

Analisis Performa Support Vector Machine (SVM) dengan Hyperparameter Tuning untuk Klasifikasi Data Stress Monitoring Siswa

Feibe Lawalata¹, Eko Sedyono¹

¹*Program Studi Magister Sistem Informasi, Universitas Kristen Satya Wacana, Salatiga, Indonesia
Email:lawalatafeibe@gmail.com*

Abstract— Stress is a significant issue in student mental health and has a direct impact on academic performance. Machine learning based stress detection offers an efficient approach for automatically monitoring psychological conditions. This study evaluates the performance of the Support Vector Machine (SVM) algorithm in classifying student stress levels using a public dataset consisting of psychological and self-reported variables. Two models were compared a baseline SVM with a linear kernel and a tuned SVM optimized using GridSearchCV. The evaluation employed accuracy, precision, recall, F1-score, confusion matrix, PCA visualization, and margin analysis. The results show that both models achieve identical aggregate performance, with an accuracy of 84.54% and a macro F1-score of 0.85. The PCA visualization indicates cluster patterns that are largely linearly separable, while the margin and decision-boundary analyses demonstrate that hyperparameter tuning only produces minimal shifts in the separating hyperplane. Furthermore, the contingency table reveals no discordant prediction pairs ($b = c = 0$), indicating that the McNemar test cannot be applied and that both models produce identical predictions across all samples. These findings highlight that hyperparameter tuning does not necessarily improve performance, particularly when the dataset exhibits a predominantly linear structure. Moreover, the results demonstrate that the linear SVM is an efficient model that can serve as a lightweight backend for mobile-based student stress-monitoring applications.

Keywords— Support Vector Machine, Hyperparameter Tuning, Stress Monitoring, Confusion Matrix, McNemar Test

Abstrak— Stres merupakan isu penting dalam kesehatan mental siswa yang berdampak langsung pada performa akademik. Deteksi stres berbasis machine learning menawarkan pendekatan yang efisien untuk memantau kondisi psikologis secara otomatis. Penelitian ini mengevaluasi performa algoritma Support Vector Machine (SVM) dalam klasifikasi tingkat stres siswa menggunakan dataset publik yang terdiri atas variabel psikologis dan self-report. Dua model dibandingkan, yaitu SVM baseline dengan kernel linear dan model hasil hyperparameter tuning menggunakan GridSearchCV. Evaluasi dilakukan menggunakan akurasi, presisi, recall, F1-score, confusion matrix, visualisasi PCA, dan analisis margin. Hasil penelitian menunjukkan bahwa kedua model memiliki performa agregat yang identik dengan akurasi 84,54% dan macro F1-score 0,85. Visualisasi PCA mengindikasikan pola kluster yang cenderung linear separable, sementara analisis margin dan batas keputusan menunjukkan bahwa proses tuning hanya menghasilkan pergeseran hyperplane yang minimal. Selain itu, tabel kontingensi prediksi memperlihatkan tidak adanya pasangan discordant ($b = c = 0$), sehingga Uji McNemar tidak dapat diterapkan. Kondisi ini menunjukkan bahwa kedua model menghasilkan prediksi yang identik pada seluruh sampel. Temuan tersebut menegaskan bahwa hyperparameter tuning tidak selalu meningkatkan performa, terutama pada dataset dengan struktur yang relatif linear. Selain itu, hasil ini menunjukkan bahwa SVM linear dapat menjadi model yang efisien untuk diterapkan sebagai backend dalam aplikasi monitoring stres siswa berbasis mobile.

Kata Kunci— Support Vector Machine, Hyperparameter Tuning, Stress Monitoring, Confusion Matrix, McNemar Test.

I. PENDAHULUAN

Kesehatan mental merupakan salah satu aspek penting dalam kehidupan siswa, di mana stres menjadi faktor dominan yang dapat memengaruhi prestasi akademik, motivasi belajar, serta kualitas hidup sehari-hari [1]. Berbagai penelitian menunjukkan sebagian besar siswa mengalami tingkat stres sedang hingga tinggi akibat beban akademik, tuntutan sosial, maupun tekanan finansial [2]. Kondisi ini menimbulkan dampak serius seperti gangguan konsentrasi, kecemasan, hingga penurunan capaian akademik [3].

Perkembangan machine learning telah membuka peluang baru dalam mendeteksi dan memantau kondisi psikologis secara otomatis. Salah satu algoritma yang banyak digunakan dalam klasifikasi data psikologis adalah Support Vector Machine (SVM), yang terbukti efektif dalam analisis sinyal

fisiologis seperti EEG dan HRV [4]. Algoritma SVM digunakan dalam penelitian karena terbukti efektif untuk analisis data berdimensi menengah hingga tinggi, terutama ketika ukuran sampel tidak terlalu besar. Namun demikian, performa SVM sangat dipengaruhi oleh pemilihan hyperparameter seperti kernel, nilai C, dan gamma [5]. Selain itu berbagai studi terbaru menunjukkan bahwa SVM memberikan performa yang kompetitif pada deteksi stres, klasifikasi sinyal fisiologis, serta prediksi kesehatan mental. Hasil penelitian [2] SVM menjadi model paling stabil pada monitoring stres berbasis konteks, sedangkan penelitian [3] menunjukkan bahwa SVM unggul dalam mendeteksi stres menggunakan sinyal EEG. Oleh karena itu, berbagai studi menekankan pentingnya hyperparameter tuning untuk meningkatkan akurasi model [6]. Meskipun banyak penelitian melaporkan bahwa hyperparameter tuning dapat meningkatkan

performa SVM, terdapat keterbatasan pada studi-studi sebelumnya, yaitu:

Fokus penelitian lebih menyoroti keberhasilan tuning dalam meningkatkan akurasi, namun jarang membahas kasus ketika tuning tidak memberikan perbaikan signifikan.

Sebagian besar evaluasi hanya menggunakan metrik agregat (akurasi, presisi, recall, F1-score), sementara uji statistik seperti *McNemar Test* jarang digunakan untuk memastikan perbedaan performa secara signifikan [7].

Berdasarkan celah tersebut, penelitian ini bertujuan untuk membangun dan membandingkan model baseline SVM dengan model hasil hyperparameter tuning menggunakan *GridSearchCV*. Perbandingan dilakukan melalui metrik agregat (akurasi, presisi, recall, F1-score) serta uji *McNemar* untuk mengidentifikasi perbedaan prediksi pada level instance. Selain itu penelitian ini menyertakan bukti visual berupa PCA 2D scatter plot, histogram margin dan decision boundary untuk menjelaskan karakteristik pemisahan kelas yang mendekati *linear separable*, serta alasan kenapa tuning tidak meningkatkan performa. Temuan dari penelitian ini diharapkan dapat memberikan pemahaman yang komprehensif mengenai efektivitas tuning pada dataset serta implikasinya terhadap pengembangan aplikasi monitoring stres siswa.

II. METODOLOGI

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komputasional. Algoritma yang digunakan yaitu *supervised learning*. *Supervised learning* adalah algoritma komputer yang di latih berdasarkan data masukan yang telah diberi label untuk keluaran tertentu. Pendekatan ini berbasis pelatihan dan unggul dalam klasifikasi [7].

A. Support Vector Machine (SVM)

Support vector machine (SVM) memiliki pengaruh yang kuat terhadap masalah klasifikasi. Kemampuan utama SVM yaitu, menemukan garis dengan margin terbesar untuk kedua grup yang memisahkan titik-titik data dari berbagai kelas dalam ruang berdimensi tinggi [8]. *Support vector machine* (SVM) akan digunakan untuk mengklasifikasikan tingkat stres siswa, serta mengevaluasi pengaruh hyperparameter tuning terhadap akurasi model. *Support vector machine* (SVM) dipilih karena kemampuannya dalam menangani masalah klasifikasi dengan data berdimensi tinggi dan menghasilkan margin pemisah (*hyperplane*) yang optimal.

B. Dataset

Dataset yang digunakan di ambil dari kaggle dengan nama *student stress monitoring datasets*, yang memuat data tingkat stres siswa. Dataset memiliki 21 fitur yang merupakan kombinasi dari data fisiologis, psikologis, perilaku dan faktor lingkungan. Struktur data tersebut sangat sesuai dengan karakteristik SVM yang mampu membentuk *hyperplane* optimal pada ruang fitur yang berdimensi tinggi tanpa memerlukan jumlah data yang sangat besar. Atribut target (*stress level*) terbagi dalam 3 kategori yaitu :

- Stres 0 masuk dalam kategori stres rendah
- Stres 1 masuk dalam kategori stres sedang
- Stres 2 masuk dalam kategori stres tinggi

PCA digunakan untuk mereduksi 21 fitur menjadi dua komponen utama. Perhitungannya mengacu pada rumus berikut :

- Rumus PCA

$$C = \frac{1}{n-1} X^T X$$

$$Cv = \lambda v$$

$$Z = XW$$

Tahap preprocessing data dilakukan untuk memastikan kualitas data dan kesiapan model [9], semua fitur dinormalisasi menggunakan *StandardScaler* sehingga nilai rata-rata menjadi 0 dan nilai standar deviasi 1.

- *Standardization*, mengubah data sehingga memiliki rata – rata = 0 dan standar deviasi = 1 (*Z-score scaling*)

$$x' = \frac{x - \mu}{\sigma} \quad [10]$$

Normalisasi penting untuk dilakukan karena SVM sensitif dengan perbedaan skala fitur, jika tidak dinormalisasi fitur dengan nilai besar dapat mendominasi penentuan *hyperplane*. Pada bagian pembagian data, dataset dibagi menjadi data latih (80%) dan data uji (20%) kemudian di olah menggunakan *scikit-learn*.

C. Pemodelan Support Vector Machine (SVM)

Pemodelan menggunakan *support vector machine* (SVM) mendukung berbagai *kernel*. Parameter penting yang dioptimasi yaitu *C* (*Regularization Parameter*) yang mengontrol *trade-off* antara memperbesar margin pemisah (*maximum-margin hyperplane*), meminimalkan kesalahan klasifikasi pada data latih [11] [12]. *Kernel* yang berfungsi memetakan data dari ruang input ke ruang fitur berdimensi lebih tinggi agar menjadi lebih mudah untuk dipisahkan secara linear [13] [5]. Serta *Gamma* yang berfungsi sebagai parameter untuk menentukan seberapa jauh pengaruh satu titik data dalam menentukan batas keputusan, khususnya pada RBF dan *Polynomial kernel* [14].

- Model SVM membentuk sebuah *hyperplane* yang memaksimalkan margin antara kelas :

$$f(x) = wTx + b$$

$$y = \text{sign}(wTx + b)$$

- Margin Histogram

$$\text{margin} = \frac{2}{\|w\|}$$

- Kernel RBF

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|_2)$$

- Kernel Polynomial

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$$

D. Optimasi Model Hyperparameter Tuning dengan GridSearchCV

Optimasi model dilakukan menggunakan metode *Hyperparameter Tuning*, salah satunya *GridSearchCV* yang tersedia pada pustaka *scikit-learn*. Metode *GridSearchCV* memungkinkan pencarian kombinasi parameter terbaik dengan validasi silang (*k-fold cross validation*) untuk memilih konfigurasi terbaik, sehingga meningkatkan kemampuan generalisasi model. Keunggulan metode ini yaitu untuk mengurangi *overfitting*, pencarian sistematis dan *reproducible* [10]. Pendekatan ini banyak digunakan karena sederhana, mudah direplikasi dan memberikan jaminan bahwa seluruh kombinasi dalam ruang pencarian telah diuji secara menyeluruh.

Beberapa studi menunjukkan bahwa *GridSearchCV* masih menjadi baseline penting dalam optimasi hyperparameter dan sering digunakan sebagai acuan pembandingan untuk menilai efektivitas. Metode optimasi seperti *Random Search*, *Bayesian Optimization*, maupun pendekatan berbasis metaheuristik [16].

Pada tahap tuning model, penelitian ini menggunakan metode *GridSearchCV* untuk melakukan pencarian kombinasi hyperparameter terbaik secara sistematis. *GridSearchCV* bekerja dengan menguji seluruh kombinasi parameter pada ruang pencarian (*parameter grid*) dan mengevaluasi performanya menggunakan *k-fold cross-validation*, sehingga hasil tuning dapat direplikasi secara akurat. Performa SVM sangat dipengaruhi oleh pemilihan hyperparameter, definisi ruang parameter secara eksplisit sebagai berikut :

- Nilai C (*Regularization Parameter*)

Parameter C berfungsi sebagai regularization parameter yang mengendalikan trade-off antara margin yang lebar dan penalti terhadap misclassification pada data latih. Nilai C yang dievaluasi pada penelitian ini :

$$C \in \{0.01, 0.1, 1, 10, 100\} \quad [17]$$

- Jenis Kernel yang diuji

Kernel berfungsi sebagai fungsi pemetaan yang menentukan bentuk pemisahan kelas pada ruang fitur, baik secara linear maupun non-linear, dengan memproyeksikan data ke ruang berdimensi lebih tinggi jika diperlukan. Pada penelitian ini, tiga jenis kernel dievaluasi untuk menguji karakteristik pemisahan kelas pada dataset, yaitu :

$$\text{kernel} \in \{\text{linear}, \text{rbf}, \text{poly}\} \quad [17]$$

- Nilai Gamma (*Kernel Coefficient*)

Parameter *gamma* mengontrol radius pengaruh setiap titik data dalam membentuk fungsi keputusan pada kernel *non-linear*.

Nilai gamma yang besar membuat model lebih sensitif terhadap titik-titik yang berada sangat dekat dengan margin, sedangkan nilai gamma yang kecil menghasilkan permukaan keputusan yang lebih halus. Parameter ini hanya berperan pada kernel non-linear seperti RBF dan polynomial. Nilai gamma yang dievaluasi dalam penelitian ini yaitu :

$$\gamma \in \{\text{scale}, \text{auto}, 0.001, 0.01, 0.1, 1\} \quad [17]$$

- Degree (khusus untuk kernel polynomial)

Pada kernel polynomial, parameter *degree* menentukan orde polinomial yang digunakan untuk memetakan data ke ruang fitur berdimensi lebih tinggi. Semakin besar nilai *degree*, semakin kompleks bentuk kurva pemisahan yang dapat dibentuk model. Dalam penelitian ini, dua nilai *degree* dievaluasi untuk menilai apakah peningkatan kompleksitas pemetaan memberikan perbaikan performa, yaitu :

$$\text{degree} \in \{2, 3\} \quad [17]$$

Seluruh kombinasi hyperparameter pada parameter grid tersebut dievaluasi menggunakan *5-fold cross-validation* dengan metrik utama F1-score macro, sehingga performa model dapat dinilai secara seimbang pada ketiga kelas target. Penyajian ruang parameter secara eksplisit ini penting untuk menjamin transparansi dan reproducibility proses tuning sesuai standar penelitian machine learning terkini.

E. Evaluasi Kinerja

Pada tahap evaluasi, kinerja model diuji menggunakan data uji yang tidak pernah digunakan pada proses pelatihan maupun tuning, sehingga hasil evaluasi benar-benar merepresentasikan kemampuan generalisasi model. Metrik evaluasi dalam penelitian ini, yaitu :

- Akurasi (*accuracy*), proporsi prediksi benar terhadap seluruh data dengan rumus :

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad [10]$$

- Presisi (*precision*), proporsi prediksi positif yang benar berguna untuk mengurangi false positive.

$$\text{Precision} = \frac{TP}{TP+FP} \quad [10]$$

- Recall (*sensitivity*), kemampuan model mendeteksi seluruh data positif, penting untuk mengurangi false negative.

$$\text{Recall} = \frac{TP}{TP+FN} \quad [10]$$

- F1-Score, ringkas trade-off antara precision dan recall, berguna saat distribusi kelas tidak seimbang.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad [10]$$

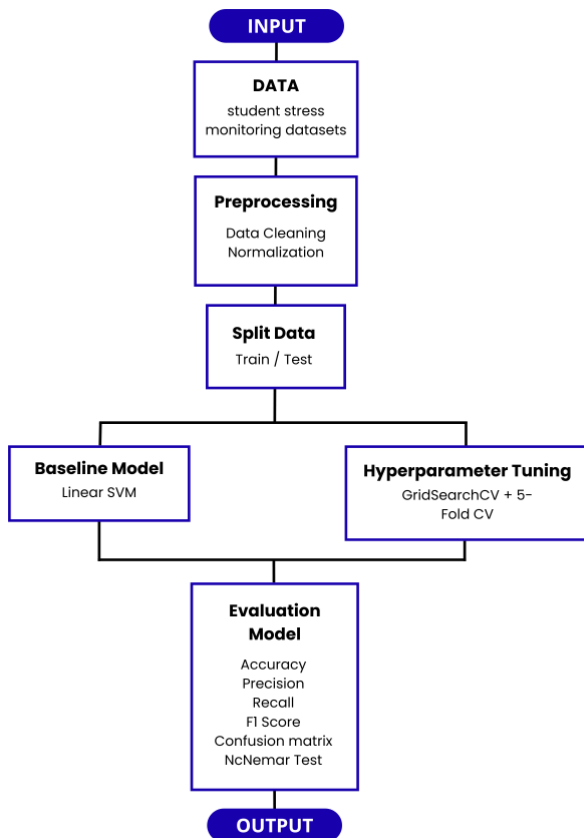
- *Confusion matrix*, digunakan untuk memberikan gambaran detail terkait distribusi prediksi model pada setiap kelas. Selain menunjukkan tingkat akurasi keseluruhan, confusion matrix juga memberikan pola kesalahan model dalam membedakan antar kelas [15].

Setelah menjalankan metrix performa, hal yang perlu diuji selanjutnya adalah uji statistik. Uji statistik digunakan untuk mengetahui apakah perbedaan hasil antar model signifikan. Salah satu uji yang banyak digunakan yaitu *McNemar Test*.

- Rumus McNemar Test

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c}$$

Secara keseluruhan. Alur penelitian dapat dilihat pada Gambar 1.



Gbr 1. Machine learning Framework untuk *student stress monitoring classification*

III. HASIL DAN PEMBAHASAN

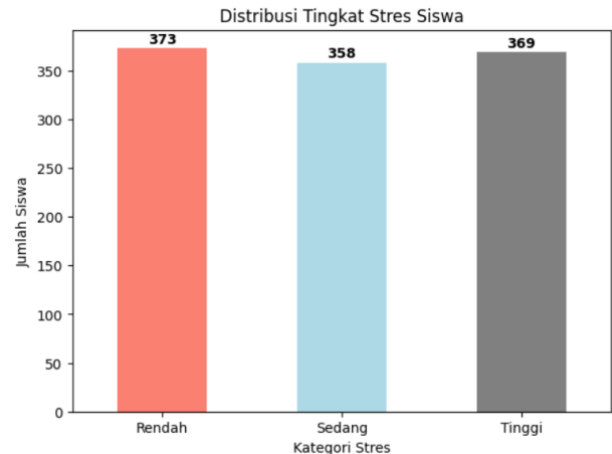
A. Preprocessing Data

Hasil dari preprocessing data yang dilakukan pada *dataset student stress monitoring* menunjukkan, total kolom berjumlah

21 dan baris berjumlah 1.100. Data types integer, *stress level* terbagi atas 3 yaitu :

- Stres 0 masuk dalam kategori stres rendah
- Stres 1 masuk dalam kategori stres sedang
- Stres 2 masuk dalam kategori stres tinggi

Visualisasi data tingkat stres siswa, dapat dilihat pada Gambar 2.



Gbr 2. Visualisasi data tingkat stres siswa

Dari gambar 2 bisa disimpulkan bahwa total stres level 0 (stres rendah) berjumlah 373 siswa, total stress level 1 (stres sedang) berjumlah 358 siswa dan stres level 2 (stres tinggi) berjumlah 369 siswa. Dalam dataset tidak terdapat *missing data*, dataset dibagi menjadi data latih (80%) dan data uji (20%).

B. Evaluasi Model

Proses tuning hyperparameter pada *support vector machine* (SVM) dilakukan menggunakan *GridSearchCV* dengan *5-fold cross validation*, ruang pencarian mencakup parameter *C*, *Gamma*, dan *Kernel*. Hasil menunjukkan konfigurasi terbaik ada pada *C=1*, *gamma=1* dengan kernel linear dan nilai rata-rata *F1-macro* = 0.8944 pada data latih, dapat dilihat pada Gambar 3.

Fitting 5 folds for each of 32 candidates, totalling 160 fits
 GridSearch Best params: {'C': 1, 'gamma': 1, 'kernel': 'linear'}
 GridSearch best cv score (f1_macro): 0.8944514029656399

Gbr 3. Hasil konfigurasi terbaik GridSearchCV (*C* = 1, *Gamma* = 1, *Kernel* = Linear, *F1-Score* = 0.89).

Hasil dari SVM *baseline* yang disajikan dalam bentuk *classification report*, dapat di lihat pada Gambar 4.

Hasil dari *hyperparameter tuning* yang disajikan dalam bentuk *classification report*, dapat di lihat pada Gambar 5 . Dimana nilai yang didapatkan identik dengan model SVM *baseline*.

Hasil evaluasi model *baseline* (*linier SVM parameter default*) dengan model *hyperparameter tuning* menunjukkan hasil yang

identik pada data uji. Nilai akurasi, presisi, recall dan F1-score yang diperoleh dari 2 model dapat dilihat pada Tabel 1.

```
=== Baseline (linear SVM) Evaluation on TEST ===
Accuracy : 0.8454545454545455
Precision: 0.8462293414696161
Recall : 0.8455955955955957
F1 Score : 0.8453530265610802
```

```
Classification report:
      precision    recall  f1-score   support

0         0.87        0.81        0.84         74
1         0.82        0.86        0.84         72
2         0.85        0.86        0.86         74

 accuracy         0.85         220
macro avg         0.85         0.85         0.85         220
weighted avg         0.85         0.85         0.85         220
```

Gbr 4. Classification report baseline SVM (macro F1 = 0.85).

```
=== Tuned (GridSearch SVM) Evaluation on TEST ===
Accuracy : 0.8454545454545455
Precision: 0.8462293414696161
Recall : 0.8455955955955957
F1 Score : 0.8453530265610802
```

```
Classification report:
      precision    recall  f1-score   support

0         0.87        0.81        0.84         74
1         0.82        0.86        0.84         72
2         0.85        0.86        0.86         74

 accuracy         0.85         220
macro avg         0.85         0.85         0.85         220
weighted avg         0.85         0.85         0.85         220
```

Gbr 5. Classification report hyperparameter tuning (macro F1 = 0.85).

TABEL 1. PERBANDINGAN PERFORMA MODEL SVM BASELINE DAN TUNING

Model	Presisi	Recall	F1-Score	Akurasi
SVM Baseline	0,846	0,845	0,845	84,54%
SVM + GridSearchCV	0,846	0,845	0,845	84,54%

Hasil yang didapati pada uji *confusion matrix* terhadap 2 model dapat dilihat pada gambar 6 dan 7.

```
Confusion matrix:
[[60  9  5]
 [ 4 62  6]
 [ 5  5 64]]
```

Gbr 6. Hasil Uji Confusion Matrix baseline

Confusion matrix pada kedua model menunjukkan hasil yang identik, hal ini memperlihatkan bahwa baik model SVM *baseline* maupun model hasil dari *hyperparameter tuning* memiliki distribusi prediksi yang sama pada ketiga kelas.

```
Confusion matrix:
[[60  9  5]
 [ 4 62  6]
 [ 5  5 64]]
```

Gbr 7. Hasil Uji Confusion Matrix GridSearchCV

Uji *McNemar* digunakan untuk mengevaluasi perbedaan prediksi antara model SVM *baseline* dan model hasil *hyperparameter tuning* pada tingkat instance. Namun, hasil tabel kontingensi pada Gambar 8 menunjukkan bahwa tidak terdapat pasangan discordant ($b = 0$ dan $c = 0$), yaitu tidak ada sampel yang diprediksi benar oleh satu model tetapi salah oleh model lainnya.

```
b (baseline correct, tuned incorrect) = 0
c (baseline incorrect, tuned correct) = 0
No discordant pairs (b+c=0): McNemar test not applicable.
```

Gbr 8. Hasil Uji McNemar perbedaan prediksi antara baseline SVM dan tuned SVM ($b+c=0$)

Sebanyak 186 sampel diklasifikasikan benar oleh kedua model, dan 34 sampel diklasifikasikan salah oleh kedua model. Dengan kondisi ini, syarat utama penerapan Uji *McNemar*, yaitu $b + c > 0$, tidak terpenuhi sehingga Uji *McNemar* tidak dapat diterapkan. Oleh karena itu, tidak dapat disimpulkan adanya perbedaan statistik pada prediksi kedua model.

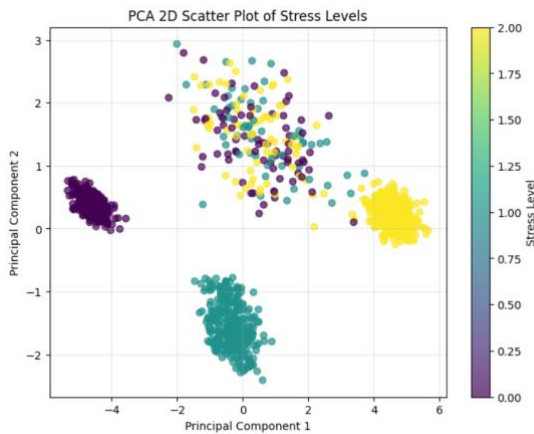
Ketiadaan pasangan discordant ini mengindikasikan bahwa kedua model menghasilkan prediksi yang sepenuhnya identik pada seluruh sampel, konsisten dengan hasil metrik agregat yang juga identik, seperti akurasi dan F1-score macro. Hal ini menunjukkan bahwa *hyperparameter tuning* tidak memberikan perubahan baik pada performa makro maupun pada keputusan klasifikasi pada tingkat instance.

Fenomena ini selaras dengan karakteristik dataset yang cenderung *linear separable*, sebagaimana ditunjukkan oleh visualisasi PCA Gambar 9 yang memperlihatkan kluster data yang relatif terpisah. Analisis margin dan batas keputusan pada Gambar 10 juga menunjukkan bahwa mayoritas sampel berada jauh dari *hyperplane*, sehingga perubahan parameter pada model tuned hanya menghasilkan pergeseran *hyperplane* yang minimal dan tidak cukup untuk mengubah prediksi.

Visualisasi PCA 2D menunjukkan bahwa ketiga kelas stres (rendah, sedang, tinggi) membentuk pola kluster yang relatif terpisah. Kelas stres rendah tampak mengelompok pada sisi kiri bawah, kelas stres sedang berada di bagian tengah dengan sedikit tumpang tindih, sedangkan kelas stres tinggi terkonsentrasi pada sisi kanan. Pola ini mengindikasikan bahwa distribusi data memiliki karakteristik yang hampir *linear separable*, sehingga batas pemisahan antar kelas dapat dibentuk oleh *hyperplane* yang relatif sederhana.

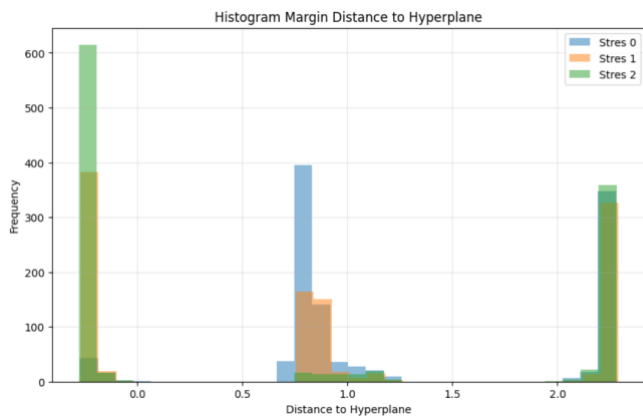
Struktur kluster yang jelas ini menunjukkan bahwa model SVM linear secara alami mampu memisahkan kelas tanpa memerlukan transformasi *non-linear* dari kernel RBF maupun polynomial. Bukti visual ini mendukung temuan bahwa *tuning*

hyperparameter tidak banyak mengubah arah atau posisi batas keputusan secara makro.



Gbr 9. PCA 2D Scatter Plot.

Meskipun terdapat sedikit area overlap antar kelas, analisis tabel kontingensi memperlihatkan bahwa tidak ada sampel yang berubah sisi keputusan ($b = c = 0$), sehingga Uji McNemar tidak dapat diterapkan. Hal ini menegaskan bahwa kedua model menghasilkan prediksi yang identik pada seluruh instance, konsisten dengan struktur data yang hampir linear dan stabil terhadap perubahan parameter.

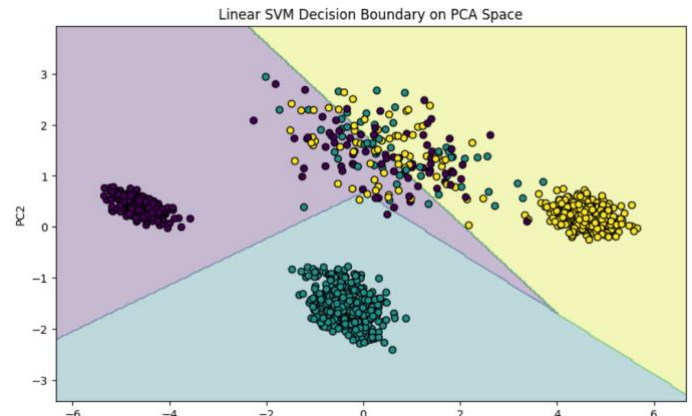


Gbr 10. Histogram Margin Distance

Histogram margin pada Gambar 10 memperlihatkan distribusi jarak titik-titik data terhadap hyperplane untuk ketiga kelas. Terlihat bahwa sebagian besar sampel berada jauh dari margin, menunjukkan bahwa prediksi model relatif stabil dan tidak sensitif terhadap perubahan kecil pada hyperparameter. Pada kelas dengan klaster yang lebih jelas, seperti stres rendah dan stres tinggi, margin yang lebar menyebabkan tuning tidak menghasilkan perubahan keputusan sehingga akurasi dan F1-score macro juga tidak berubah.

Meskipun terdapat sebagian kecil titik yang berada dekat margin khususnya pada kelas stres sedang yang memiliki tumpang tindih lebih besar analisis tabel kontingensi menunjukkan bahwa tidak ada sampel yang berpindah sisi

keputusan antara model baseline dan model hasil tuning ($b = c = 0$). Dengan demikian, tuning tidak memengaruhi prediksi pada tingkat instance, dan Uji McNemar tidak dapat diterapkan. Hal ini menegaskan bahwa perubahan parameter pada proses tuning tidak cukup untuk menggeser batas keputusan maupun mengubah hasil klasifikasi, konsisten dengan karakteristik data yang bersifat hampir linear.



Gbr 11. Linear SVM Decision Boundary.

Gambar 11 menampilkan decision boundary model SVM linear pada ruang PCA. Batas keputusan yang terbentuk sangat konsisten dengan pola klaster pada Gambar 9, menunjukkan bahwa ketiga kelas dapat dipisahkan secara efektif menggunakan hyperplane linear. Area berwarna merepresentasikan wilayah keputusan masing-masing kelas, dan bentuk wilayah yang luas serta mulus menunjukkan bahwa model tidak memerlukan transformasi non-linear yang kompleks.

Visualisasi ini menjadi bukti bahwa struktur data mendukung pemisahan linear, sehingga tuning hyperparameter—terutama parameter yang relevan untuk kernel non-linear seperti γ —tidak memberikan keuntungan tambahan. Meskipun terdapat area dengan sedikit tumpang tindih antar kelas, perbandingan antara model baseline dan model hasil tuning menunjukkan bahwa perubahan hyperparameter hanya menghasilkan pergeseran boundary yang sangat minimal. Analisis tabel kontingensi mengonfirmasi bahwa tidak ada sampel yang berpindah sisi keputusan ($b = c = 0$), sehingga Uji McNemar tidak dapat diterapkan dan kedua model menghasilkan prediksi yang identik pada seluruh instance.

Dengan demikian, Gambar 11 memperkuat temuan bahwa tuning tidak memengaruhi keputusan klasifikasi maupun performa agregat, selaras dengan karakteristik data yang bersifat hampir linear.

IV. KESIMPULAN

Penelitian ini menganalisis performa algoritma *Support Vector Machine* (SVM) dalam klasifikasi tingkat stres siswa dengan membandingkan model *baseline* (*linear SVM default*) dan model hasil hyperparameter tuning menggunakan

GridSearchCV. Hasil menunjukkan bahwa kedua model menghasilkan performa agregat yang identik dengan akurasi sebesar 84,54% dan macro F1-score sebesar 0,85. Visualisasi PCA memperlihatkan pola kluster yang relatif terpisah dan cenderung linear separable, sehingga konfigurasi linear SVM telah bekerja secara optimal pada struktur data ini.

Analisis margin dan batas keputusan turut mengonfirmasi bahwa sebagian besar sampel berada jauh dari margin, menyebabkan tuning tidak memberikan perubahan yang cukup besar untuk memengaruhi performa makro. Selain itu, tabel kontingensi prediksi menunjukkan bahwa tidak terdapat pasangan discordant ($b = c = 0$), sehingga Uji *McNemar* tidak dapat diterapkan. Kondisi ini menegaskan bahwa kedua model menghasilkan prediksi yang identik pada tingkat instance, dan tuning hyperparameter tidak memberikan perubahan baik pada performa agregat maupun pada keputusan klasifikasi.

Secara praktis, temuan tersebut menunjukkan bahwa model *baseline* dengan parameter *default* sudah memadai untuk dataset yang berkarakteristik linear dan berukuran menengah, sehingga dapat menjadi pilihan yang efisien dalam implementasi sistem monitoring stres siswa. Pendekatan ini relevan untuk aplikasi real-time, di mana efisiensi komputasi dan kecepatan inferensi menjadi pertimbangan penting.

Penelitian ini memiliki beberapa keterbatasan. Pertama, dataset yang digunakan merupakan data publik berbasis gabungan variabel psikologis dan self-report, bukan data sensor fisiologis seperti EEG atau detak jantung yang lazim digunakan dalam penelitian stress sensing. Oleh karena itu, generalisasi model dapat ditingkatkan apabila diuji pada dataset lapangan yang lebih representatif dan multikanal. Kedua, ruang hyperparameter dalam *GridSearchCV* masih terbatas pada kombinasi nilai C , γ , dan jenis kernel tertentu, sehingga belum mengeksplorasi potensi peningkatan performa melalui ruang parameter yang lebih luas maupun metode optimasi yang lebih adaptif.

Untuk penelitian selanjutnya, disarankan untuk:

1. Menguji model pada dataset lapangan yang menggabungkan data psikologis, perilaku, serta data sensor fisiologis untuk meningkatkan generalisasi.
2. Membandingkan SVM dengan algoritma lain seperti Random Forest, XGBoost, dan model Deep Learning yang mungkin lebih sesuai untuk pola non-linear.
3. Mengeksplorasi teknik optimasi hyperparameter yang lebih canggih, seperti Bayesian Optimization, Random Search, atau Hyperband, dengan ruang parameter yang lebih luas.

Dengan demikian, penelitian ini memberikan kontribusi dalam memahami keterbatasan efektivitas *hyperparameter tuning* pada dataset yang bersifat hampir linear, sekaligus menyediakan dasar praktis bagi pengembangan sistem monitoring stres berbasis SVM yang efisien dan dapat diimplementasikan pada berbagai platform termasuk aplikasi mobile.

DAFTAR PUSTAKA

- [1] World Health Organization, "Stress," 21 Februari 2023. [Online]. Available: <https://www.who.int/news-room/questions-and-answers/item/stress>.
- [2] M. S. I. Ovi, J. Hossain, M. R. A. Rahi and F. Akter, "Protecting Student Mental Health with a Context-Aware Machine Learning Framework for Stress Monitoring," IEEEAccess, 2025.
- [3] R. Gupta, A. M. Alam and P. Agarwal, "Modified Support Vector Machine for Detecting Stress Level Using EEG Signals," Computational Intelligence and Neuroscience, vol. 2020, no. 1687-5265, pp. 1-14, 2020.
- [4] J. A. Ilemobayo, O. Durodola, O. Alade, O. J. Awotunde, A. T. Olanrewaju, A. Osinuga, A. Ogungbire, I. E. Odezuligbo, A. Ifeanyi, D. Ogunbiyi and O. E. Edu, "Hyperparameter Tuning in Machine Learning : A Comprehensive Review," Journal of Engineering Research and Reports, vol. 26, no. 2582-2926 , pp. 388-395, 2024.
- [5] V. T. Saradhi, "A Study on Hyperparameter Tuning in Support Vector Machines and its Impact on Model Accuracy," Global Journal of Engineering Innovations & Interdisciplinary Research, vol. 5, no. 1, pp. 1-6, 2025.
- [6] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," Journal of Machine Learning Research, vol. 7, pp. 1-30, 2006.
- [7] . S. GEDAM and S. PAUL, "A Review on Mental Stress Detection Using Wearable Sensors and Machine Learning Techniques," IEEEAccess, vol. 9, pp. 84045-84066, 2021.
- [8] L. W. Rizkallah, "Optimizing SVM hyperparameters for satellite imagery classification using metaheuristic and statistical techniques," International Journal of Data Science and Analytics, no. <https://doi.org/10.1007/s41060-025-00762-7>, 2025.
- [9] G. Al-Tameemi, J. Xue, I. H. Ali and S. Ajit, "A Hybrid MachineLearning Approach for Predicting Student Performance Using Multi-class Educational Datasets," in The 14th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS2024), Belgium, 2024.
- [10] Y. Bengio and J. Bergstra, "Random search for hyper-parameter optimization," Journal of Machine Learning Research, vol. 13, pp. 281-305, 2012.
- [11] "An Extensive Analysis of Machine Learning Techniques With Hyper-Parameter Tuning by Bayesian Optimized SVM Kernel for the Detection of Human Lung Disease," IEEEAccess, vol. 12, pp. 97752-97770, 2024.
- [12] C. W. Hsu, . C. C. Chang and . C. J. Lin, "A Practical Guide to Support Vector Classification," Department of Computer Science National Taiwan University, pp. 1-20, 2003.
- [13] B. Schölkopf and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, The MIT Press, 2001.
- [14] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," Association for Computing Machinery, vol. 2, no. 3, pp. 1-27, 2011.
- [15] G. Lapalme and M. Sokolova, "A systematic analysis of performance measures for classification tasks," Information Processing & Management Elsevier, vol. 45, no. 4, pp. 427-437, 2009.
- [16] Y.-J. Chang, Y.-L. Lin, and P.-F. Pai, "Support Vector Machines with Hyperparameter Optimization Frameworks for Classifying Mobile Phone Prices in Multi-Class," Electronics, vol. 14, no. 11, p. 2173, 2025
- [17] R. Sholehurrohman and G. Valentine, "Penilaian Kualitas Udara dan Polusi Menggunakan Algoritma Support Vector Machine," J. Saintekom, vol. 15, no. 1, pp. 56-68, 2025