

Optimasi Model *Machine Learning* dan Analisis Interpretabilitas untuk Klasifikasi Tingkat Stres Mahasiswa

Machine Learning Model Optimization and Interpretability Analysis for Classifying Student Stress Levels

¹Samuel Wijayadi Sugiharto*, ²Yessica Nataliani

^{1,2}Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana

^{1,2}Jl. O. Notohamidjojo 1-10, Salatiga, Indonesia

*e-mail: yessica.nataliani@uksw.edu

(received: 10 March 2026, revised: 28 April 2026, accepted: 30 April 2026)

Abstrak

Penelitian ini bertujuan untuk membandingkan dan menganalisis beberapa kinerja algoritma dalam melakukan klasifikasi tingkat stres mahasiswa. Dataset yang digunakan adalah *Student Lifestyle Dataset* yang diperoleh dari repositori *Kaggle*, terdiri dari 2.000 data dengan delapan fitur gaya hidup mahasiswa. Metode yang digunakan meliputi penerapan tiga algoritma klasifikasi, yaitu *Logistic Regression*, *K-Nearest Neighbors*, dan *Support Vector Machine* pada empat skenario eksperimen yang mencakup model *baseline*, penanganan ketidakseimbangan data menggunakan *Synthetic Minority Oversampling Technique*, seleksi fitur dengan *Recursive Feature Elimination*, serta *hyperparameter tuning*. Hasil akan dievaluasi menggunakan metrik akurasi, *precision*, *recall*, *F1-Score*, kemudian model terbaik akan dilakukan analisis interpretabilitas menggunakan *SHAP*. Hasil penelitian menunjukkan bahwa integrasi teknik penyeimbangan data, seleksi fitur, dan optimasi parameter dengan algoritma *SVM* membuat performa yang signifikan dan optimal dengan akurasi sebesar 0,98, *precision* 0,96, *recall* 0,98, *F1-Score* 0,97, dan waktu komputasi sebesar 0,011 detik. Analisis interpretabilitas menunjukkan bahwa faktor gaya hidup seperti durasi belajar dan durasi tidur berpengaruh secara dominan terhadap tingkat stres. Hasil ini menunjukkan bahwa dengan strategi optimasi terintegrasi tersebut berhasil mendukung deteksi tingkat stres mahasiswa secara cepat dan akurat.

Kata kunci: klasifikasi, *machine learning*, optimasi model, stres mahasiswa, *SHAP*

Abstract

This study aims to compare and analyze the performance of several algorithms in classifying student stress levels. The dataset used in this research is the Student Lifestyle Dataset obtained from the Kaggle repository, consisting of 2,000 records with eight student lifestyle features. The methods employed include the implementation of three classification algorithms: Logistic Regression, K-Nearest Neighbors, and Support Vector Machine (SVM), across four experimental scenarios. These scenarios include a baseline model, handling imbalanced data using the Synthetic Minority Oversampling Technique (SMOTE), feature selection using Recursive Feature Elimination (RFE), and hyperparameter tuning. The results were evaluated using accuracy, precision, recall, and F1-score metrics. Furthermore, interpretability analysis of the best-performing model was conducted using SHAP. The findings indicate that the integration of data balancing techniques, feature selection, and parameter optimization with the SVM algorithm significantly improved performance, achieving an accuracy of 0.98, precision of 0.96, recall of 0.98, F1-score of 0.97, and a computation time of 0.011 seconds. The interpretability analysis revealed that lifestyle factors such as study duration and sleep duration had the most dominant influence on stress levels. These results demonstrate that the integrated optimization strategy successfully supports fast and accurate detection of student stress levels.

Keywords: classification, machine learning, model optimization, student stress, *SHAP*

1 Pendahuluan

Hampir seluruh mahasiswa di dunia saat ini mengalami stres yang merupakan bagian dari kesehatan mental [1]. Kondisi ini timbul akibat tubuh merespon berbagai tekanan, seperti meningkatnya tuntutan hidup di era modern, ekspektasi dari diri sendiri maupun lingkungan sosial, masalah keuangan, hingga upaya untuk menyeimbangkan kehidupan pribadi dan akademik [2][3]. Tekanan yang menimbulkan stres ini memberikan dampak secara langsung pada berbagai aspek di dalam kehidupan mahasiswa, seperti kinerja akademik, daya ingat, manajemen waktu, hingga kesejahteraan mental secara keseluruhan [4]. Tingkat stres tinggi menyebabkan efek domino yang menimbulkan dampak negatif pada gangguan tidur, tingkat kecemasan, hingga depresi. Kondisi ini akan menimbulkan siklus yang sulit untuk diakhiri, jika tidak terdeteksi dan ditangani sejak dini [5].

Di tengah tantangan tingkat stres yang terus meningkat, penggunaan teknologi menjadi bagian krusial dalam upaya mendeteksi dan penanganannya. Teknik klasifikasi dalam *machine learning* telah banyak digunakan dalam penelitian klasifikasi tingkat stres mahasiswa, dengan algoritma seperti *Logistic Regression*, *K-Nearest Neighbors* (K-NN), *Random Forest*, hingga *Decision Family*. Namun, penelitian-penelitian tersebut masih menghadapi sejumlah keterbatasan yang dapat mempengaruhi hasil klasifikasi. Pertama, permasalahan ketidakseimbangan data yang dapat menyebabkan model cenderung bias terhadap kelas mayoritas sehingga dapat menurunkan akurasi secara keseluruhan [6]. Penggunaan fitur yang berlebihan dapat menyebabkan model lebih rentan terhadap *noise* dan proses komputasi menjadi lebih lambat [7]. Selanjutnya, masih minimnya penggunaan optimasi *hyperparameter* yang dapat membuat model tidak mencapai performa terbaiknya [8].

Dalam klasifikasi penting untuk mendapatkan hasil yang berkualitas agar dapat membuat prediksi yang akurat. Untuk mendapatkan hasil yang maksimal, penelitian ini menggunakan beberapa variasi metode seperti *Synthetic Minority Oversampling Technique* (SMOTE), *feature selection*, dan *hyperparameter tuning* pada algoritma *machine learning*. SMOTE digunakan untuk mengatasi jumlah data yang tidak seimbang, dan mengurangi *overfitting*, sehingga data yang diolah dapat menghasilkan akurasi yang baik [9]. *Feature selection* merupakan metode untuk mengurangi fitur yang kurang relevan, sehingga waktu berjalannya proses klasifikasi dapat berlangsung dengan lebih cepat [10]. *Hyperparameter tuning* dilakukan untuk meningkatkan performa dari model klasifikasi dengan memilih nilai *hyperparameter* yang paling tepat [11].

Selain aspek performa, tantangan lain dalam penerapan *machine learning* adalah rendahnya interpretabilitas model. Model yang kompleks biasanya menghasilkan hasil klasifikasi tanpa memberikan informasi mengenai kontribusi masing-masing fitur, yang dikenal sebagai *black box* [12]. Kurangnya transparansi dapat menurunkan tingkat kepercayaan terhadap sistem, terlebih ketika hasil klasifikasi digunakan sebagai dasar perancangan intervensi atau kebijakan akademik. Untuk mengatasi permasalahan tersebut, diperlukan pendekatan yang mampu menjelaskan setiap fitur terhadap hasil klasifikasi. Salah satu metode interpretabilitas yang bisa digunakan untuk menjelaskan hal tersebut adalah *Shapley Additive Explanations* (SHAP). Melalui SHAP, model dapat diidentifikasi fitur mana yang paling berpengaruh dalam menentukan klasifikasi tingkat stres mahasiswa [13]. Pendekatan ini memungkinkan pengembangan model yang tidak hanya optimal dari sisi performa, tetapi juga transparan dan dapat dijelaskan.

Penelitian ini akan menggunakan model yang dibangun pada beberapa algoritma klasifikasi diantaranya, *Logistic Regression*, K-NN, dan *Support Vector Machine* (SVM). *Logistic Regression* dipilih karena mampu memodelkan hubungan antar fitur-fitur gaya hidup mahasiswa dengan tingkat stres secara probabilistik, serta menghasilkan output yang mudah diinterpretasikan sebagai hasil awal perbandingan. K-NN digunakan karena pendekatannya yang berbasis kemiripan pola antar data [14], memungkinkan klasifikasi tingkat stres berdasarkan karakteristik mahasiswa yang serupa. SVM dipilih karena kemampuannya untuk memisahkan kelas stres secara maksimal dengan menemukan *hyperplane* optimal, hal ini membuat model *robust* terhadap data dengan batas kelas yang tidak linear [15].

Penelitian ini bertujuan untuk melakukan analisis komparatif terhadap algoritma klasifikasi dalam konteks tingkat stres mahasiswa. Model akan melalui uji coba menggunakan metode SMOTE, *feature selection* dan *hyperparameter tuning*, serta dilakukan evaluasi untuk menemukan model terbaik dan kemudian mengetahui faktor penyebab tingkat stres mahasiswa menggunakan SHAP. Dengan menggunakan *dataset* yang didapatkan dari *Kaggle* mengenai tingkat stres mahasiswa,

penelitian ini diharapkan dapat mampu untuk diinterpretasikan dan digeneralisasikan bagi pihak-pihak terkait dalam memahami faktor-faktor penyebab stres mahasiswa, serta memahami teknik terbaik untuk mendapatkan hasil yang optimal dari setiap model.

2 Tinjauan Literatur

Beberapa penelitian telah menggunakan algoritma *machine learning* dalam melakukan klasifikasi tingkat stres mahasiswa, sehingga beberapa penelitian tersebut dapat menjadi bahan yang relevan dalam melakukan perbandingan ataupun referensi. Penelitian [16] melakukan klasifikasi tingkat stres pada mahasiswa akhir di STIKOM Tunas Bangsa menggunakan algoritma C4.5, dan mendapatkan hasil akurasi sebesar 87,88% dengan 20 *rules*. Penelitian sejenis yang bertujuan untuk mengklasifikasikan tingkat stres mahasiswa juga dilakukan oleh [17] yang menggunakan metode K-NN. Penelitian ini memanfaatkan *dataset* yang didapatkan dari halaman web *data.world* dengan jumlah data sebesar 16.385, dan menghasilkan nilai akurasi sebesar 91,58%.

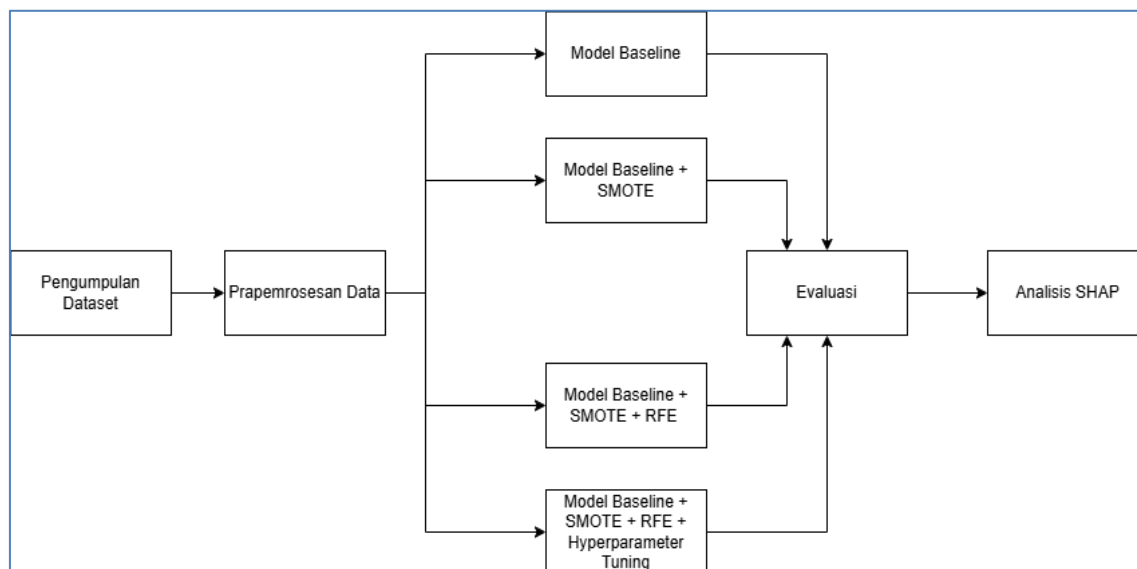
Selanjutnya, terdapat penelitian oleh [18] yang bertujuan untuk mencari metode berbasis *tree* yang paling optimal dalam mengklasifikasikan tingkat stres mahasiswa. Penelitian ini menggunakan 100 *dataset* yang didapatkan melalui penyebaran kuesioner kepada mahasiswa Universitas Negeri Surabaya dan menggunakan metode *tree family* diantaranya, *decision tree*, *random forest*, *random tree*, ID3, CHAID dan *decision stump*. Hasil penelitian menunjukkan bahwa metode yang mendapatkan akurasi paling baik yaitu metode ID3 dengan akurasi sebesar 95%. Penelitian [19] melakukan deteksi dini tingkat stres pada mahasiswa menggunakan dua metode yaitu *Iterative Dichotomiser 3* (ID3) dan K-NN untuk melakukan perbandingan akurasi. Hasil yang didapatkan kemudian dievaluasi menggunakan *cross-validation 10 folds* dan K-NN mendapatkan hasil terbaik dibandingkan dengan ID3, dimana K-NN mendapatkan akurasi 100% pada *fold 10*, sementara ID3 hanya mendapatkan akurasi 76,67% pada *fold 7*.

Seiring dengan berkembangnya pendekatan *machine learning*, klasifikasi tingkat stres dilakukan berbasis teks dengan NLP. Penelitian [20] mengembangkan model klasifikasi tingkat stres mahasiswa berbasis teks dengan menerapkan ekstraksi fitur TF-IDF dan membandingkan beberapa algoritma, yaitu *Logistic Regression*, *Naive Bayes*, *Random Forest*, dan SVM. Hasil penelitian menunjukkan bahwa *Logistic Regression* memberikan performa terbaik dengan akurasi sebesar 88,33% dan F1-score 0,88.

Berdasarkan tinjauan literatur yang telah dibahas sebelumnya, metode *machine learning* sudah banyak dilakukan dalam mengklasifikasikan tingkat stres mahasiswa, dengan hasil yang bervariasi. Namun, sebagian besar penelitian tersebut belum membahas permasalahan seperti *class imbalance* pada data, yang dapat mempengaruhi performa pada model secara signifikan terutama pada kelas minoritas. Selain itu, penelitian terdahulu umumnya belum menggunakan metode *feature selection* dan *hyperparameter tuning* secara sistematis dan belum memberikan analisis interpretabilitas pada model yang dapat menjelaskan faktor-faktor dominan penyebab stres mahasiswa secara transparan. Penelitian ini memberikan inovasi baru dengan menerapkan SMOTE, *feature selection* dan *hyperparameter tuning* pada metode *Logistic Regression*, K-NN, dan SVM untuk klasifikasi tingkat stres mahasiswa, serta memberikan analisis interpretabilitas menggunakan SHAP. Pendekatan ini dilakukan dan dibandingkan guna menemukan metode terbaik dalam klasifikasi tingkat stres, sekaligus memberikan pengetahuan mengenai faktor paling berpengaruh penyebab stres mahasiswa.

3 Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komparatif yang melakukan perbandingan performa beberapa algoritma klasifikasi pada beberapa kondisi. Penelitian ini menggunakan bahasa pemrograman *Python* versi 3.10 dengan bantuan pustaka seperti *pandas*, *scikit-learn*, *matplotlib*, dan *imbalanced-learn*. Secara garis besar, penelitian ini menerapkan beberapa tahapan yang mencakup pengumpulan data, pra-pemrosesan data, pelatihan model, evaluasi model, dan analisis hasil, yang ditampilkan pada Gambar 1.



Gambar 1 Alur penelitian

3.1 Pengumpulan Dataset

Data yang digunakan di dalam penelitian ini adalah data sekunder bersifat publik yang diperoleh dari repositori *Kaggle* dengan judul "*Student Lifestyle Dataset*". *Dataset* terdiri dari 2.000 data dengan delapan fitur yang mencakup *student id*, *study hours per day*, *extracurricular hours*, *sleep hours*, *social hours*, *physical activity hours*, *stress level*, dan *cgpa*. Label target yang digunakan dalam penelitian ini adalah tingkat stres yang diklasifikasikan menjadi tiga kategori, yaitu *high*, *moderate*, dan *low*. Data ini diperoleh melalui *Google Form survey* yang dibagikan pada tahun akademik 2023 hingga 2024 di India. Definisi dari masing-masing variabel yang terdapat dalam penelitian ditunjukkan pada Tabel 1.

Tabel 1 Definisi variabel

No	Nama	Deskripsi
1	<i>Student_ID</i>	ID unik pada setiap mahasiswa
2	<i>Study_Hours_Per_Day</i>	Jam belajar per hari
3	<i>Extracurricular_Hours_Per_Day</i>	Jam ekstrakurikuler per hari
4	<i>Sleep_Hours_Per_Day</i>	Jam tidur per hari
5	<i>Social_Hours_Per_Day</i>	Jam bersosial per hari
6	<i>Physical_Activity_Hours_Per_Day</i>	Jam beraktivitas fisik per hari
7	GPA	GPA
8	<i>Stress_Level</i>	Tingkat stress mahasiswa

3.2 Pra-pemrosesan Data

Pra-pemrosesan data merupakan tahap awal untuk membersihkan, menghilangkan, hingga mentransformasi data agar siap digunakan untuk proses pembuatan model. Beberapa proses yang terlibat di dalam *preprocessing* data adalah pencarian dan penghapusan data yang terduplikasi, pengecekan dan penanganan *missing value*, hingga melakukan normalisasi untuk menyamakan skala antar fitur menggunakan *StandardScaler*. Proses ini akan dilakukan hingga dapat diputuskan bahwa data sudah bersih dan dapat untuk digunakan untuk proses *modeling*.

3.3 Implementasi Model

Setelah melalui tahap prapemrosesan data, tahap selanjutnya adalah untuk membuat model klasifikasi. Penelitian ini akan menggunakan tiga algoritma yaitu *Logistic Regression*, K-NN, dan SVM. *Logistic Regression* digunakan karena mudah untuk menemukan makna dalam sebuah hubungan, K-NN digunakan karena sifatnya yang non-parametrik, sehingga efektif dalam menangani data dengan pola distribusi yang kompleks atau non-linear melalui pendekatan kesamaan jarak, dan

<http://sistemasi.ftik.unisi.ac.id>

SVM dipilih karena mampu untuk menangani data berdimensi tinggi. Setiap model dibangun dalam kerangka pipeline yang terdiri atas *StandardScaler* sebagai tahap normalisasi dan proses algoritma klasifikasi yang berhubungan. Penggunaan *pipeline* ini digunakan untuk memastikan bahwa seluruh tahap pemrosesan diterapkan secara konsisten dan terurut pada setiap eksperimen. Dalam pembuatan model terdapat empat skenario yang dilakukan dan kemudian dilakukan perbandingan untuk menentukan model klasifikasi dengan skenario terbaik. Keempat skenario tersebut dapat dijelaskan sebagai berikut.

3.3.1 Model Baseline

Model *baseline* merupakan model pertama yang dibangun tanpa menggunakan teknik optimasi seperti SMOTE, *feature selection*, dan *hyperparameter tuning*. Tujuan dari pembangunan model ini adalah untuk memberikan visualisasi awal mengenai performa dasar dari masing-masing algoritma klasifikasi sebelum dilakukan optimasi lebih lanjut. Adapun tahapan yang dilakukan pada model ini adalah sebagai berikut:

1. Membagi data menjadi pelatihan dan pengujian dengan rasio (80:20) menggunakan metode *train test split* dengan parameter *random_state* = 42 untuk memastikan konsistensi hasil. Sebesar 80% data akan digunakan untuk melatih model, sementara sebesar 20% data akan digunakan untuk mendapatkan evaluasi dan performa dari model.
2. Membangun *pipeline* untuk masing-masing algoritma yang terdiri atas *StandardScaler* dan *classifier* dengan parameter *default*.
3. Melatih tiga algoritma klasifikasi yang digunakan, yaitu *Logistic Regression*, K-NN, dan SVM menggunakan bahasa pemrograman *Python*.

3.3.2 Model menggunakan SMOTE

Pada skenario kedua, model dikembangkan dengan tujuan untuk mengatasi masalah ketidakseimbangan distribusi data dengan menggunakan metode *Synthetic Minority Oversampling Technique* (SMOTE) pada data pelatihan. Ketidakseimbangan data dapat berpengaruh pada model yang cenderung memprediksi pada kelas mayoritas dan kurang mengenali kelas minoritas. SMOTE beroperasi dengan meningkatkan data sintesis pada sampel kelas minoritas sehingga pembagian data label menjadi lebih seimbang dan model dapat belajar dengan lebih baik dari seluruh kelas [9]. Adapun tahapan yang dilakukan pada model ini adalah sebagai berikut:

1. Membagi data menjadi pelatihan dan pengujian dengan rasio (80:20) menggunakan metode *train test split* dengan parameter *random_state* = 42 untuk memastikan konsistensi hasil. Sebesar 80% data akan digunakan untuk melatih model, sementara sebesar 20% data akan digunakan untuk mendapatkan evaluasi dan performa dari model.
2. Menggunakan metode SMOTE pada data pelatihan untuk menangani jumlah data label yang tidak seimbang. SMOTE diinisialisasi dengan parameter *random_state* = 42 dan *n_neighbors* = 5, kemudian menerapkan *fit_resample* pada data pelatihan.
3. Melatih tiga algoritma klasifikasi yang digunakan, yaitu *Logistic Regression*, K-NN, dan SVM menggunakan bahasa pemrograman *Python*.

3.3.3 Model menggunakan SMOTE dan Feature Selection

Pada skenario ketiga, dilakukan penambahan teknik *feature selection* menggunakan metode *Recursive Feature Elimination* (RFE). Penambahan metode ini bertujuan untuk mencari fitur yang paling relevan dengan cara menghapus fitur yang memiliki pengaruh lemah terhadap model secara bertahap hingga didapatkan fitur terbaik [21]. Proses seleksi fitur ini akan dilakukan secara terpisah pada masing-masing algoritma klasifikasi sehingga model dapat dilatih menggunakan kombinasi fitur yang paling optimal. Adapun tahapan yang dilakukan pada model ini adalah sebagai berikut:

1. Membagi data menjadi pelatihan dan pengujian dengan rasio (80:20) menggunakan metode *train test split* dengan parameter *random_state* = 42 untuk memastikan konsistensi hasil. Sebesar 80% data akan digunakan untuk melatih model, sementara sebesar 20% data akan digunakan untuk mendapatkan evaluasi dan performa dari model.

2. Menggunakan metode SMOTE pada data pelatihan untuk menangani jumlah data label yang tidak seimbang dengan konfigurasi yang sama dengan parameter ($random_state = 42$, $n_neighbors = 5$).
3. Melakukan seleksi fitur menggunakan RFE dengan *logistic regression* sebagai estimator dasar dan $n_features_to_select = 4$ dari total fitur yang tersedia untuk menemukan subset atribut yang paling berpengaruh.
4. Melatih tiga algoritma klasifikasi yang digunakan, yaitu *Logistic Regression*, K-NN, dan SVM menggunakan bahasa pemrograman *Python*.

3.3.4 Model menggunakan SMOTE, Feature Selection, dan Hyperparameter Tuning

Pada skenario keempat, dilakukan optimasi pada model dengan menambahkan teknik *hyperparameter tuning* untuk mendapatkan kombinasi parameter terbaik dari masing-masing algoritma. Proses *tuning* dilakukan dengan menggunakan metode *GridSearch*, yang bekerja dengan mencoba berbagai kombinasi nilai parameter untuk menentukan komposisi yang menghasilkan performa paling optimal dari model [11]. Metode ini akan dikombinasikan dengan *StratifiedKfold* sebagai mekanisme validasi silang. Adapun tahapan yang dilakukan pada model ini adalah sebagai berikut:

1. Membagi data menjadi pelatihan dan pengujian dengan rasio (80:20) menggunakan metode *train test split* dengan parameter $random_state = 42$ untuk memastikan konsistensi hasil. Sebesar 80% data akan digunakan untuk melatih model, sementara sebesar 20% data akan digunakan untuk mendapatkan evaluasi dan performa dari model.
2. Menggunakan metode SMOTE pada data pelatihan untuk menangani jumlah data label yang tidak seimbang dengan konfigurasi yang sama dengan parameter ($random_state = 42$, $n_neighbors = 5$).
3. Melakukan seleksi fitur menggunakan RFE dengan *logistic regression* sebagai estimator dasar dan $n_features_to_select = 4$ dari total fitur yang tersedia untuk menemukan subset atribut yang paling berpengaruh.
4. Melakukan *hyperparameter tuning* menggunakan *GridSearchCV* dengan *StratifiedKfold* ($n_splits = 5$, $shuffle = True$, $random_state = 42$). Pencarian parameter yang digunakan untuk masing-masing algoritma adalah sebagai berikut, *logistic regression*: nilai regularisasi $C \in \{0,001; 0,01; 0,1; 1; 10; 100\}$ dan $solver \in \{liblinear, saga\}$, *K-Nearest Neighbors*: jumlah tetangga $n_neighbors \in \{3, 5, 7, 9\}$ dan $weights \in \{uniform, distance\}$, serta *Support Vector Machine*: nilai regularisasi $C \in \{0,1; 1; 10\}$, fungsi kernel $\in \{linear, rbf\}$, dan parameter $gamma \in \{scale, auto\}$.
5. Melatih tiga algoritma klasifikasi yang digunakan, yaitu *Logistic Regression*, K-NN, dan SVM menggunakan bahasa pemrograman *Python* menggunakan model terbaik dari setiap algoritma.

3.4 Evaluasi dan Analisis Hasil

Guna dapat membandingkan performa dari model klasifikasi, perlu dilakukan evaluasi yang komprehensif sehingga dapat melihat penggunaan metode mana yang mendapatkan performa paling baik. Dalam penelitian ini digunakan metrik evaluasi yaitu akurasi, *precision*, *recall*, dan *F1-score*. Akurasi bermanfaat untuk mengukur berapa jumlah prediksi yang benar terhadap seluruh data pengujian. *Precision* digunakan untuk memperlihatkan tingkat ketepatan model dalam memprediksi satu kelas tertentu. *Recall* digunakan untuk memperlihatkan kapabilitas model dalam menemukan seluruh data yang betul-betul termasuk kedalam kelas tersebut. Sementara itu, *F1-Score* adalah rata-rata hasil antara *precision* dan *recall* yang dimanfaatkan untuk memberikan keseimbangan antara kedua metrik tersebut [22]. Nilai evaluasi dari masing-masing algoritma pada tiap skenario eksperimen kemudian dibandingkan secara menyeluruh untuk mengetahui pengaruh penerapan teknik optimasi terhadap performa model. Melalui perbandingan tersebut, dapat diidentifikasi model mana yang menghasilkan kinerja terbaik untuk melakukan klasifikasi tingkat stres mahasiswa. Model dengan performa paling optimal selanjutnya akan digunakan untuk melakukan analisis interpretabilitas menggunakan metode SHAP.

3.5 Analisis Interpretabilitas SHAP

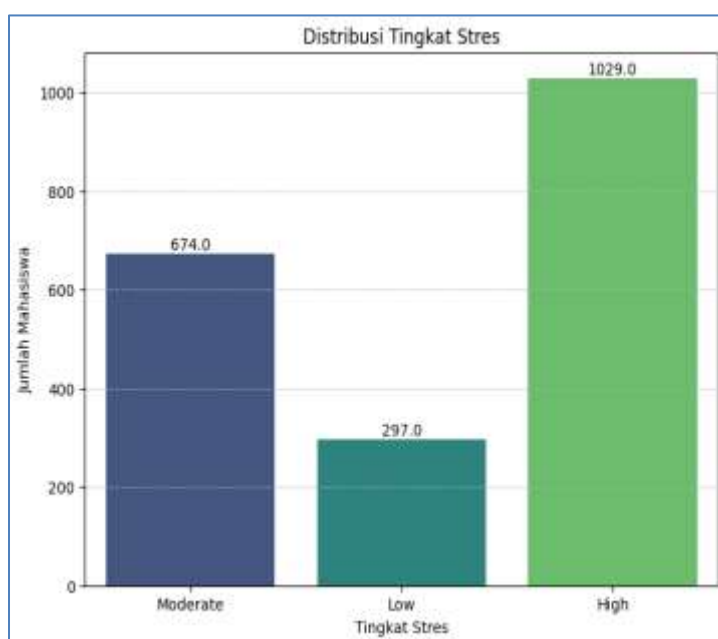
Analisis interpretabilitas dalam penelitian ini menggunakan metode *Shapley Additive Explanations* (SHAP) untuk memberikan transparansi terhadap model klasifikasi yang dihasilkan. Hal ini bertujuan untuk memahami faktor-faktor krusial yang mempengaruhi hasil prediksi model, sehingga model tidak hanya memiliki performa yang baik tetapi juga dapat dijelaskan secara logis. Metode SHAP bekerja dengan menghitung nilai kontribusi dari setiap fitur terhadap output hasil prediksi model berdasarkan konsep *Shapley value* dari teori permainan. Nilai SHAP menunjukkan seberapa besar pengaruh suatu fitur dalam meningkatkan atau menurunkan probabilitas prediksi suatu kelas [23]. Dengan hal tersebut, model yang sebelumnya bersifat *black-box* dapat dilakukan analisis sehingga hasil dapat lebih mudah dipahami. Dalam penelitian ini, analisis SHAP akan digunakan pada model terbaik yang diperoleh dari hasil evaluasi skenario eksperimen. Hasil analisis kemudian digunakan untuk memberikan interpretasi terhadap variabel-variabel gaya hidup yang paling berpengaruh terhadap tingkat stres mahasiswa.

4 Hasil dan Pembahasan

Pembahasan dimulai dengan analisis deskriptif karakteristik *Student Lifestyle dataset*, diikuti evaluasi komparatif kinerja algoritma *Logistic Regression*, K-NN, dan SVM melalui empat skenario optimasi. Analisis kemudian diperdalam dengan menggunakan metode SHAP untuk membedah kontribusi fitur dominan dalam menentukan tingkat stres mahasiswa secara transparan dan objektif.

4.1 Analisis Deskriptif Karakteristik Dataset

Bagian awal analisis dilakukan dengan meninjau karakteristik dari 2.000 data mahasiswa dalam "*Student Lifestyle Dataset*". Analisis ini akan dimulai dengan melihat persebaran distribusi dari label target yaitu *Stress Level* yang memiliki tiga kategori yaitu *low*, *moderate*, dan *high*. Dari Gambar 2 terlihat bahwa distribusi label target tidak seimbang dengan kategori stres "*high*" memiliki jumlah data terbanyak sebesar 1.029 data, diikuti dengan kategori stres "*Moderate*" sebanyak 674 data, dan kategori stres "*low*" sebanyak 297 data. Distribusi ini menunjukkan bahwa hampir setengah data berada pada kategori stres "*high*". Ketidakseimbangan ini berpotensi mempengaruhi performa model klasifikasi, karena algoritma akan cenderung lebih mudah mengenali kelas dengan jumlah data yang lebih besar dibandingkan kelas minoritas. Kondisi ini menjadi alasan utama penggunaan metode SMOTE dalam skenario pembuatan model.



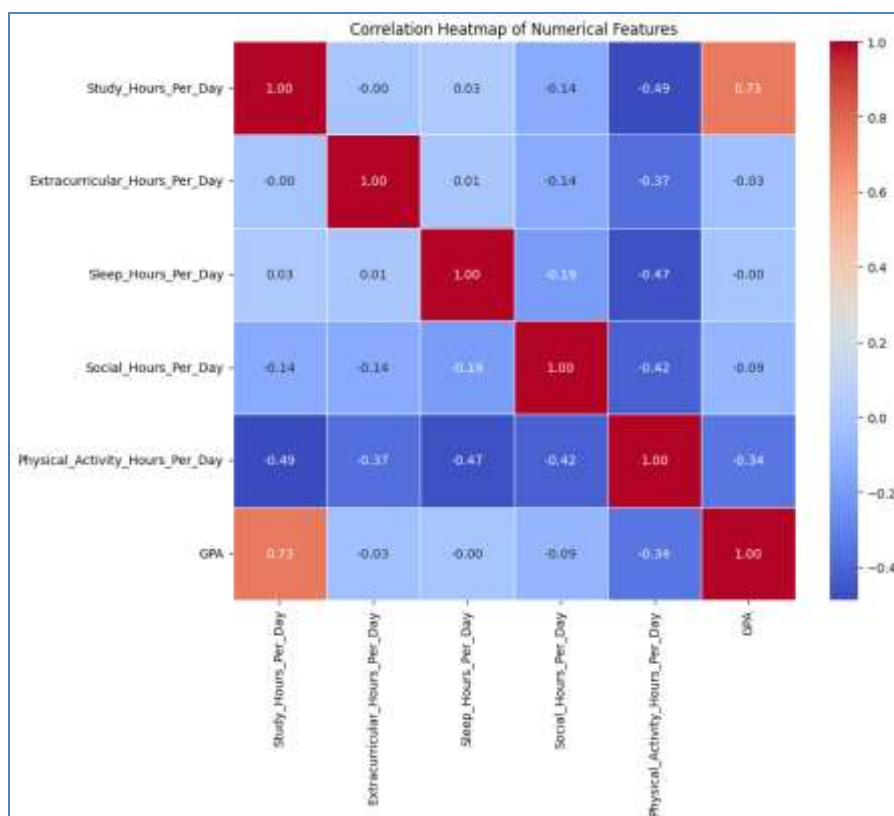
Gambar 2 Distribusi label target

Selain distribusi label target, dilakukan analisis deskriptif terhadap fitur numerik untuk memahami karakteristik data secara umum. Profil perilaku mahasiswa akan diidentifikasi melalui nilai rata-rata, minimum, dan maksimum yang terlihat pada Tabel 2. Secara rata-rata, mahasiswa menghabiskan waktu tujuh jam untuk belajar dan tujuh jam untuk tidur. Meski hal tersebut terlihat ideal, namun terdapat nilai minimum lima jam tidur dan nilai maksimum 10 jam belajar yang dapat mengindikasikan adanya kelompok mahasiswa dengan risiko stres yang lebih tinggi. Selain itu aktivitas fisik memiliki rentang yang sangat lebar 0 hingga 13 jam, di sisi lain rata-rata GPA berada di angka 3,11. Perbedaan skala antar fitur ini menjustifikasi penggunaan *StandardScaler* agar setiap variabel memiliki bobot yang setara dalam proses klasifikasi.

Tabel 2 Deskripsi data

Variabel	Mean	Min	Max
1. <i>Study_Hours_Per_Day</i>	7,48	5,00	10,00
2. <i>Extracurricular_Hours_Per_Day</i>	1,99	0,00	4,00
3. <i>Sleep_Hours_Per_Day</i>	7,50	5,00	10,00
4. <i>Social_Hours_Per_Day</i>	2,70	0,00	6,00
5. <i>Physical_Activity_Hours_Per_Day</i>	4,33	0,00	13,00
6. GPA	3,12	2,24	4,00

Untuk mengetahui korelasi awal yang terbentuk antar fitur, maka dilakukan analisis korelasi yang terdapat pada Gambar 3. Melalui hal tersebut, terlihat bahwa sebagian besar hubungan antar variabel memiliki kekuatan korelasi rendah hingga sedang. *Variabel Study_Hours_Per_Day* memiliki korelasi positif cukup kuat dengan GPA (0,73). Hal tersebut menunjukkan bahwa peningkatan waktu belajar cenderung diikuti dengan kenaikan nilai akademik. Sebaliknya, variabel *Physical_Activity_Hours_Per_Day* berkorelasi negatif dengan GPA serta dengan beberapa variabel lain seperti *Study_Hours_Per_Day*, *Sleep_Hours_Per_Day*, dan *Social_Hours_Per_Day*, yang mengindikasikan terdapat konsekuensi dalam pembagian waktu. Secara keseluruhan, tidak ditemukan korelasi tinggi antar variabel independen, sehingga tidak terdapat indikasi multikolinearitas dan masih layak digunakan dalam pemodelan.



Gambar 3 Heatmap correlation

4.2 Hasil Skenario Baseline

Berdasarkan karakteristik dataset yang telah dianalisis, tahap selanjutnya adalah melakukan pengujian model klasifikasi. Pengujian awal dilakukan pada skenario *baseline* tanpa penerapan teknik apapun, skenario ini digunakan untuk memperoleh gambaran performa dasar dari masing-masing algoritma sebelum dilakukan perlakuan lanjutan. Sebesar 80% data akan digunakan untuk pelatihan dan sebesar 20% data digunakan pengujian. Hasil evaluasi pada skenario *baseline* disajikan pada Tabel 3 dengan waktu komputasi selama 0,231 detik.

Tabel 3 Hasil model skenario 1

Algoritma Klasifikasi	Skenario	Akurasi	Precision	Recall	F1-Score	Waktu Komputasi
Logistic Regression		0,85	0,85	0,85	0,85	
K-NN	Baseline	0,90	0,89	0,90	0,89	0,231s
SVM		0,95	0,95	0,95	0,95	

Berdasarkan hasil pengujian pada skenario *baseline*, terlihat adanya perbedaan performa yang cukup signifikan. Hasil menunjukkan bahwa SVM memberikan performa tertinggi dengan tingkat akurasi, *precision*, *recall*, dan *F1-score* yang konsisten di angka 0,95. Hal ini mengindikasikan bahwa pendekatan berbasis margin pada SVM lebih mampu memisahkan pola antar kelas tingkat stres dibandingkan pendekatan sederhana. Meskipun SVM menunjukkan performa tertinggi, hasil ini masih perlu dianalisis lebih lanjut mengingat distribusi kelas dalam *dataset* tidak seimbang. K-NN juga memperlihatkan performa yang cukup baik dengan nilai akurasi 0,90, mengindikasikan bahwa pola distribusi data antar kelas memiliki kedekatan jarak yang cukup jelas dalam ruang fitur. Meskipun demikian, pendekatan ini belum sepenuhnya mampu menangkap kompleksitas batas keputusan antar kelas seperti SVM. Sementara itu, *Logistic Regression* menunjukkan hasil terendah diantara ketiga algoritma dengan hasil akurasi sebesar 0,85. Hal tersebut dapat disebabkan keterbatasan model linear dalam menangkap hubungan non linear yang mungkin terdapat dalam data gaya hidup mahasiswa.

4.3 Hasil Skenario SMOTE

Penerapan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dilakukan untuk mengatasi masalah ketidakseimbangan data yang ditemukan pada data label. Kondisi awal data menunjukkan bahwa distribusi yang tidak merata, yaitu kelas *high* sebanyak 823 sampel, sementara kelas *moderate* (547 sampel), dan *low* (230 sampel). Hal ini menunjukkan ketimpangan pada data, dan dapat beresiko untuk model berpihak hanya pada kelas mayoritas. Melalui SMOTE, sampel pada kelas *moderate* dan *low* ditingkatkan secara sintesis hingga masing-masing mencapai 823 sampel, sehingga total data latih meningkat secara signifikan dari 1.600 menjadi 2.469 sampel. Hasil skenario model dengan teknik SMOTE disajikan dalam Tabel 4.

Tabel 4 Hasil model skenario 2

Algoritma Klasifikasi	Skenario	Akurasi	Precision	Recall	F1-Score	Waktu Komputasi
Logistic Regression		0.84	0.83	0.86	0.84	
K-NN	Baseline + SMOTE	0.88	0.85	0.88	0.86	0,206s
SVM		0.95	0.93	0.95	0.94	

Berdasarkan hasil pengujian pada skenario ini menunjukkan dampak yang berbeda pada masing-masing algoritma dengan waktu komputasi sebesar 0,206 detik. Sama dengan skenario *baseline*, SVM tetap berhasil mempertahankan performa tingginya dengan tingkat akurasi sebesar 0,95 dan *F1-Score* sebesar 0,94. Hal tersebut menandakan bahwa SVM sangat stabil dalam menentukan *hyperplane* pemisah secara optimal meskipun terjadi kepadatan data akibat penambahan

sampel sintesis. Sementara itu, *Logistic Regression* dan K-NN mengalami sedikit penurunan performa dibandingkan skenario *baseline*. Penurunan ini dapat disebabkan oleh perubahan lokal yang mungkin menciptakan *overlap* antar kelas, sehingga menyamarkan batas keputusan model linear maupun berbasis jarak. Meskipun demikian, *recall* pada *Logistic Regression* meningkat menjadi 0,86. Hal ini membuktikan walaupun akurasi keseluruhan terkoreksi sedikit, model menjadi lebih sensitif dan dapat mendeteksi kelas minoritas dibandingkan skenario *baseline*. Secara umum, hasil menunjukkan bahwa teknik SMOTE efektif memberikan kestabilan pada algoritma berbasis margin seperti SVM dalam menangani perubahan distribusi data dibandingkan model linear atau berbasis jarak dalam kasus klasifikasi tingkat stres.

4.4 Hasil Skenario SMOTE dan RFE

Setelah menangani ketidakseimbangan data pada kelas label, skenario berikutnya adalah mengoptimalkan model melalui metode RFE. Proses ini dimulai dengan total 6 fitur awal yang tersedia dalam dataset. Dalam penelitian ini, jumlah fitur dikurangi menjadi 4 fitur utama. Penentuan jumlah fitur yang dipertahankan tersebut dipilih karena merupakan fitur dengan skor tertinggi yang mampu merepresentasikan pola tingkat stres mahasiswa dalam model. Fitur yang terpilih adalah *study hours per day*, *sleep hours per day*, *social hours per day*, dan *physical activity hours per day*. Eliminasi terhadap dua fitur lainnya dilakukan karena fitur tersebut memiliki nilai pengaruh yang kecil, sehingga apabila dipertahankan berpotensi menjadi gangguan bagi performa algoritma. Hasil performa untuk skenario ini terdapat pada Tabel 5 dengan waktu komputasi selama 0,529 detik.

Tabel 5 Hasil model skenario 3

Algoritma Klasifikasi	Skenario	Akurasi	Precision	Recall	F1-Score	Waktu Komputasi
<i>Logistic Regression</i>	<i>Baseline</i> +	0,83	0,82	0,85	0,83	
K-NN	SMOTE +	0,93	0,90	0,93	0,91	0,529s
SVM	RFE	0,95	0,93	0,95	0,94	

Hasil pengujian pada skenario tiga yang ditampilkan pada Tabel 5 menunjukkan bahwa kombinasi SMOTE dan RFE memberikan dampak positif yang sangat signifikan pada algoritma K-NN. Akurasi K-NN meningkat dari 0,88 pada skenario SMOTE menjadi 0,93, diikuti dengan kenaikan *precision* menjadi 0,90, dan *recall* mencapai 0,93. Peningkatan ini terjadi karena K-NN merupakan algoritma berbasis jarak yang sensitif terhadap data. Dengan membuang fitur yang kurang relevan melalui RFE, "noise" dalam data berkurang, sehingga struktur jarak menjadi lebih jelas yang mempermudah K-NN. Sementara itu, pada SVM, performa model mampu dipertahankan dengan angka 0,95, dan membuktikan bahwa model tetap mampu bekerja secara optimal meskipun dengan jumlah fitur yang lebih sedikit. Sebaliknya, *Logistic Regression* menunjukkan sedikit penurunan performa pada *precision* dan *F1-score* menjadi 0,83 dibandingkan dengan skenario SMOTE ataupun *baseline*. Hal ini menunjukkan bahwa sebagai model linear, pengurangan fitur tidak memberikan hasil yang signifikan, tetapi justru mengurangi informasi yang dibutuhkan untuk menemukan garis pemisah. Secara keseluruhan, penerapan RFE terbukti efektif dalam menyederhanakan kompleksitas model tanpa mengorbankan performa, bahkan dalam beberapa algoritma seperti K-NN justru meningkatkan performa.

4.5 Hasil Skenario SMOTE, RFE, dan Hyperparameter Tuning

Skenario final dalam penelitian ini dilakukan dengan mengintegrasikan SMOTE, RFE, dan *hyperparameter tuning* menggunakan metode *GridSearch*. Tahap ini bertujuan untuk mencari kombinasi parameter terbaik yang menghasilkan model dengan performa paling maksimal bagi ketiga algoritma yang diuji. Berdasarkan hasil pencarian parameter, konfigurasi terbaik adalah sebagai berikut: Algoritma *Logistic Regression* menggunakan $C = 1$ dan *solver* = 'saga', kemudian K-NN menggunakan $n_neighbors = 5$ dan *weights* = 'distance', serta SVM menggunakan $C = 10$, *gamma* = 'scale' dan *kernel* = 'rbf'. Penggunaan kombinasi parameter tersebut menghasilkan performa klasifikasi akhir yang disajikan dalam Tabel 6 dengan waktu komputasi selama 0,011 detik.

Tabel 6 Hasil model skenario 4

Algoritma Klasifikasi	Skenario	Akurasi	Precision	Recall	F1-Score	Waktu Komputasi
Logistic Regression	Baseline + SMOTE + RFE +	0,83	0,82	0,85	0,83	0,011s
K-NN	Hyperparameter	0,94	0,91	0,94	0,92	
SVM	Tuning	0,98	0,96	0,98	0,97	

Berdasarkan hasil pengujian skenario 4, terdapat lonjakan performa signifikan terkhusus pada algoritma SVM. Dengan menggunakan parameter terbaik, SVM berhasil mencapai akurasi sebesar 0,98 dengan *recall* 0,98 dan *F1-Score* 0,97. Penggunaan parameter yang terpilih untuk SVM memungkinkan model mampu memetakan data ke dimensi yang lebih tinggi secara non-linear yang membuat batas keputusan antar tingkat stres menjadi jauh lebih presisi dibandingkan skenario sebelumnya. Peningkatan performa juga terlihat pada K-NN yang mencapai akurasi sebesar 0,94 dengan kombinasi parameter $n_neighbors = 5$ dan $weights = 'distance'$. Penggunaan parameter ini membuat K-NN menjadi lebih sensitif terhadap kedekatan antar titik data yang relevan, sehingga proses pemilihan menjadi lebih akurat. Disisi lain, *Logistic Regression* cenderung stabil pada angka 0,83. Stagnasi hasil pada model linear ini semakin memperjelas analisis bahwa hubungan antara variabel gaya hidup terhadap tingkat stres mahasiswa cenderung bersifat non-linear. Secara keseluruhan, skenario final ini membuktikan bahwa kombinasi antara data yang seimbang, fitur yang sudah diseleksi, dan parameter yang tepat mampu menghasilkan model klasifikasi yang akurat dan efisien.

4.6 Evaluasi Hasil

Secara keseluruhan, integrasi teknik SMOTE, RFE, dan *hyperparameter tuning* yang merupakan skenario empat terbukti sebagai strategi paling optimal untuk membangun model klasifikasi pada dataset ini. Kombinasi ketiga teknik tersebut mampu meningkatkan performa model secara lebih konsisten. Melalui skenario paling optimal ini, SVM dipilih sebagai model terbaik dengan nilai akurasi sebesar 0,98 dan *F1-score* sebesar 0,97. Performa tersebut menunjukkan bahwa pendekatan berbasis margin dan dioptimasi melalui teknik yang tepat mampu memisahkan kelas tingkat stres secara efektif. Selain itu, waktu komputasi untuk skenario empat ini juga menunjukkan hasil yang paling efisien dibandingkan model dengan skenario lainnya, dengan waktu komputasi hanya 0,011 detik. Model terbaik ini, selanjutnya akan dianalisis dengan pendekatan interpretabilitas menggunakan metode SHAP. Metode ini bekerja dengan menghitung kontribusi dari setiap fitur terhadap prediksi yang dibuat oleh model. Melalui metode ini, SHAP akan membandingkan selisih dari hasil prediksi model saat suatu fitur digunakan dan saat fitur tersebut tidak digunakan, sehingga peran setiap variabel dalam menentukan tingkat stres dapat diketahui secara pasti [24].

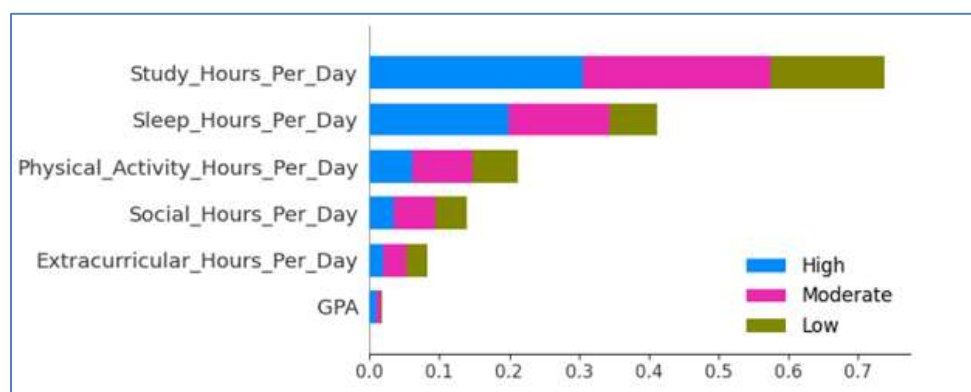
4.7 Analisis Interpretabilitas SHAP

Berdasarkan Gambar 4, fitur atau variabel *Study_Hours_Per_Day* diketahui sebagai prediktor paling dominan dalam menentukan tingkat stres mahasiswa. Dominannya kontribusi variabel ini mengindikasikan bahwa intensitas waktu belajar harian merupakan sumber tekanan utama yang secara signifikan mendorong peningkatan beban mental mahasiswa. Hasil ini selaras dengan penelitian [24] yang menunjukkan bahwa aktivitas belajar bersifat menuntut secara kognitif, semakin lama dan semakin intens aktivitas belajar dilakukan, maka akan semakin tinggi pula tingkat stres psikologis mahasiswa.

Selain faktor tuntutan akademik, variabel *Sleep_Hours_Per_Day* juga menunjukkan pengaruh yang besar dalam tahap pengambilan keputusan model. Hal ini memperlihatkan pentingnya durasi tidur sebagai mekanisme pemulihan psikologis. Sejalan dengan penelitian [25], durasi tidur terbukti memiliki hubungan signifikan dengan *perceived* stres pada mahasiswa. Penelitian tersebut menekankan bahwa kecukupan tidur berperan untuk meningkatkan self control dan resilience pada individu dalam mengurangi stres. Dengan kurangnya waktu tidur tidak hanya akan berperan langsung

terhadap kenaikan tingkat stres, tetapi juga melemahkan fungsi yang berperan dalam regulasi stres. Sementara itu, variabel gaya hidup seperti *Physical_Activity_Hours_Per_Day* dan *Social_Hours_Per_Day* menunjukkan kontribusi sedang dalam model. Hal ini mengindikasikan bahwa aktivitas fisik dan interaksi sosial turut dipertimbangkan dalam proses klasifikasi tingkat stres mahasiswa, meskipun hal tersebut tidak sekuat fitur pengaruh jam belajar dan tidur.

Temuan selanjutnya adalah fitur GPA menempati posisi terendah dalam gambar. Meskipun secara korelasi terdapat hubungan antara jam belajar dan GPA, namun model SVM mengidentifikasi bahwa tingkat stres mahasiswa lebih ditentukan oleh proses harian dibandingkan dengan hasil akademik itu sendiri. Hasil transparansi ini memberikan validasi bahwa model tidak hanya akurat secara angka, namun juga memiliki logika yang relevan dengan realitas psikologis yang dialami mahasiswa, dimana tekanan harian lebih berdampak langsung pada kesehatan mental daripada angka prestasi akademik.



Gambar 4 Grafik hasil SHAP

5 Kesimpulan

Berdasarkan analisis hasil yang telah dilakukan, disimpulkan bahwa integrasi beberapa metode yaitu penyeimbangan data menggunakan SMOTE, seleksi fitur menggunakan RFE, dan *hyperparameter tuning* dapat secara signifikan meningkatkan hasil untuk klasifikasi tingkat stres mahasiswa dengan waktu komputasi yang singkat. Algoritma SVM dipilih sebagai model terbaik dengan hasil akurasi sebesar 0,98, *precision* sebesar 0,96, *recall* sebesar 0,98, dan *F1-score* sebesar 0,97 dengan waktu komputasi yang paling efisien sebesar 0,011 detik pada kombinasi tiga teknik optimasi. Temuan utama melalui analisis SHAP berhasil mengungkapkan bahwa variabel *Study_Hours_Per_Day* dan *Sleep_Hours_Per_Day* memiliki pengaruh paling besar dalam menentukan tingkat stres mahasiswa. Hasil analisis menunjukkan bahwa semakin tinggi intensitas waktu belajar harian, maka kecenderungan tingkat stres mahasiswa juga meningkat, karena hal tersebut meningkatkan tuntutan akademik dan beban mental. Selain itu, durasi tidur yang lebih cukup justru membuat tingkat stres yang lebih rendah, sehingga kurangnya waktu tidur dapat memperburuk kondisi stres mahasiswa. Hal ini menunjukkan bahwa untuk menekan tingkat stres mahasiswa, sebaiknya difokuskan pada perbaikan pola hidup dan manajemen waktu belajar.

Referensi

- [1] E. Sheldon *et al.*, "Prevalence and Risk Factors for Mental Health Problems in University Undergraduate Students: A Systematic Review with Meta-Analysis," *J. Affect. Disord.*, Vol. 287, No. December 2020, pp. 282–292, 2021, DOI: 10.1016/j.jad.2021.03.054.
- [2] A. Abdul Rahim *et al.*, "The Pressure of Financial Status and its Effects on Public University Students Around Kuala Lumpur," *Int. J. Acad. Res. Bus. Soc. SCI.*, Vol. 13, No. 12, pp. 1545–1556, 2023, DOI: 10.6007/ijarbss/v13-i12/19568.
- [3] G. Navya and S. Sharma, "Impact of Spiritual Intelligence on Perceived Stress among Male and Female University Students," *Int. J. Bio-resource Stress Manag.*, Vol. 13, No. 1, pp. 62–68, 2022, DOI: 10.23910/1.2022.2511a.
- [4] J. Tri, V. Harefa, K. A. Manik, and Z. L. Habehan, "Stress , Anxiety , Depression and Social

- Media in Generation Z: A Scoping Review*,” Vol. 4, No. 3, pp. 174–187, 2025, DOI: 10.56922/mhc.v4i3.1362.
- [5] R. K. Djoar and A. P. M. Anggarani, “Faktor - Faktor yang mempengaruhi Stress Akademik Mahasiswa Tingkat Akhir,” *Jambura Heal. Sport J.*, Vol. 6, No. 1, pp. 52–59, 2024, DOI: 10.37311/jhsj.v6i1.24064.
- [6] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, and H. Yuanyue, “*Learning from Class-Imbalanced Data: Review of Methods and Applications*,” *Expert Syst. Appl.*, Vol. 73, pp. 220–239, 2017, DOI: 10.1016/j.eswa.2016.12.035.
- [7] J. Cai, J. Luo, S. Wang, and S. Yang, “*Neurocomputing Feature Selection in Machine Learning: A New Perspective*,” *Neurocomputing*, Vol. 300, pp. 70–79, 2018, DOI: 10.1016/j.neucom.2017.11.077.
- [8] P. Probst, “*Tunability: Importance of Hyperparameters of Machine Learning Algorithms*,” vol. 20, pp. 1–32, 2019.
- [9] M. Sulistiyono, Y. Pristyanto, S. Adi, and G. Gumelar, “*Implementasi Algoritma Synthetic Minority Over-Sampling Technique untuk Menangani Ketidakseimbangan Kelas pada Dataset Klasifikasi*,” *Sistemasi*, Vol. 10, No. 2, p. 445, 2021, DOI: 10.32520/stmsi.v10i2.1303.
- [10] A. M. R. Armaya, “*Pengaruh Feature Selection dan Feature Extraction dalam Peningkatan Akurasi Klasifikasi Kebakaran Hutan*,” *JuTI “Jurnal Teknol. Informasi*,” Vol. 3, No. 1, p. 13, 2024, DOI: 10.26798/juti.v3i1.1039.
- [11] I. Muhamad Malik Matin, “*Hyperparameter Tuning Menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware*,” *Multinetics*, Vol. 9, No. 1, pp. 43–50, 2023, DOI: 10.32722/multinetics.v9i1.5578.
- [12] V. Hassija, V. Chamola, A. Mahapatra, A. Singal, D. Goel, and K. Huang, *Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence*. Springer US, 2024. DOI: 10.1007/s12559-023-10179-8.
- [13] H. Lamane, L. Mouhir, R. Moussadek, B. Baghdad, O. Kisi, and A. El, “*Interpreting Machine Learning Models Based on Shap Values in Predicting Suspended Sediment Concentration*,” *Int. J. Sediment Res.*, Vol. 40, No. 1, pp. 91–107, 2025, DOI: 10.1016/j.ijsrc.2024.10.002.
- [14] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, “*A Novel KNN Algorithm with Data-Driven k Parameter Computation*,” *Pattern Recognit. Lett.*, Vol. 109, pp. 44–54, 2018, DOI: 10.1016/j.patrec.2017.09.036.
- [15] J. Cervantes, F. Garcia-lamont, L. Rodríguez-mazahua, and A. Lopez, “*Neurocomputing A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends*,” *Neurocomputing*, No. xxxx, 2019, DOI: 10.1016/j.neucom.2019.10.118.
- [16] Anggi Trifani, Agus Perdana Windarto, and Hendry Qurniawan, “*Penerapan Data Mining Klasifikasi C4.5 dalam Menentukan Tingkat Stres Mahasiswa Akhir*,” *Jural Ris. Rumpun Ilmu Tek.*, Vol. 1, No. 2, pp. 91–105, 2022, DOI: 10.55606/jurritek.v1i2.414.
- [17] S. Anisa, A. Komarudin, and E. Ramadhan, “*Sistem Klasifikasi untuk Menentukan Tingkat Stress Mahasiswa secara umum Menggunakan Metode K-Nearest Neighbors*,” *J. Inform. Teknol. dan Sains*, Vol. 6, No. 3, pp. 568–578, 2024, DOI: 10.51401/jinteks.v6i3.4317.
- [18] A. Anjani and Y. Yamasari, “*Klasifikasi Tingkat Stres Mahasiswa menggunakan Metode berbasis Tree*,” *J. Informatics Comput. SCI.*, Vol. 05, pp. 83–89, 2023.
- [19] A. Aldi, S. R. C. Nursari, and F. Maspiyanti, “*Deteksi Dini Tingkat Stres pada Mahasiswa menggunakan Metode Iterative Dichotomiser 3 dan K-Nearest Neighbour*,” *J. Informatics Adv. Comput.*, Vol. 1, No. 1, pp. 1–7, 2020.
- [20] I. Rusdiansyah, R. Pangestu, D. Azalia, M. F. Zhafran, and F. Saputra, “*Integrasi Model Klasifikasi Tingkat Stress Mahasiswa berbasis Natural Language Processing*,” Vol. 4, No. 4, pp. 7823–7831, 2025.
- [21] A. R. I. Pratama, S. A. Latipah, and B. N. Sari, “*Optimasi Klasifikasi Curah Hujan menggunakan Support Vector Machine (SVM) dan Recursive Feature Elimination (RFE)*,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, Vol. 7, No. 2, pp. 314–324, 2022, DOI: 10.29100/jupi.v7i2.2675.
- [22] G. M. F. Id, “*Challenges in the Real World use of Classification Accuracy Metrics: From Recall and Precision to the Matthews Correlation Coefficient*,” pp. 1–27, 2023, DOI:

- 10.1371/journal.pone.0291908.
- [23] S. M. Lundberg and S. Lee, “Predictions,” No. Section 2, pp. 1–10, 2017.
- [24] M. D. Work and M. Students, “*Mentally Demanding Work and Strain: Effects of Study Duration on Fatigue, Vigor, and Distress in Undergraduate Medical Students*,” 2023.
- [25] S. A. Tafoya, V. Aldrete-cortez, F. Tafoya-ramos, C. Fouilloux-morales, and D. Claudia, “*Sleep and Perceived Stress: An Exploratory Mediation Analysis of the Role of Self-Control and Resilience among University Students*,” 2023.