersebut sering kali memicu munculnya stres, terutama ketika mahasiswa harus menghadapi beban akademik yang tinggi, tekanan ujian, keterbatasan waktu, serta permasalahan non-akademik seperti kondisi finansial dan hubungan sosial (Nyoman et al., 2018).

Stres yang berlangsung dalam jangka waktu panjang dan tidak ditangani dengan baik dapat berdampak pada menurunnya konsentrasi belajar, motivasi akademik, hingga gangguan kesehatan mental (Parraga-alava, 2024).

Dalam konteks pendidikan tinggi, permasalahan stres mahasiswa menjadi isu yang semakin mendapat perhatian. Perguruan tinggi tidak hanya dituntut untuk menghasilkan lulusan yang unggul secara akademik, tetapi juga memiliki tanggung jawab dalam menjaga kesejahteraan mental mahasiswa (Dhinora & Mailoa, 2025).

Namun demikian, identifikasi tingkat stres mahasiswa masih sering dilakukan secara subjektif melalui survei manual atau konseling individual, yang membutuhkan waktu dan sumber daya yang tidak sedikit (Luci et al., 2025). Oleh karena itu, diperlukan pendekatan alternatif yang lebih efisien, objektif, dan berbasis data (Tan et al., 2024).

Perkembangan teknologi informasi, khususnya dalam bidang machine learning, membuka peluang baru dalam analisis permasalahan kesehatan mental, termasuk stres mahasiswa (Sasane, S., & Mulla, Z. A. S., 2024). *Machine learning* mampu mempelajari pola dari data historis dan menghasilkan prediksi yang akurat berdasarkan variabel-variabel yang relevan.

Berbagai penelitian sebelumnya

menunjukkan bahwa algoritma *machine learning* dapat digunakan untuk klasifikasi dan prediksi kondisi psikologis dengan tingkat akurasi yang cukup baik (Adie et al., 2025). Meskipun demikian, setiap algoritma memiliki karakteristik dan performa yang berbeda-beda tergantung pada jenis data dan permasalahan yang dihadapi (Chowdhury et al., 2024).

Berdasarkan hal tersebut, penelitian ini memfokuskan pada analisis perbandingan kinerja beberapa algoritma *machine learning* (Amin et al., 2024) dalam memprediksi tingkat stres mahasiswa.

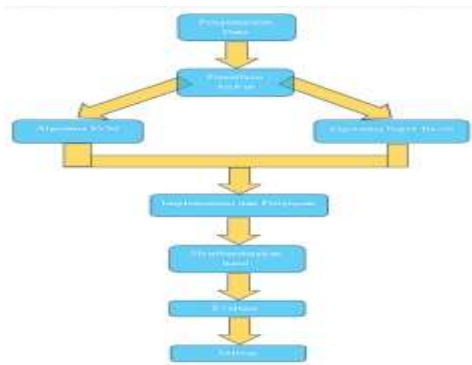
Algoritma yang digunakan meliputi *Support Vector Machine* dan *Naive Bayes* (Novelan & Iqbal, 2025) dengan menggunakan *Hyperparameter Tuning* Berbasis *GridSearchCV*, yang merupakan algoritma populer dan banyak digunakan dalam penelitian klasifikasi. Dataset yang digunakan mencakup faktor akademik dan non-akademik, sehingga diharapkan mampu merepresentasikan kondisi stres mahasiswa secara lebih komprehensif (Fatah & Hasanah, 2025).

Melalui penelitian ini, diharapkan dapat diperoleh gambaran mengenai algoritma machine learning yang paling efektif dalam memprediksi tingkat stres mahasiswa. Hasil penelitian diharapkan tidak hanya memberikan kontribusi akademik dalam bidang data mining dan machine learning, tetapi juga dapat dimanfaatkan sebagai dasar pengembangan sistem deteksi dini stres mahasiswa di lingkungan perguruan tinggi.

Dengan demikian, institusi pendidikan dapat mengambil langkah preventif yang lebih tepat dalam menjaga kesehatan mental mahasiswa.

METODE

Penelitian ini memiliki tahapan yang dapat dilihat pada gambar di bawah ini.



Gambar 1 Metode dan Tahapan Penelitian

Dalam pengembangan metode terdiri dari beberapa tahapan penelitian, yaitu:

Pengumpulan Data

Dalam tahap ini peneliti mengumpulkan dataset *Student Stress Factors* melalui web *Kaggle.com* dan juga melakukan studi literatur terhadap artikel jurnal penulis lain yang berkaitan. Dataset *Student Stress Factors* memiliki 800 data dan 13 atribut parameter yang relevan untuk di uji dengan kedua algoritma yang di pakai untuk penelitian ini.

Penentuan Atribut (*Feature Selection*)

Dataset penelitian ini memiliki atribut variabel seperti *Sleep Quality*, *Academic Performance*, *Study Load*, *Headache Frequency*, *Extracurricular Activity*, *Social Support Level*, *Screen Time Hours*, *Physical Activity Min*, *Commute Time Min*, *Financial Pressure*, *Academic Workload*, *Part Time Job*, *Stress Level*. D

Metode Algoritma *Support Vector Machine* (SVM)

Algoritma SVM Ini dianggap sebagai algoritma klasifikasi yang menggabungkan teknik pembelajaran mesin, termasuk Random Forest, untuk menciptakan model yang tangguh yang umumnya digunakan dalam masalah klasifikasi (Sharma, 2023)(Noi & Kappas, 2018). Representasi matematisnya sebagai berikut :

$$\text{Si } Y_i = +1; wx_i + b \geq 1 \quad (1)$$

$$\text{Si } Y_i = -1; wx_i + b \leq -1 \quad (2)$$

$$i; Y_i(wx_i + b) \geq 1 \quad (3)$$

Persamaan (1) dan (2) menunjukkan kondisi pemisahan data oleh sebuah *hyperplane* yang didefinisikan oleh persamaan $wx_i + b = 0$, di mana:

1. w merupakan vektor bobot (weight vector),
2. x_i merupakan vektor fitur dari data ke- i ,
3. b merupakan nilai bias.

Data dengan label +1 diharapkan berada di satu sisi *hyperplane* dengan jarak minimal tertentu, sedangkan data dengan label -1 berada di sisi yang berlawanan. Nilai batas ± 1 digunakan untuk menentukan *margin*, yaitu jarak antara *hyperplane* dan titik data terdekat dari masing-masing kelas.

Persamaan (3) merupakan bentuk gabungan dari persamaan (1) dan (2) yang menyederhanakan syarat klasifikasi SVM. Persamaan ini menyatakan bahwa setiap data harus diklasifikasikan dengan benar dan berada di luar atau tepat pada batas margin. Jika nilai $Y_i(wx_i + b)$ lebih besar atau sama dengan 1, maka data tersebut diklasifikasikan dengan benar oleh model.

Metode Algoritma *Naïve Bayes*

Naïve Bayes merupakan algoritma probabilistik yang bekerja berdasarkan Teorema Bayes. Algoritma ini mengasumsikan bahwa setiap fitur bersifat independen, sehingga dapat digunakan untuk kategori data besar (Kartianom et al., 2022).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Keterangan:

1. $P(y|X)$ merupakan probabilitas *posterior*, yaitu peluang suatu data dengan fitur X termasuk ke dalam kelas y .
2. $P(X|y)$ merupakan probabilitas *likelihood*, yaitu peluang munculnya

- fitur X pada kelas y.
3. $P(y)$ merupakan probabilitas *prior*, yaitu peluang awal suatu kelas y sebelum mempertimbangkan data X.
 4. $P(X)$ merupakan probabilitas *evidence*, yaitu peluang munculnya fitur X tanpa memperhatikan kelas tertentu.

Implementasi Dan Pengujian

Pada tahap ini, dataset diimplementasikan dan diuji menggunakan bahasa pemrograman Python dengan bantuan platform *Google Colaboratory*. Proses implementasi meliputi tahap pengumpulan data, pembagian data menjadi data latih dan data uji, serta penerapan algoritma *Support Vector Machine (SVM)* dan *Naive Bayes*.

Pengujian dilakukan untuk mengevaluasi kinerja masing-masing algoritma dalam melakukan klasifikasi, dengan menggunakan metrik evaluasi berupa nilai akurasi. Hasil pengujian ini digunakan untuk membandingkan performa kedua algoritma dan menentukan algoritma yang memiliki tingkat akurasi terbaik pada dataset yang digunakan.

Membandingkan Hasil

Pada tahap ini, dilakukan perbandingan kinerja dari masing-masing algoritma yang digunakan. Perbandingan dilakukan berdasarkan nilai evaluasi yang meliputi akurasi, *presisi*, *recall*, dan *F1-score*. Seluruh metrik tersebut diperoleh dari hasil pengolahan data menggunakan bahasa pemrograman Python melalui platform *Google Colaboratory*.

Hasil perbandingan ini bertujuan untuk menentukan algoritma yang memiliki kinerja terbaik dalam melakukan klasifikasi berdasarkan nilai evaluasi yang dihasilkan secara otomatis oleh sistem.

Evaluasi

Tahap ini biasanya melibatkan pengukuran menggunakan metrik statistik seperti:

1. *Akurasi*: Berapa banyak prediksi

yang benar secara keseluruhan.

2. *Precision & Recall*: Ketepatan model dalam mengidentifikasi kategori tertentu.
3. *F1-Score*: Keseimbangan antara *precision* dan *recall*.

1. Selesai

Tahap akhir di mana kesimpulan diambil. Peneliti akan menentukan algoritma mana yang paling direkomendasikan untuk digunakan berdasarkan hasil evaluasi yang paling tinggi.

Analisis Singkat:

Alur ini sangat standar dalam penelitian *Data Science*. Fokus utamanya adalah membuktikan efektivitas metode *SVM vs Naive Bayes*. Biasanya, *SVM* cenderung lebih kuat untuk data yang kompleks, sedangkan *Naive Bayes* unggul dalam kecepatan dan kesederhanaan.

HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk membandingkan kinerja algoritma machine learning menggunakan metode *Support Vector Machine (SVM)* dan *Naive Bayes* dengan memanfaatkan bahasa pemrograman Python. Dataset yang digunakan adalah *Student Stress Factors* yang diperoleh dari situs Kaggle. Dataset tersebut terdiri dari 800 data, yang dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%, serta memiliki 13 atribut yang relevan dengan tujuan penelitian.

1. Pengujian Algoritma *Support Vector Machine (SVM)*

Hasil pengujian dataset menggunakan algoritma *Support Vector Machine (SVM)* yang diimplementasikan dengan bahasa pemrograman Python melalui *Google Colaboratory* menunjukkan nilai akurasi sebesar 82%, *presisi* sebesar 82%, *recall* sebesar 82%, dan *F1-score* sebesar 82%. Implementasi kode Python serta hasil perhitungan

metrik evaluasi tersebut ditampilkan pada Gambar 2, Gambar 3, dan Gambar 4.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.naive_bayes import GaussianNB

# Load dataset
df = pd.read_csv("stresstest_factors_dataset.csv")

# FEATURES & TARGET
X = df.drop("stress_level", axis=1)
y = df["stress_level"]

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42,
                                                    stratify=y)

scaler = StandardScaler()
X_train = pd.DataFrame(scaler.fit_transform(X_train),
                       columns=X.columns)
X_test = pd.DataFrame(scaler.transform(X_test),
                      columns=X.columns)

# MODEL
model = GaussianNB()

# TRAIN & EVALUATION
print("--- CLASSIFICATION REPORTS ---")

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # 1. Print Classification Report to console
    print(f"[{name}] Classification Report:")
    print(classification_report(y_test, y_pred))
    # 2. Print metrics with label percentages
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, average="weighted", zero_division=0)
    rec = recall_score(y_test, y_pred, average="weighted", zero_division=0)
    f1 = f1_score(y_test, y_pred, average="weighted", zero_division=0)

    performance_results.append({
        "model": name,
        "accuracy": acc,
        "precision": prec,
        "recall": rec,
        "f1_score": f1
    })
```

Gambar 2 Implementasi kode Python SVM

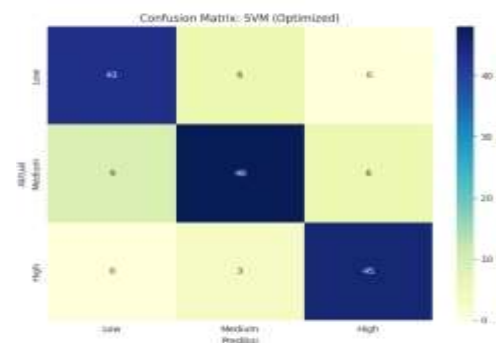
```
===== SVM =====
Accuracy : 0.8187
Precision : 0.8178
Recall : 0.8187
F1-Score : 0.8183

Classification Report:

```

	precision	recall	f1-score	support
0	0.78	0.78	0.78	49
1	0.77	0.76	0.77	63
2	0.92	0.94	0.93	48
accuracy			0.82	160
macro avg	0.82	0.82	0.82	160
weighted avg	0.82	0.82	0.82	160

Gambar 3 Hasil Pengujian Algoritma SVM



Gambar 4 Confusion Matrix Algoritma SVM

Pengujian Algoritma Naïve Bayes

Hasil pengujian dataset menggunakan algoritma *Naïve Bayes* yang diimplementasikan dengan bahasa pemrograman Python melalui *Google Colaboratory* menunjukkan nilai akurasi sebesar 83%, presisi sebesar 83%, *recall* sebesar 83%, dan *F1-score* sebesar 83%. Implementasi kode Python serta hasil perhitungan metrik evaluasi tersebut disajikan pada Gambar 5, Gambar 6, dan Gambar 7.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.naive_bayes import GaussianNB

# Load dataset
df = pd.read_csv("stresstest_factors_dataset.csv")

# FEATURES & TARGET
X = df.drop("stress_level", axis=1)
y = df["stress_level"]

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42,
                                                    stratify=y)

scaler = StandardScaler()
X_train = pd.DataFrame(scaler.fit_transform(X_train),
                       columns=X.columns)
X_test = pd.DataFrame(scaler.transform(X_test),
                      columns=X.columns)

# MODEL
model = GaussianNB()

# TRAIN & EVALUATION
print("--- CLASSIFICATION REPORTS ---")

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # 1. Print Classification Report to console
    print(f"[{name}] Classification Report:")
    print(classification_report(y_test, y_pred))
    # 2. Print metrics with label percentages
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, average="weighted", zero_division=0)
    rec = recall_score(y_test, y_pred, average="weighted", zero_division=0)
    f1 = f1_score(y_test, y_pred, average="weighted", zero_division=0)

    performance_results.append({
        "model": name,
        "accuracy": acc,
        "precision": prec,
        "recall": rec,
        "f1_score": f1
    })
```

Gambar 5 Implementasi kode Python Naïve Bayes

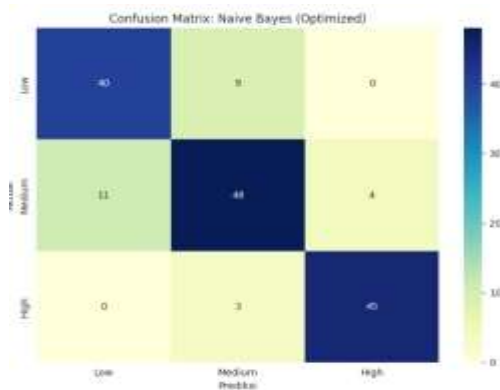
```
===== NAIVE BAYES =====
Accuracy : 0.8313
Precision : 0.8307
Recall : 0.8313
F1-Score : 0.8307

Classification Report:

```

	precision	recall	f1-score	support
0	0.78	0.82	0.80	49
1	0.80	0.76	0.78	63
2	0.92	0.94	0.93	48
accuracy			0.83	160
macro avg	0.83	0.84	0.84	160
weighted avg	0.83	0.83	0.83	160

Gambar 6 Hasil Pengujian Algoritma Naïve Bayes



Gambar 7 Confusion Matrix Algoritma Naïve Bayes

Pengujian Algoritma Support Vector Mechine dan Algoritma Naïve Bayes

Perbandingan performa dilakukan berdasarkan hasil pengujian dengan menggunakan metrik evaluasi berupa akurasi, presisi, recall, dan F1-score dari masing-masing algoritma yang dibandingkan. Perbandingan ini bertujuan untuk menentukan algoritma yang paling

tepat dan sesuai untuk diterapkan pada dataset yang digunakan. Implementasi kode bahasa pemrograman Python serta hasil perbandingan disajikan pada Gambar 8, Tabel 1, dan Gambar 9.

```

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report

# ... (code for loading data and training models) ...

# Evaluasi hasil model ...
plt.figure(figsize=(10, 7))

# Transformasi data agar mudah di plot oleh seaborn (file dataframe)
df_result = df_result.melt(id_vars='Model', var_name='Metric', value_name='Score')

sns.set_style('whitegrid')
ax = sns.barplot(data=df_result, x='Model', y='Score', hue='Metric', palette='magma')

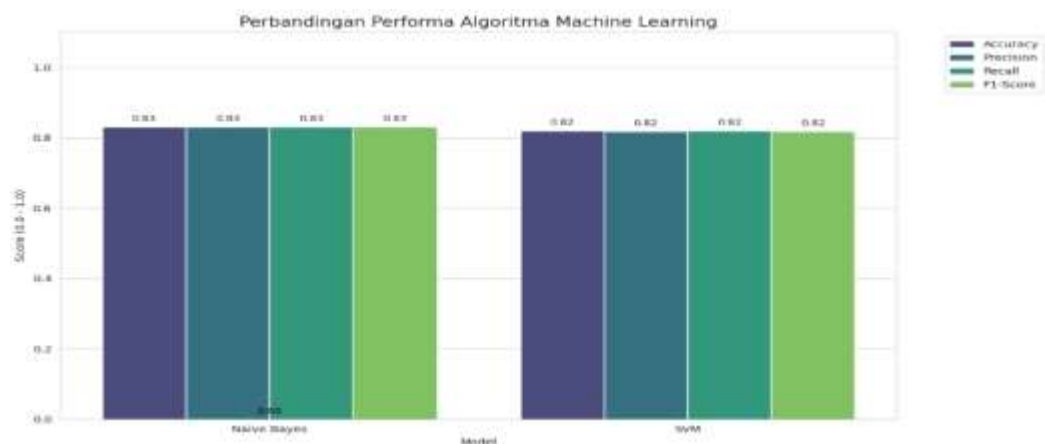
plt.title('Perbandingan Performa Algoritma Machine Learning', fontsize=11)
plt.xlabel('Model') # Nama kelas
plt.ylabel('Score (0.0 - 1.0)')
plt.legend(['Acc', 'Pre', 'Rec', 'F1'])

# Menambahkan angka di atas bar
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='bottom',
                xytext=(0, 0),
                textcoords='offset points',
                fontsize=10)

plt.tight_layout()
plt.show()
    
```

Gambar 8 Implementasi kode Python Perbandingan dan Diagram Algoritma

No.	Model	Accuracy	Precision	Recall	F1-Score
1	Naïve Bayes	0.83125	0.830706	0.83125	0.830668
2	SVM	0.81875	0.817849	0.81875	0.818251



Gambar 9 Diagram Perbandingan Performa Algoritma Machine Learning

Berdasarkan Tabel 3 dan Gambar 9, hasil pengujian menunjukkan bahwa algoritma Naïve Bayes memperoleh nilai akurasi sebesar 83%, presisi sebesar 83%, recall sebesar 83%, dan F1-score sebesar 83%. Sementara itu, algoritma Support

Vector Machine (SVM) memperoleh nilai akurasi sebesar 82%, presisi sebesar 82%, recall sebesar 82%, dan F1-score sebesar 82%.

Perbedaan nilai performa antara kedua algoritma relatif kecil, namun hasil

tersebut mengindikasikan bahwa *Naïve Bayes* memiliki kinerja yang sedikit lebih unggul dibandingkan SVM dalam konteks klasifikasi pada dataset *Student Stress Factors*. Konsistensi nilai pada setiap metrik menunjukkan bahwa model *Naïve Bayes* mampu menjaga keseimbangan antara ketepatan prediksi dan kemampuan dalam mengenali data secara menyeluruh.

Sementara itu, algoritma SVM juga menunjukkan performa yang cukup baik dengan nilai yang stabil pada seluruh metrik evaluasi. Hal ini menandakan bahwa SVM tetap memiliki kemampuan klasifikasi yang andal, meskipun secara numerik masih berada di bawah *Naïve Bayes*. Selisih performa yang tipis ini menunjukkan bahwa kedua algoritma sama-sama layak digunakan, namun *Naïve Bayes* lebih sesuai diterapkan pada dataset yang digunakan dalam penelitian ini.

Secara keseluruhan, hasil visualisasi ini memperkuat temuan bahwa algoritma *Naïve Bayes* memberikan performa yang lebih optimal dalam memprediksi tingkat stres mahasiswa berdasarkan faktor akademik dan non-akademik, meskipun peningkatan yang diperoleh dibandingkan SVM tidak terlalu signifikan.

SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma machine learning pada dataset *Student Stress Factors* menunjukkan perbedaan kinerja antara algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM). Dengan pembagian data latih sebesar 20%, algoritma *Naïve Bayes* menghasilkan nilai akurasi, presisi, *recall*, dan *F1-score* masing-masing sebesar 83%.

Pada dataset dan skenario pengujian yang sama, algoritma SVM memperoleh nilai akurasi, presisi, *recall*, dan *F1-score* masing-masing sebesar 82%. Meskipun perbedaan nilai yang dihasilkan relatif kecil, hasil pengujian

menunjukkan bahwa algoritma *Naïve Bayes* memiliki performa yang sedikit lebih unggul dibandingkan dengan SVM.

Keunggulan tersebut menunjukkan bahwa algoritma *Naïve Bayes* lebih efektif dalam memprediksi tingkat stres mahasiswa berdasarkan faktor akademik dan non-akademik pada dataset yang digunakan. Hal ini diduga karena karakteristik *Naïve Bayes* yang mampu menangani data dengan distribusi fitur yang sederhana dan bersifat probabilistik secara efisien.

Penelitian ini diharapkan dapat menjadi bahan referensi bagi penelitian selanjutnya dalam bidang prediksi tingkat stres mahasiswa. Untuk pengembangan ke depan, disarankan agar penelitian selanjutnya mempertimbangkan penambahan variabel yang lebih beragam serta melakukan perbandingan dengan algoritma klasifikasi lainnya, sehingga diperoleh hasil analisis yang lebih mendalam dan akurat.

DAFTAR PUSTAKA

- AFISTIAN, R., BETIE, F., & WAHYU, E. S. (2025). HUBUNGAN ADAPTASI DIRI DENGAN TINGKAT STRES MAHASISWA BARU FAKULTAS ILMU KEPERAWATAN UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG. CORONA: JURNAL ILMU KESEHATAN UMUM, PSIKOLOG, KEPERAWATAN DAN KEBIDANAN Учредители: Asosiasi Riset Ilmu Manajemen dan Bisnis Indonesia, 3(1), 243-252.
- Wistarini, N. N. I. P., & Marheni, A. (2019). Peran dukungan sosial keluarga dan efikasi diri terhadap stres akademik mahasiswa baru Fakultas Kedokteran Universitas Udayana angkatan 2018. Jurnal Psikologi Udayana Edisi Khusus Psikologi Pendidikan, 4(3), 164-173.
- Parraga-alava, J. (2024). Stress Detection among Higher Education Students: A Comprehensive Systematic

- Review of Machine Learning Approaches. 2024 Tenth International Conference on EDemocracy & EGovernment (ICEDEG), 1–8. <https://doi.org/10.1109/ICEDEG61611.2024.10702055>.
- Dhinora, M. Y., & Mailoa, E. (2025). Analisa Tweet Mahasiswa untuk Deteksi Gejala Depresi dengan Penerapan Natural Language Processing. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 6(2), 1193-1211.
- Luci, M., Hernandez, Z., Daza, A., Dario, R., Rivera, M., Gustavo, E., Alamo, S., & Apaza-perez, G. (2025). *Comparative Analysis of Machine Learning Algorithms for Academic Performance Prediction Using Oversampling Techniques*. 12(3), 227–245. <https://doi.org/10.33168/JLISS.2025>.
- Tan, J., Mao, Y., & Li, Y. (2024). The Application of Big Data Technology in Teaching College Students' Mental Health Education in Colleges and Universities. *Applied Mathematics and Nonlinear Sciences*, 9(1), 1–17.
- Sasane, S., & Mulla, Z. A. S. (2024). Predictive Modelling Of Stress Levels: A Comparative Analysis Of Machine Learning Algorithms. *Journal of Advanced Zoology*, 45(S-4), 153–158.
- Adie Wahyudi Oktavia Gama, Agustini Degni Melsy Grren, I Gusti Ngurah Darma Paramartha, G., & Humaswara Prathama, Ni Made Widnyani, M. W. P. D. (2025). PENERAPAN ALGORITMA NAÏVE BAYES UNTUK PREDIKSI PENYAKIT DEPRESI PADA MAHASISWA. *Journal of Language and Health*, 6(2), 199–216.
- Chowdhury, S., Chowdhury, A. F., Islam, A., & Bary, A. N. (2024). *Machine Learning and Deep Learning Models for Predicting Mental Health Disorders and Performance Analysis through Chatbot Interactions*. 12(9), 38–60.
- Amin, M., Efendi, S., & Elveny, M. (2024, October). Analysis of Machine Learning Methods for Solving Cumulative Vehicle Routing Problem. In 2024 Ninth International Conference on Informatics and Computing (ICIC) (pp. 1-4). IEEE.
- Novelan, M. S., & Iqbal, M. (2025, November). Comparative Analysis of the Performance of K-Nearest Neighbor (K-NN) and Naive Bayes Algorithms on User Satisfaction Levels of the Tokopedia Application. In Proceedings of The International Conference on Computer Science, Engineering, Social Science, and Multi-Disciplinary Studies (Vol. 1, pp. 218-224).
- Fatah, Z., & Hasanah, U. (2025). *Prediksi Tingkat Stress dan Kesehatan Mental Mahasiswa Menggunakan Algoritma SVM*. 4, 200–207.
- Sharma, N. (2023). *Analysis of Student's Academic Performance based on their Time Spent on Extra-Curricular Activities using Machine Learning Techniques*. 15(1), 46–57. <https://doi.org/10.5815/ijmecs.2023.01.04>.
- Thanh Noi, P., & Kappas, M. (2018). Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. *Sensors*, 18(1), 18. <https://doi.org/10.3390/s18010018>.
- Kartianom, K., Arpandi, A., K. Kassymova, G., & Ndayizeye, O. (2022). Prediction model of teacher candidate student graduation status: Decision Tree C4.5, Naive Bayes, and k-NN. *Ekspose: Jurnal Penelitian Hukum Dan Pendidikan*, 21(2), 1419–1427. <https://doi.org/10.30863/ekspose.v21i2.3407>