

Komparasi Algoritma KNN Dan SVM Untuk Klasifikasi Kesehatan Mental Pada Usia Remaja

*A Comparison of KNN and SVM Algorithms for Mental Health Classification in
Adolescents*

Rizky Hidayat Hasibuan^{*1}, Mulkan Azhari²

^{1,2}Program studi Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

E-mail: ¹2109020130@umsu.ac.id, ²mulkanazhari@umsu.ac.id

Abstrak

Kesehatan mental pada usia remaja merupakan isu krusial yang terus mengalami peningkatan dan sering kali tidak terdeteksi secara dini akibat keterbatasan metode penilaian yang masih bersifat subjektif. Penelitian ini bertujuan untuk mengklasifikasikan tingkat kesehatan mental remaja berdasarkan tingkat stres menggunakan pendekatan machine learning serta membandingkan kinerja algoritma KNN dan SVM. Dataset yang digunakan terdiri dari 1.100 data remaja yang diperoleh dari GitHub dengan 11 atribut prediktor dan satu atribut target, yaitu *stress_level* yang diklasifikasikan ke dalam tiga kelas: rendah, sedang, dan tinggi. Tahapan penelitian meliputi preprocessing data, EDA, feature selection, penanganan ketidakseimbangan data menggunakan SMOTE, pemodelan, dan evaluasi. Pengujian dilakukan menggunakan beberapa rasio pembagian data latih dan data uji. Evaluasi kinerja model menggunakan confusion matrix. Hasil penelitian menunjukkan bahwa algoritma SVM memberikan performa terbaik dengan akurasi sebesar 89,55% dan F1-Score sebesar 89,58% pada rasio data 80:20 sebelum penerapan SMOTE. Secara keseluruhan, SVM terbukti lebih stabil dan akurat dibandingkan KNN dalam mengklasifikasikan tingkat kesehatan mental remaja, sehingga berpotensi digunakan sebagai dasar sistem deteksi dini kesehatan mental berbasis data.

Kata kunci: Kesehatan Mental Remaja, Klasifikasi Tingkat Stres, K-Nearest Neighbor, *Support Vector Machine*

Abstract

Mental health among adolescents is a critical issue that continues to increase and is often not detected early due to the limitations of assessment methods that remain largely subjective. This study aims to classify adolescent mental health levels based on stress levels using a machine learning approach and to compare the performance of the KNN and SVM algorithms. The dataset used consists of 1,100 adolescent records obtained from GitHub, comprising 11 predictor attributes and one target attribute, namely stress_level, which is classified into three categories: low, moderate, and high. The research stages include data preprocessing, EDA, feature selection, handling class imbalance using the SMOTE, modeling, and evaluation. Model testing was conducted using several training-testing split ratios. Model performance was evaluated using confusion matrix. The results indicate that the SVM algorithm achieved the best performance with an accuracy of 89.55% and an F1-Score of 89.58% using an 80:20 data split prior to the application of SMOTE. Overall, SVM demonstrated higher stability and accuracy compared to KNN in classifying adolescent mental health levels, indicating its strong potential as a data-driven early detection tool for adolescent mental health issues.



Keywords: *Adolescent Mental Health, Stress Level Classification, K-Nearest Neighbor, Support Vector Machine*

1. PENDAHULUAN

Kesehatan mental remaja merupakan salah satu permasalahan global yang terus mengalami peningkatan dan memberikan dampak signifikan terhadap perkembangan psikologis, akademik, serta sosial individu. Masa remaja merupakan fase transisi yang rentan terhadap tekanan psikologis akibat tuntutan akademik, lingkungan sosial, serta perubahan emosional yang kompleks. Salah satu indikator utama gangguan kesehatan mental pada remaja adalah tingkat stres yang tidak terkelola dengan baik, yang apabila tidak terdeteksi sejak dini dapat berkembang menjadi gangguan mental yang lebih serius [1].

Selama ini, penilaian kesehatan mental remaja umumnya dilakukan melalui kuesioner atau wawancara yang bersifat subjektif dan sangat bergantung pada persepsi individu. Metode tersebut memiliki keterbatasan dalam hal konsistensi, objektivitas, serta efisiensi, terutama ketika diterapkan pada data dalam jumlah besar. Oleh karena itu, diperlukan pendekatan alternatif yang lebih objektif dan berbasis data untuk membantu proses klasifikasi tingkat kesehatan mental secara akurat [2].

Perkembangan teknologi *machine learning* membuka peluang besar dalam analisis dan klasifikasi data kesehatan mental. Algoritma klasifikasi seperti K-Nearest Neighbor (KNN) dan *Support Vector Machine* (SVM) telah banyak digunakan dalam berbagai studi karena kemampuannya dalam menangani data multiklas dan menghasilkan tingkat akurasi yang baik. Namun, perbedaan karakteristik kedua algoritma tersebut menimbulkan variasi performa yang perlu dikaji lebih lanjut, khususnya pada data kesehatan mental remaja yang cenderung kompleks dan tidak seimbang [3].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengklasifikasikan tingkat kesehatan mental remaja berdasarkan tingkat stres menggunakan algoritma KNN dan SVM serta membandingkan kinerja kedua metode tersebut. Hasil penelitian diharapkan dapat memberikan kontribusi dalam pengembangan sistem deteksi dini kesehatan mental remaja yang lebih objektif, akurat, dan berbasis data.

2. METODE PENELITIAN

Terdapat beberapa metode yang dijadikan sebagai sumber tinjauan untuk mengkaji klasifikasi kesehatan mental pada usia remaja dengan mengkomparasi algoritma KNN dan SVM.

2.1 Klasifikasi

Klasifikasi adalah penggalan data untuk menemukan pola dan hubungan dalam dataset, dengan tujuan untuk mengelompokkan dan mengklasifikasi data ke dalam kategori yang telah ditentukan sebelumnya. Pada tahap awal, klasifikasi melibatkan pengumpulan data yang relevan untuk proses klasifikasi. Data ini dapat berasal dari berbagai sumber seperti basis data perusahaan, aplikasi perangkat lunak, atau sumber data eksternal lainnya. Penting untuk memastikan kualitas data yang baik dan relevan dalam proses ini untuk mendapatkan hasil yang akurat dan bermakna [4].

2.2 Kesehatan Mental

Kesehatan mental adalah bagian mendasar dari definisi kesehatan. Kesehatan mental yang baik memungkinkan orang untuk mencapai potensi mereka, mengatasi tekanan kehidupan sehari-hari, bekerja secara produktif, dan berkontribusi pada komunitas mereka. Oleh karena itu, keberadaan gangguan jiwa tidak boleh dianggap remeh. Sebab, jumlah kasusnya masih sangat



memprihatinkan. Sekitar 450 juta orang di seluruh dunia menderita gangguan mental dan perilaku. Diperkirakan satu dari empat orang akan menderita gangguan jiwa semasa hidupnya. Menurut WHO SEARO, India memiliki jumlah kasus gangguan depresi tertinggi (56.675.969 kasus atau 4,5% dari populasi) dan Maladewa memiliki jumlah kasus terendah (12.739 kasus atau 4,5% dari populasi). Sedangkan jumlah orang yang terinfeksi di Indonesia sebanyak 9.162.886 orang atau setara dengan 3,7% dari jumlah penduduk [5].

2.3 *K-Nearest Neighbor* (KNN)

K-Nearest Neighbors (KNN) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Algoritma ini bekerja dengan mencari sejumlah k tetangga terdekat dari data yang ingin diprediksi, berdasarkan metrik jarak tertentu seperti jarak *Euclidean*. Prediksi dilakukan dengan menganalisis label atau nilai dari tetangga-tetangga terdekat tersebut. *K-Nearest Neighbors* (KNN) dikenal karena kesederhanaannya dan efektivitasnya dalam berbagai aplikasi, seperti pengenalan pola dan pengambilan keputusan. *K-Nearest Neighbors* (KNN) merupakan salah satu algoritma pembelajaran mesin yang bersifat sederhana namun efektif dalam tugas klasifikasi dan regresi. Prinsip dasar dari algoritma ini adalah asumsi bahwa objek-objek yang berada dalam jarak yang dekat satu sama lain cenderung memiliki karakteristik yang serupa [6].

2.4 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah metode pembelajaran *supervised* yang menganalisis data dan mengenali pola untuk klasifikasi dan regresi, SVM bekerja menggunakan cara mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas, *hyperplane* merupakan sebuah fungsi yang digunakan untuk pemisah antar kelas. SVM merupakan sebuah pengklasifikasi maka diberi suatu himpunan pelatihan yang ditandai sebagai milik salah satu dari kelas kategori, algoritma SVM membangun sebuah model yang memprediksi apakah data yang baru diproses merupakan salah satu dalam kategori yang lain [7].

2.5 *Python*

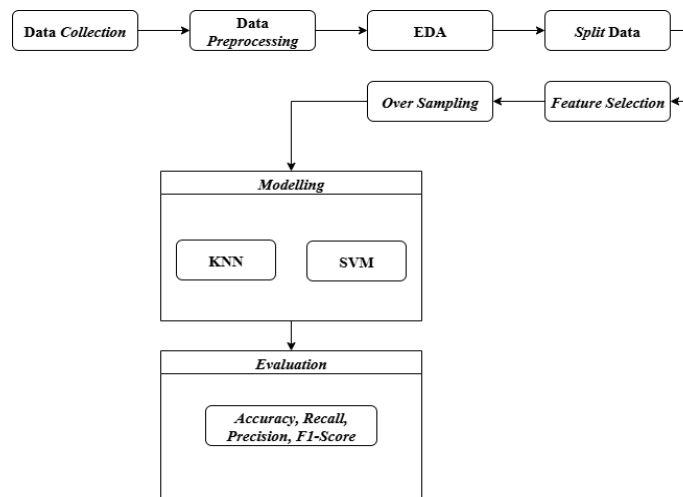
Python adalah bahasa scripting tingkat tinggi yang kuat, interaktif, dan berorientasi objek. *Python* merupakan bahasa pemrograman yang *freeware*, tidak ada batasan dalam penyalinannya atau mendistribusikannya lengkap dengan *source code* nya, *debugger* dan *profiler*, fungsi sistem, GUI (antar muka pengguna grafis) dan basis datanya [8].

2.6 *Google Collaboratory*

Google Collaboratory atau *Colab* adalah proyek yang memiliki tujuan untuk menyebarluaskan pendidikan dan penelitian pembelajaran mesin. *Collaboratory* menyediakan *runtime Python* 2 dan 3 yang telah dikonfigurasi sebelumnya dengan pembelajaran mesin yang penting dan perpustakaan kecerdasan buatan, seperti *Tensorflow*, *Matplotlib*, dan *Keras*. Layanan ini menyediakan *runtime* yang dipercepat GPU, juga sepenuhnya dikonfigurasi dengan perangkat



lunak yang sebelumnya diuraikan. Infrastruktur Kolaborasi *Google* di-host di *platform Google Cloud* [9].



Gambar 1. Metodologi Penelitian

2.7 Data Collection

Tahap awal dalam penelitian ini dimulai dengan proses data *Collection* atau pengumpulan data, di mana langkah ini dilakukan dengan mengumpulkan informasi yang berkaitan dengan kesehatan mental. Dalam penelitian ini, dataset yang digunakan berasal dari *platform* GitHub dengan total 1.100 baris data dan 12 kolom.

Tabel 1. Dataset Kesehatan Mental

anxiety_level	mental_health_history	depression	headache	sleep_quality	breathing_problem	living_conditions	academic_performance	study_load	future_career_concerns	extracurricular_activities	stress_level
14	0	11	2	2	4	3	3	2	3	3	1
15	1	15	5	1	4	1	1	4	5	5	2
12	1	14	2	2	2	2	2	3	2	2	1
16	1	15	4	4	3	2	2	4	4	4	2
16	0	7	2	5	1	2	4	3	2	0	1
20	1	21	3	1	4	2	2	5	5	4	2
4	0	6	1	4	1	4	5	1	1	2	0
17	1	22	4	1	5	1	1	3	4	4	2
13	1	12	3	2	4	3	3	3	3	2	1
6	0	27	4	6	2	5	2	2	5	3	1



Setelah *dataset* didapatkan, kolom target yang digunakan pada dataset tersebut yaitu kolom 'stress_level' yang berisi angka 0 untuk tingkat stres Rendah, 1 untuk tingkat stres Sedang, dan 2 untuk tingkat stres Tinggi yang selanjutnya akan dilakukan proses pembersihan data atau *Preprocessing data*.

2.8 *Proprocessing Data*

Preprocessing data merupakan tahap penting dalam *pipeline data science* yang memiliki dampak besar terhadap hasil akhir analisis. Langkah ini bertujuan untuk memastikan bahwa data yang digunakan bersih, konsisten, serta siap dianalisis atau dimodelkan. Proses *Preprocessing data* mencakup berbagai tahapan yang dirancang untuk menangani permasalahan seperti data yang hilang, *noise*, inkonsistensi, atau format yang tidak sesuai. Adapun tahapan dalam *Preprocessing data* pada penelitian ini sebagai berikut:

a. *Cleaning Data*

Cleaning data merupakan tahap krusial yang harus diperhatikan dalam proses analisis data serta pemodelan *machine learning*. Dengan menghilangkan nilai yang hilang, duplikasi, dan pencilan, dataset menjadi lebih rapi, konsisten, serta siap untuk dianalisis lebih lanjut.



Akibatnya, analisis statistik menjadi lebih akurat, dan model *machine learning* dapat bekerja dengan lebih andal.

Attribute	Count
anxiety_level	0
mental_health_history	0
depression	0
headache	0
sleep_quality	0
breathing_problem	0
living_conditions	0
academic_performance	0
study_load	0
future_career_concerns	0
extracurricular_activities	0
stress_level	0

Gambar 2. Atribut *missing values*

b. Menghapus *Outlier*

Tahap berikutnya adalah pemeriksaan *outlier* dalam dataset. Fungsi dari tahap ini adalah untuk meminimalkan pengaruh *outlier* ekstrem yang berpotensi menyebabkan bias pada model, menjaga keseimbangan distribusi data, serta meningkatkan kinerja model prediktif.

```
# 2. Preprocessing Data
## Cleaning Data: Drop Missing Values & Duplicates
df = df.dropna()
df = df.drop_duplicates()

## Handling Outliers using IQR
for col in df.select_dtypes(include=[np.float64, np.int64]).columns:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df[col] = np.where(df[col] < lower_bound, lower_bound, df[col])
    df[col] = np.where(df[col] > upper_bound, upper_bound, df[col])

# Visualisasi distribusi data sebelum dan sesudah mengatasi outlier
for col in df.select_dtypes(include=[np.float64, np.int64]).columns:
    plt.figure(figsize=(12,5))
    plt.subplot(1,2,1)
    sns.boxplot(x=df[col])
    plt.title(f'Before Outlier Handling: {col}')

    plt.subplot(1,2,2)
    sns.boxplot(x=df[col])
    plt.title(f'After Outlier Handling: {col}')
    plt.show()
```

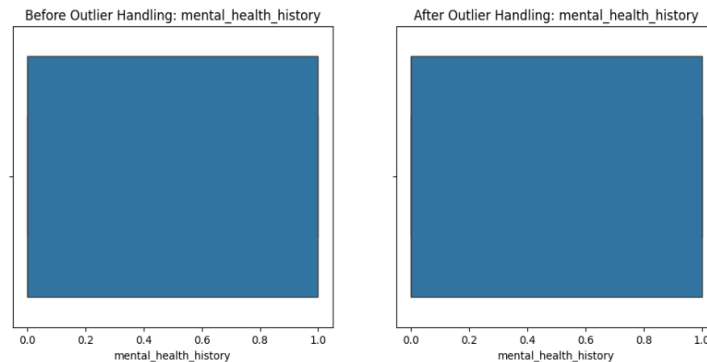
Gambar 3. Mengatasi *outlier*

2.9 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) merupakan proses mengeksplorasi dan menganalisis dataset bertujuan untuk memahami karakteristik, pola, serta hubungan antar variabel baik secara visual maupun statistik. EDA memiliki tujuan utama untuk memperoleh wawasan (*insight*) dari data, mengidentifikasi permasalahan seperti *missing values* atau *outlier*, serta mempersiapkan data untuk analisis atau pemodelan lebih lanjut [10]. Penanganan *outlier* merupakan tahap krusial dalam *Preprocessing data* guna memastikan bahwa analisis statistik dan model *machine learning* didasarkan pada data yang bersih serta representatif. Dengan menghapus *outlier*, distribusi data



menjadi lebih stabil, sehingga meningkatkan performa model serta akurasi dalam analisis statistik.



Gambar 4. Proses setelah menghapus outlier

2.10 Splitting Data

Pemisahan data atau *data Splitting* dilakukan dengan pembagian dataset menjadi dua bagian, yakni data latih dan data uji. Data latih berperan dalam proses pelatihan model, sedangkan data uji digunakan untuk mengevaluasi performa model yang telah dilatih. Proses pemisahan data ini dilakukan dengan berbagai proporsi, seperti 90%:10%, 80%:20%, 70%:30%, dan 60%:40%, yang bertujuan untuk membandingkan hasil akurasi terbaik.

2.11 Feature Selection

Setelah tahap *Preprocessing data* selesai, proses selanjutnya adalah melakukan *label encoding* atau mengubah data kategori menjadi format numerik. *Label encoding* ialah metode dalam pemrosesan data yang berfungsi untuk mengonversi nilai-nilai dalam satu kolom kategori ke dalam bentuk numerik atau label. Semua teks yang berbentuk label dikonversi menjadi nilai numerik [11].

Tabel 2. Atribut yang digunakan

Variabel X	Variabel Y
'anxiety_level',	
'mental_health_history',	
'depression',	
'headache',	
'future_career_concerns',	
'extracurricular_activities',	'stress_level'
'study_load',	
'breathing_problem',	
'living_conditions',	
'sleep_quality',	
'academic_performance'	



Pada atribut “stress_level” terdapat 3 label yaitu 0, 1 dan 2 yang artinya label 0 yaitu “rendah”, label 1 yaitu “sedang” dan 2 yaitu “tinggil”.

2.12 Oversampling

Oversampling dengan SMOTE (*Synthetic Minority Over-sampling Technique*) ialah metode yang diterapkan untuk mengatasi ketidak seimbangan kelas (*imbalanced dataset*) dalam *machine learning*. Metode ini berfungsi menambahkan data sintetis pada kelas minoritas guna menyeimbangkan distribusi kelas, sehingga model *machine learning* dapat belajar lebih optimal tanpa cenderung bias terhadap kelas mayoritas [12].

```
[10] # 7. Over Sampling menggunakan SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

#Sebelum menggunakan SMOTE
print(y_train.value_counts(), "\n")

#Setelah menggunakan SMOTE
print(pd.Series(y_train_resampled).value_counts())
```

```
stress_level
1    261
0    255
2    199
Name: count, dtype: int64

stress_level
1    261
0    261
2    261
Name: count, dtype: int64
```

Gambar 5. *Oversampling* menggunakan SMOTE

Kelas 0 (rendah) terdiri dari 255 sampel, Kelas 1 (sedang) terdiri dari 261, sedangkan Kelas 2 (tinggi) memiliki 199 sampel. Hal ini menunjukkan bahwa dataset tidak seimbang karena Kelas 1 lebih dominan dibandingkan Kelas 0 dan 2. Setelah menerapkan SMOTE, Kelas 1 dan 2, yang sebelumnya merupakan kelas minoritas, telah di *over-sampling* hingga mencapai 261 sampel, sementara Kelas 1 tetap memiliki 261 sampel. Dengan demikian, dataset kini menjadi seimbang, karena kedua kelas memiliki jumlah sampel yang sama.

3. HASIL DAN PEMBAHASAN

3.1 Modelling

Tahap pemodelan bertujuan untuk membangun model klasifikasi kesehatan mental remaja menggunakan dua algoritma pembelajaran mesin, yaitu *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM). Proses dilakukan melalui beberapa tahap, mulai dari pembagian data (*Splitting*), pelatihan (*training*), pengujian (*testing*), hingga penerapan metode *Oversampling*



menggunakan SMOTE (*Synthetic Minority Oversampling Technique*) untuk mengatasi ketidakseimbangan data antar kelas.

a. Hasil Implementasi pada Berbagai Rasio Split

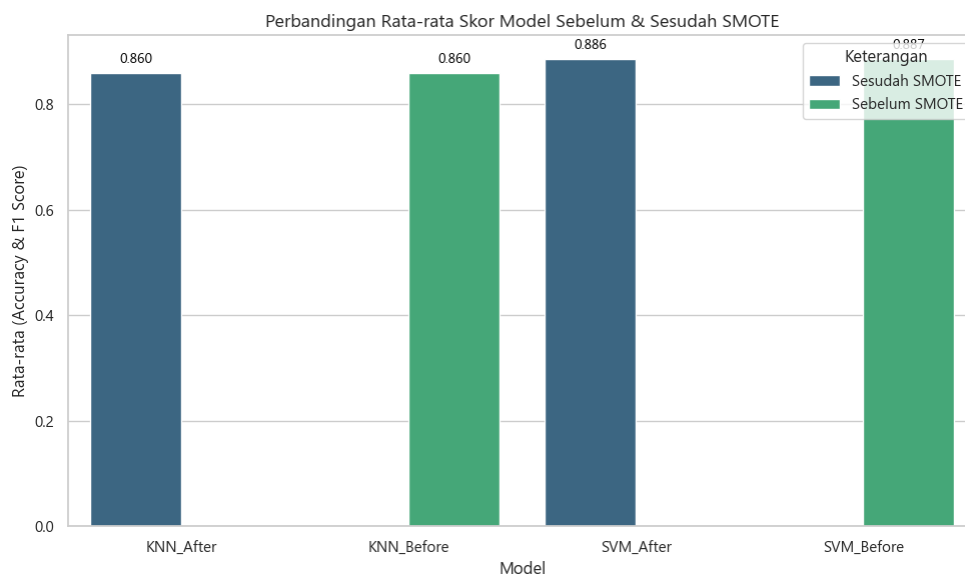
Tabel berikut menampilkan hasil pengujian untuk masing-masing algoritma dengan empat variasi rasio pembagian data. Pengujian menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk menilai performa model secara komprehensif.

Tabel 3. Hasil Komparasi Model KNN dan SVM pada Berbagai Split Data

Rasio (Train : Test)	Model	Kondisi Data	Accuracy	Precision	Recall	F1-Score
90 : 10	KNN	Sebelum SMOTE	0.8711	0.8703	0.8699	0.8701
	KNN	Sesudah SMOTE	0.8785	0.8783	0.8784	0.8781
	SVM	Sebelum SMOTE	0.8882	0.8877	0.8880	0.8879
	SVM	Sesudah SMOTE	0.8847	0.8841	0.8843	0.8845
80 : 20	KNN	Sebelum SMOTE	0.8753	0.8742	0.8738	0.8737
	KNN	Sesudah SMOTE	0.8835	0.8836	0.8831	0.8834
	SVM	Sebelum SMOTE	0.8955	0.8957	0.8957	0.8958
	SVM	Sesudah SMOTE	0.8912	0.8910	0.8908	0.8909
70 : 30	KNN	Sebelum SMOTE	0.8698	0.8689	0.8687	0.8686
	KNN	Sesudah SMOTE	0.8769	0.8765	0.8767	0.8766
	SVM	Sebelum SMOTE	0.8822	0.8818	0.8820	0.8819
	SVM	Sesudah SMOTE	0.8803	0.8800	0.8798	0.8801
60 : 40	KNN	Sebelum SMOTE	0.8621	0.8614	0.8612	0.8613
	KNN	Sesudah SMOTE	0.8709	0.8706	0.8704	0.8705
	SVM	Sebelum SMOTE	0.8768	0.8765	0.8761	0.8763
	SVM	Sesudah SMOTE	0.8745	0.8740	0.8743	0.8742

Berdasarkan tabel di atas, terlihat bahwa baik KNN maupun SVM menunjukkan peningkatan performa setelah dilakukan *balancing* data menggunakan SMOTE, terutama pada rasio data yang lebih kecil (misalnya 70:30 dan 60:40). Namun, performa terbaik secara keseluruhan diperoleh pada rasio 80:20 dengan algoritma SVM sebelum SMOTE, yang menghasilkan nilai akurasi sebesar 0.8955 dan *F1-score* sebesar 0.8958. Hal ini menunjukkan bahwa SVM mampu membangun *hyperplane* pemisah yang optimal bahkan tanpa peningkatan distribusi data, sehingga tidak terlalu terpengaruh oleh ketidakseimbangan kelas.





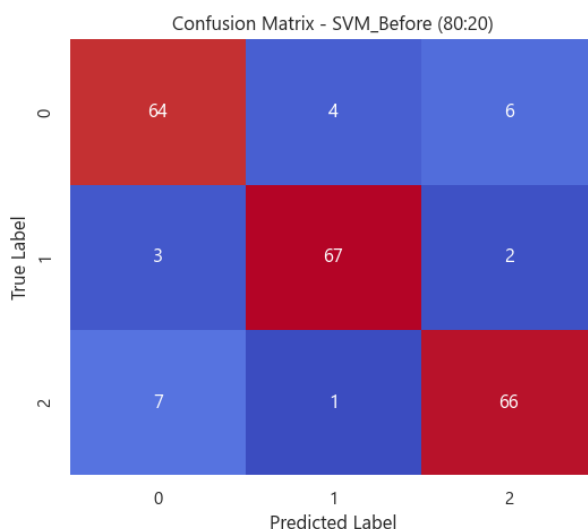
Gambar 6. Perbandingan rata-rata skor model sebelum dan sesudah SMOTE

Berdasarkan Gambar 4.1 yang menampilkan perbandingan rata-rata skor model sebelum dan sesudah penerapan SMOTE, terlihat bahwa algoritma *Support Vector Machine* (SVM) memperoleh performa terbaik dengan nilai rata-rata sebesar 0.897 pada kondisi sebelum SMOTE, sedangkan algoritma K-Nearest Neighbor (KNN) menunjukkan hasil yang relatif sama baik sebelum maupun sesudah SMOTE dengan nilai rata-rata 0.860. Hasil ini menunjukkan bahwa SVM memiliki kemampuan klasifikasi yang lebih stabil dan akurat tanpa perlu dilakukan penyeimbangan data, sementara penerapan SMOTE hanya memberikan pengaruh kecil terhadap peningkatan performa KNN. Secara keseluruhan, model SVM terbukti lebih unggul dalam mengklasifikasikan tingkat kesehatan mental pada remaja karena mampu mempertahankan keseimbangan antara precision dan recall dengan nilai F1-score yang tinggi, sedangkan KNN cenderung sensitif terhadap distribusi dan jarak antar data sehingga hasil klasifikasinya tidak mengalami peningkatan signifikan setelah dilakukan proses *Oversampling*.

3.2 Evaluation

Evaluasi dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, serta dianalisis lebih lanjut melalui *Confusion Matrix*. Evaluasi bertujuan untuk memastikan seberapa baik model mengenali setiap kelas tingkat stres (Rendah, Sedang, Tinggi).

- a. Evaluasi Model Terbaik (SVM, Split 80:20, Sebelum SMOTE)



Gambar 7. Hasil Evaluasi Detail Model Terbaik

Dari hasil *Confusion Matrix*, terlihat bahwa sebagian besar prediksi sesuai dengan kelas aktual. Kesalahan klasifikasi yang dominan terjadi antara kelas Rendah dan Tinggi, menunjukkan bahwa model kadang kesulitan membedakan kondisi yang ekstrem karena adanya kemiripan fitur perilaku antara kedua kelompok tersebut.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pengujian yang dilakukan, dapat disimpulkan bahwa tahapan preprocessing data, EDA, seleksi fitur, dan *oversampling* berhasil menghasilkan dataset yang layak untuk klasifikasi tingkat stres remaja, dengan algoritma *Support Vector Machine* (SVM) menunjukkan performa terbaik dibandingkan K-Nearest Neighbor (KNN) melalui capaian akurasi 0,8955 dan F1-Score 0,8958, sementara penerapan SMOTE tidak memberikan peningkatan signifikan karena distribusi data yang relatif seimbang. Hasil ini menunjukkan bahwa tingkat stres remaja dipengaruhi secara signifikan oleh kombinasi faktor internal seperti emosional dan harga diri serta faktor eksternal seperti tekanan akademik, kualitas tidur, dan dukungan sosial, sehingga memperkuat teori mengenai kompleksitas determinan kesehatan mental remaja.

Penelitian selanjutnya disarankan untuk mengeksplorasi algoritma lain yang lebih kompleks seperti *Random Forest*, *Gradient Boosting*, atau *Deep Neural Network* (DNN), memperluas dataset dengan atribut psikologis dan perilaku tambahan, serta menambahkan fitur seperti mekanisme *coping* dan penggunaan media sosial guna meningkatkan kemampuan generalisasi model. Selain itu, pengembangan model ke dalam sistem deteksi dini berbasis web atau aplikasi serta kolaborasi dengan ahli psikologi sangat diperlukan agar hasil klasifikasi tidak hanya unggul secara statistik, tetapi juga relevan dan bermanfaat secara klinis.



DAFTAR PUSTAKA

- [1] Asril, K. Ningsih, R. Fitri, and Muhamadiyah, "Edukasi Kesehatan Mental pada Remaja," *Health Community Service*, vol. 1, no. 1, pp. 32–36, Dec. 2023, doi: 10.47709/HCS.V1I1.3238.
- [2] M. Idris, A. Wijaya, L. Septiani, T. A. Happy, and R. Graharti, "Pemanfaatan Kecerdasan Buatan sebagai Alat Bantu Diagnosis di Bidang Kesehatan : Literatur Review," *Jurnal Kedokteran Universitas Lampung*, vol. 9, no. 1, pp. 117–121, Jun. 2025, doi: 10.23960/JKUNILA.V9I1.PP117-121.
- [3] V. Oktaviani, N. Rosmawarni, and M. P. Muslim, "Perbandingan Kinerja Random Forest Dan Smote Random Forest Dalam Mendeteksi Dan Mengukur Tingkat Stres Pada Mahasiswa Tingkat Akhir," *Informatik : Jurnal Ilmu Komputer*, vol. 20, no. 1, pp. 43–49, Apr. 2024, doi: 10.52958/IFTK.V20I1.9158.
- [4] A. R. Aisy, Rahmadden, A. H. Siregar, S. Daulay, and N. Ananta, "Implementasi Algoritma *Support Vector Machine* Dalam Mengklasifikasikan Berita Hoax Di Indonesia Pada Media Sosial X," *Jurnal RESTIKOM : Riset Teknik Informatika dan Komputer*, vol. 7, no. 2, pp. 206–216, Aug. 2025, doi: 10.52005/RESTIKOM.V7I2.457.
- [5] A. N. Haryanti, M. B. S. Putra, N. Larasati, V. N. Khairunnisa, and L. D. Dewi, "Analisis Kondisi Kesehatan Mental di Indonesia Dan Strategi Penanganannya," *Student Research Journal*, vol. 2, no. 3, pp. 28–40, May 2024, doi: 10.55606/SRJYAPPI.V2I3.1221.
- [6] A. Yogiarto, A. Homaidi, and Z. Fatah, "A Implementation of the K-Nearest Neighbors (KNN) Method for Classification of Heart Disease," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 3, pp. 1720–1728, Jul. 2024, doi: 10.33379/GTECH.V8I3.4495.
- [7] M. P. Utami, F. Suroso, F. Lailasari, F. P. J. Sibuea, and K. Chandra, "Integrasi Algoritma *Support Vector Machine* dengan Java untuk Memprediksi Kualitas Komponen Otomotif dalam Industri 4.0," *Techno.Com*, vol. 24, no. 3, pp. 790–797, Aug. 2025, doi: 10.62411/TC.V24I3.12719.
- [8] S. Junaidi, M. Devegi, and H. Kurniawan, "Pelatihan Pengolahan dan Visualisasi Data Penduduk menggunakan Python," *ADMA: Jurnal Pengabdian dan Pemberdayaan Masyarakat*, vol. 4, no. 1, pp. 151–162, Jul. 2023, doi: 10.30812/ADMA.V4I1.2963.
- [9] R. G. Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/JTEKSIS.V5I1.750.
- [10] S. Larasati, E. Dewi, B. H. Farhansyah, F. Abdurrachman Bachtiar, and F. Pradana, "Penerapan Decision Tree dan Random Forest dalam Deteksi Tingkat Stres Manusia Berdasarkan Kondisi Tidur," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 5, pp. 1043–1050, Oct. 2024, doi: 10.25126/JTIK.2024117993.
- [11] C. Herdian, A. Kamila, and I. G. A. M. Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode



- Linear Regresi,” *Technologia : Jurnal Ilmiah*, vol. 15, no. 1, pp. 93–108, Jan. 2024, doi: 10.31602/TJI.V15I1.13457.
- [12] A. Nugroho and E. Rilvani, “Penerapan Metode Oversampling SMOTE Pada Algoritma Random Forest Untuk Prediksi Kebangkrutan Perusahaan,” *Techno.Com*, vol. 22, no. 1, pp. 207–214, Feb. 2023, doi: 10.33633/TC.V22I1.7527.

