

# Implementasi Metode Machine Learning Untuk Mendeteksi Tingkat Stres Manusia Berdasarkan Kualitas Tidur

Almira Van Fadhila<sup>1</sup>, Jasmine Athira Azzahra<sup>2</sup>, Kiana Rizki Tsaniyah Zulkarnain<sup>3</sup>, Nabilah Dwi Lathifah<sup>4</sup>,  
Salwa Ziada Salsabiila<sup>5</sup>, Nurul Chamidah<sup>6</sup>  
Program Studi S1 Informatika / Fakultas Ilmu Komputer  
Universitas Pembangunan Nasional Veteran Jakarta  
Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12460, Indonesia.  
Email: almiravf@upnvj.ac.id<sup>1</sup>, jasmineaa@upnvj.ac.id<sup>2</sup>, kianartz@upnvj.ac.id<sup>3</sup>, nabilahl@upnvj.ac.id<sup>4</sup>,  
salwazs@upnvj.ac.id<sup>5</sup>, nurul.chamidah@upnvj.ac.id<sup>6</sup>

**Abstrak.** Hubungan antara gangguan stres dengan kualitas tidur memiliki keterkaitan satu sama lainnya. Kualitas tidur yang buruk dapat memicu timbulnya stres. Stres yang timbul memicu buruknya kualitas tidur suatu individu. Hal ini dapat menimbulkan ketidakseimbangan jasmani dan rohani. Penelitian ini bertujuan untuk mendeteksi tingkat stres manusia berdasarkan kualitas tidur dengan mengimplementasikan metode *machine learning* yang fokus dengan tiga macam algoritma klasifikasi yaitu K-Nearest Neighbour, *Decision Tree*, dan SVM. Pengumpulan data menggunakan dataset *Human Stres Detection in and through Sleep* (SaYoPillow) berasal dari *open source* Kaggle. Data berjumlah 630 dengan 8 fitur dan 1 label. Dengan 80% *training data* dan 20% *testing data*, Pada penelitian ini menghasilkan nilai hasil *Accuracy*, *Precision*, *Recall*, *F1-score* pada metode KNN dan SVM sebesar 100% dan pada *Decision Tree* menghasilkan *Accuracy* sebesar 99% dan *Precision*, *Recall*, *F1-score* sebesar 100%. Penelitian ini menguji ketepatan model klasifikasi dalam mendeteksi seseorang terprediksi ke dalam kategori tingkat *stres* apa yang telah ditentukan.

**Kata Kunci:** Kualitas Tidur, Mendeteksi Tingkat Stres, Algoritma Klasifikasi, *Accuracy*.

## 1 Pendahuluan

### 1.1 Latar Belakang

Stres termasuk ke dalam suatu gejala yang dialami individu secara fisik maupun non fisik (emosional) sebagai hasil reaksi dari proses penyesuaian diri pada perubahan situasi yang mungkin sedang terjadi. Stres menjadi bagian alami dari kehidupan manusia namun akan merusak kesehatan apabila berlangsung lama dan dalam kategori berat. Hormon adrenalin dan kortisol yang dilepaskan oleh tubuh saat menghadapi stres membuat detak jantung dan tekanan darah meningkat, pernapasan menjadi lebih cepat, dan otot menjadi tegang. Umumnya stres dialami setiap individu, baik dewasa maupun anak-anak dengan cara yang berbeda-beda. Tubuh bisa merespon stres dengan reaksi positif maupun sebaliknya dengan beragam faktor risiko timbulnya stres dari setiap individu. Menurut jenisnya, stres dibagi menjadi 3 bagian yaitu Eustres (stres yang positif), neustres (stres yang netral), dan distres (stres yang negatif). Distres sendiri terbagi menjadi 2 jenis klasifikasi yaitu stres yang kronis dan stres yang akut. Stres yang akut dalam waktu singkat menghilang namun sering muncul. Stres kronis membutuhkan durasi waktu yang lama untuk menghilang [1].

Tidur adalah suatu keadaan diluar kesadaran individu yang menyebabkan reaksi terhadap lingkungan sekitar individu menurun atau hilang [2]. Tidur sebagai salah satu aktivitas yang banyak dilakukan memiliki dampak yang baik pada kesehatan fisik dan non fisik (mental).

Kualitas tidur yang baik dapat menjaga produktivitas kegiatan sehari-hari dan meminimalisir terjadinya stres. Sebaliknya, kualitas tidur yang buruk secara bertahap dapat meningkatkan gejala depresi atau gangguan kecemasan. Penelitian di Fakultas Kedokteran Universitas Malahayati Lampung melaporkan bahwa terdapat hubungan keterkaitan antara variabel kualitas tidur yang buruk dengan variabel stres, dengan prevalensi mahasiswa yang mengalami stres sebesar 78,4%, terdiri dari gejala stres yang ringan 8%, stres yang sedang 16%, stres yang berat 34,6%, dan stres berat sekali 19,8% [3].

## 1.2 Studi Literatur

Sebelum melakukan penelitian, peneliti melakukan studi literatur terhadap 3 jurnal penelitian. Berikut penelitian terkait yang telah dilakukan, antara lain:

Jurnal pertama disusun oleh Risa, Diva Fardiana, dkk [4] yang melakukan penelitian Implementasi Metode *Naive Bayes* untuk Mendeteksi Stres Siswa Berdasarkan *Tweet* pada Sistem *Monitoring Stres*. Pada penelitian ini proses pengumpulan data *tweet* yang diambil secara acak pada beberapa akun di *Twitter*. Kata kunci yang digunakan mengenai stres yang kemudian data diklasifikasikan oleh pakar menjadi tiga tingkat stres yaitu stres tingkat rendah, sedang dan tinggi. Jumlah data latih yang digunakan adalah 90 data *tweet* yang terdiri dari 30 data *tweet* dengan kelas stres ringan, 30 data *tweet* dengan kelas stres sedang dan 30 data *tweet* dengan kelas stres berat. Serta menggunakan 4 data sebagai data uji. Pada pengujian dengan metode *Naive Bayes* ini menghasilkan nilai akurasi fitur mencapai angka 75%.

Jurnal kedua disusun oleh Nurrochman, M. M. dkk [5] yang melakukan penelitian Implementasi *Machine Learning* untuk Mendeteksi Unsur Depresi pada *Tweet* Menggunakan Metode *Naive Bayes*. Pada penelitian ini proses pengumpulan data *tweet* menggunakan *library tweepy* pada *python* dan akses API dari *twitter* dengan format akhir berupa file csv. Dilakukan beberapa perbandingan data latih dan data uji yaitu dengan perbandingan 50:50, 60:40, 70:30, 80:10 dan 90:10. Dan hasil akurasi tertinggi didapat pada perbandingan 90:10 dengan nilai paling baik *accuracy* 74%, dan dengan data validasi sebanyak 100 *tweet* menghasilkan nilai akurasi 70%.

Jurnal ketiga disusun oleh Menaka, A. dkk [6] yang melakukan penelitian *Recognition of Human Mental Stres Using Random Forest and SVM*. Penulis menggunakan metode *Random Forest* dan *Support Vector Machine* (SVM). Pada penelitian proses pengumpulan data didapatkan dengan menggabungkan beberapa dataset gelombang ECG dan RESP. Dataset dilatih sesuai dengan nilai input yang diberikan. Sistem ini dapat mengidentifikasi tiga keadaan stres yaitu *stressed*, *half stressed*, dan *relaxed*. *Simple Classification Rule* dapat mendeteksi dan memeriksa 60% stres.

## 1.3 Usulan Metode

Berdasarkan permasalahan pada riset penelitian-penelitian sebelumnya, ditemukan adanya keterkaitan antara variabel stres dengan variabel kualitas tidur. Oleh karena itu, peneliti memutuskan untuk melakukan riset mengenai pendeteksian tingkat stres yang terjadi pada setiap individu berdasarkan kualitas tidur dengan mengimplementasikan metode *machine learning* dengan fokus 3 jenis algoritma pengklasifikasian sebagai perbandingan tingginya tingkat akurasi.

Perbandingan tingkat akurasi dari tiga jenis metode *machine learning* yaitu algoritma KNN, SVM, *Decision Tree* berdasarkan kasus tersebut, dan melakukan evaluasi menggunakan *Cross Validation*. Yang menjadi batasan dalam penelitian ini adalah membandingkan hasil evaluasi dan mendapat akurasi tertinggi pada ketiga algoritma klasifikasi *machine learning* tersebut.

## 1.4 Data yang Digunakan

Pada penelitian ini digunakan dataset "*Human Stres Detection in and through Sleep (SaYoPillow)*" yang berasal dari situs *Kaggle.com*.

## 2 Tinjauan Pustaka

### 2.1 Human Stres

Stres merupakan respon yang dihasilkan oleh tubuh kita saat mengalami tekanan atas suatu situasi dan peristiwa tertentu. Hal ini dapat dipicu ketika dalam keadaan hal baru atau mengancam diri yang dapat menyebabkan stres. Tidur adalah proses fisiologi yang berputar dan bergantian, dengan periode jaga yang lebih lama [7]. Fungsi fisiologis dan psikologis seseorang dipengaruhi oleh siklus tidur. Dalam sebuah penelitian mengatakan bahwa kualitas tidur yang buruk menjadi faktor risiko yang dapat meningkatkan stres.

### 2.2 Klasifikasi

Klasifikasi adalah metode prediktif yang melakukan pembelajaran terhadap data-data yang sudah ada sehingga menghasilkan suatu model yang digunakan untuk memprediksi data-data baru [8]. Metode klasifikasi memiliki variabel target kategori yang dimana terjadipenggolongan atau pengelompokan berdasarkan suatu keadaan tertentu. Proses dalam menemukan sebuah model atau fungsi menjelaskan dan mencirikan konsep atau kelas data [9].

### 2.3 Algoritma *K-Nearest Neighbor (K-NN)*

Algoritma *K-Nearest Neighbor (K-NN)* merupakan suatu metode *supervised Learning* pada *Machine Learning* dimana hasil dari kejadian *query* yang diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN untuk mengklasifikasikan objek baru berdasarkan atribut dan training sample dihitung menggunakan jarak *euclidean* [10]. Jarak *Euclidean* dapat dihitung menggunakan rumus berikut ini :

$$Dist(a, b) = \sqrt{\sum^n (x_i - y_i)^2}, \quad (1)$$

Keterangan :

D(a,b) : jarak *Euclidean* antara data a dan data b

x : data *training*

y : data *testing*

n : jumlah atribut individu antara 1 s.d. n

i : Atribut individu antara 1 sampai dengan n

Tahapan algoritma K-NN :

1. Menentukan nilai K.
2. Hitung jarak *Euclidean* ke input data pelatihan.
3. Urutkan hasil jarak *Euclidean* dalam urutan menaik.
4. Kumpulkan objek klasifikasi tetangga terdekat sesuai dengan parameter input K.
5. Cari kelas mayoritas berdasarkan hasil prediksi.

### 2.4 Algoritma *Support Vector Machine (SVM)*

Algoritma Support Vector Machine atau SVM adalah algoritma pembelajaran terawasi untuk pembelajaran mesin yang dapat digunakan untuk menyelesaikan masalah klasifikasi dan regresi. SVM bekerja berdasarkan SRM atau *Structural Risk Minimization* dan bertujuan untuk memproses data menjadi *hyperplane* atau *decision boundary*, membagi ruang input menjadi dua kategori. Teori SVM dimulai dengan mengelompokkan kasus-kasus linier yang dapat dipisahkan oleh *hyperplanes* dan dibagi menurut kategorinya [11].

Konsep SVM dimulai dengan masalah klasifikasi dua kelas yang membutuhkan set pelatihan positif dan negatif. SVM akan berusaha mendapatkan *hyperplane* (separator) terbaik untuk memisahkan kedua kelas dan memaksimalkan margin kedua kelas [11].

Secara umum, algoritma SVM hanya digunakan untuk klasifikasi menjadi dua kategori. Selanjutnya, telah dikembangkan untuk klasifikasi lebih dari dua kelas atau beberapa kelas. Untuk kasus multi-kelas yang sering, pada dasarnya ada dua pendekatan yang digunakan dalam algoritma SVM, yaitu teknik OVO dan OVA. Teknologi

OVO adalah salah satunya dengan membandingkan teknik klasifikasi kelas ke kelas lain. Namun, Teknologi algoritma OVA terdiri dari: Bandingkan kelas dengan semua Selain kelasnya sendiri masih dalam satu kesatuan [12].

Parameter OVO terdapat N kelas yang dimana N tersebut akan menjadi  $N(N-1)/2$  *decision boundary*. Hasil dari *decision boundary* tersebut merupakan hasil dari pencarian *hyperplane* dari satu kelas dengan setiap kelas lainnya [13].

Parameter OVA terdapat N kelas yang dimana N tersebut akan menjadi N *decision boundary*. *Decision boundary* yang dihasilkan merupakan hasil dari pencarian *hyperplane* dari setiap kelas ke-*i* dengan sisa kelas yang lain [13].

## 2.5 Algoritma Decision Tree

Algoritma *Decision Trees* (DTs) adalah metode non-parametrik *supervised learning* yang digunakan untuk menyelesaikan permasalahan klasifikasi dan regresi. Model yang dibuat bertujuan untuk memprediksi nilai variabel dengan mempelajari *decision rules* sederhana yang dihasilkan berdasarkan data fitur. *Decision Tree* menggunakan aturan keputusan. Pada suatu kasus tertentu dapat dikatakan semakin kompleks aturan keputusannya, maka semakin fit modelnya.

Model *decision tree* merupakan sebuah *tree* yang terdiri dari *root node*, *internal node* dan *terminal node*. *Root node* (simpul akar) dan *internal node* adalah variabel/fitur, node terminal adalah label kelas. Dalam klasifikasi, kueri data akan menelusuri simpul akar dan simpul internal sampai simpul terminal tercapai. Label Kelas data kueri berdasarkan tag simpul internal. Dalam pohon keputusan, *root node* adalah simpul akar. Untuk menentukan akarnya dapat menggunakan *gain* tertinggi untuk setiap atribut atau berdasarkan nilai indeks entropi terendah [14]. Terlebih dahulu, cari nilai entropi dan dilanjutkan menghitung nilai gain dengan menggunakan rumus sebagai berikut :

$$\text{Entropy (S)} = \sum_{i=0}^n - p_i * \log^2 p_i \quad (2)$$

$$\text{Gain (S,A)} = \sum_i^n \frac{|s_i|}{|s|} * \text{Entropy (S}_i) \quad (3)$$

## 2.6 Pengujian Metode Evaluasi

### 2.6.1 Confusion Matrix

*Confusion matrix* merupakan salah satu metode yang digunakan untuk mengukur kinerja dan keakuratan suatu metode klasifikasi [15]. *Confusion Matrix* tersusun atas empat sel yang diberi label *TP*, *FP*, *FN* dan *TN*. Kinerja metode *K-Nearest Neighbour*, *SVM*, dan *Decision Tree* dapat dihitung menggunakan *Confusion Matrix*.

Tabel 1. *Confusion Matrix*

	Prediksi Tingkat Stres Level 0	Prediksi Tingkat Stres Level 1	Prediksi Tingkat Stres Level 2	Prediksi Tingkat Stres Level 3	Prediksi Tingkat Stres Level 4
Aktual Tingkat Stres Level 0	<i>True Negative (TN)</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>
Aktual Tingkat Stres Level 1	<i>False Positive (FP)</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>	<i>False Positive (FP)</i>	<i>False Positive (FP)</i>
Aktual Tingkat Stres Level 2	<i>True Negative (TN)</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>
Aktual Tingkat Stres Level 3	<i>True Negative (TN)</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>
Aktual Tingkat Stres Level 4	<i>True Negative (TN)</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>	<i>True Negative (TN)</i>

Keterangan :

TP = *True Positive*, yaitu banyaknya data positif yang diklasifikasikan dengan benar oleh sistem untuk kelas i.

TN = *True Negative*, banyaknya data negatif yang diklasifikasikan oleh sistem dengan benar untuk kelas i.

FP = *False Positive*, yaitu banyaknya data kelas ke-i yang diprediksi positif salah diklasifikasikan oleh sistem

FN = *False Negative*, yaitu banyaknya data kelas ke-i yang diprediksi negatif salah diklasifikasikan oleh sistem

### 2.6.2 Classification Report

*Accuracy* menunjukkan seberapa baik klasifikasi benar terprediksi tingkat stres atau dengan tingkat stres pada data aktual. *Precision* menunjukkan rasio prediksi menyatakan dengan benar seseorang terprediksi kedalam kategori tingkat stres yang telah dibuat dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Recall* (Sensitivitas) menunjukkan rasio memprediksi dengan benar seseorang terprediksi kedalam kategori tingkat stres dibandingkan dengan keseluruhan data yang benar positif. Hasil kombinasi perhitungan nilai *Precision* dan *Recall* akan dijadikan sebagai nilai pengukuran (*F1-score*) [16]. Untuk menentukan nilai *accuracy*, *precision*, *recall* dan *f1-score* digunakan rumus ditunjukkan pada persamaan sebagai berikut :

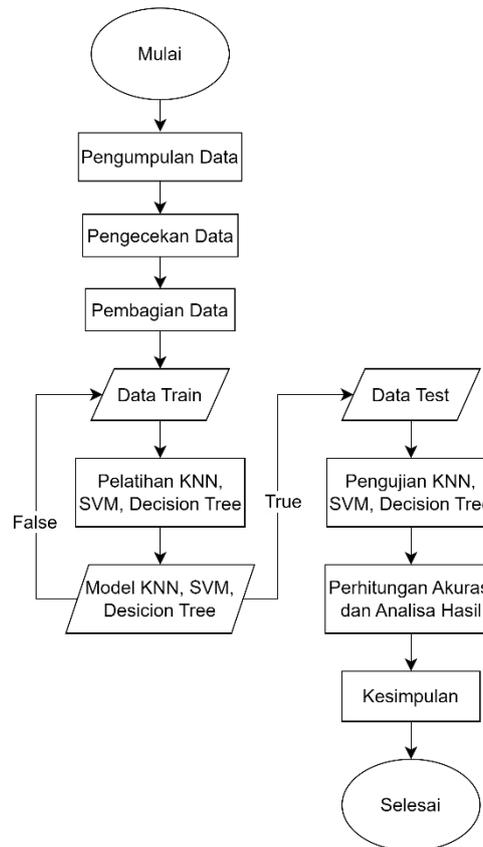
$$Accuracy = \frac{TP}{(TP + TN + FP + FN)} \times 100\% \quad (4)$$

$$Precision = \frac{TP}{(FP + TP)} \times 100\% \quad (5)$$

$$Recall = \frac{TP}{FN + TP} \times 100\% \quad (6)$$

$$F1-Score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (7)$$

### 3 Metodologi Penelitian



**Gambar 1.** Alur Penelitian

Pada penelitian ini memerlukan beberapa tahap penyusunan yang dapat mendukung kebenaran serta keberhasilan dari penelitian ini. Proses penelitian dimulai dari proses pengumpulan data yang digunakan, melakukan pembagian data yaitu data training dan data testing, melakukan pemodelan, dan terakhir kesimpulan. Berikut penjabaran alur penelitian secara lengkap:

#### 3.1 Pengumpulan Data

Pada penelitian ini digunakan dataset “*Human Stress Detection in and through Sleep*” yang berasal dari situs Kaggle.com, untuk membantu dalam memahami hubungan antara stres dan kualitas tidur. Kemudian, peneliti melakukan studi literatur. Yang dimana dilakukan beberapa cara seperti mencari jurnal, paper, artikel maupun website yang berkaitan dengan penelitian.

#### 3.2 Pengecekan Data

Pada langkah ini, data yang diperoleh akan di cek apakah ada *missing value* atau tidak, serta pengecekan duplikasi data. Jadi, sebelum memasuki langkah selanjutnya untuk pembagian data, dataset sudah bersih.

#### 3.3 Pembagian Data

Dalam dataset tersebut terdapat sebanyak 630 data. Yang dimana dari 630 tersebut peneliti sorting kembali menjadi 80% data yang akan dijadikan data training dan 20% data akan dijadikan data testing.

#### 3.4 Pemodelan Data

Model penelitian ini menggunakan metode klasifikasi *Machine Learning* dengan 3 algoritma yaitu *K-Nearest Neighbour*, *Decision Tree*, dan *SVM*.

### 3.5 Evaluasi dan Hasil

Pada tahap ini merupakan tahap pengujian dengan melihat hasil akurasi pada proses klasifikasi menggunakan ketiga algoritma yang digunakan yaitu *K-Nearest Neighbour*, *Decision Tree*, dan *SVM*. Dan dievaluasi menggunakan metode *Confusion Matrix* dan *Classification Report*.

## 4 Hasil dan Pembahasan

### 4.1 Implementasi

Data yang digunakan untuk melakukan penelitian ini menggunakan data dari situs *open source Kaggle.com* yang berjudul "*Human Stres Detection in and through Sleep*". untuk membantu dalam memahami hubungan antara stres dan kualitas tidur. Dengan menggunakan *database Smart-Yoga Pillow (SaYoPillow.csv)* akan melihat hubungan antara parameter-parameter dari snoring range, respiration rate, body temperature, limb movement rate, blood oxygen levels, eye movement, sleeping hours, heart rate and Stres Levels (0-rendah/normal, 1 – sedang rendah, 2-sedang, 3-sedang tinggi, 4-tinggi) diusulkan dalam bantuan algoritma klasifikasi *SVM*, *K-Nearest Neighbour* dan *Decision Tree* yang digunakan akan membantu dalam memahami hubungan antara stres dan tidur untuk sepenuhnya mewujudkan gagasan "Tidur Cerdas".

Dilakukan pengolahan data menggunakan pemrograman python, data yang diambil *SaYo Pillow.csv* ini memiliki 630 data. Dari 630 tersebut peneliti melakukan pembagian data menjadi 80% data yang akan dijadikan data training dan sisanya dengan persentase 20% dipakai untuk data testing. Dengan pembagian data training dan testing secara random. Melakukan pengecekan *missing value* pada dataset *SaYoPillow.csv*, dan tidak memiliki nilai yang *missing (missing value)* serta antara label kelas sudah *balanced* satu sama lain dengan memiliki 20% persentase *balanced*.

Tabel 2. Dataset

snoring rate	respiration rate	body temperature	limb movement	blood oxygen	eye movement	sleeping hours	heart rate	Stress levels
93.8	25.68	91.84	16.6	89.84	99.6	1.84	74.2	3
91.64	25.104	91.552	15.88	89.552	98.88	1.552	72.76	3
60	20	96	10	95	85	7	60	1
85.76	23.536	90.768	13.92	88.768	96.92	0.768	68.84	3
48.12	17.248	97.872	6.496	96.248	72.48	8.248	53.12	0
56.88	19.376	95.376	9.376	94.064	83.44	6.376	58.44	1
47	16.8	97.2	5.6	95.8	68	7.8	52	0
50	18	99	8	97	80	9	55	0
45.28	16.112	96.168	4.224	95.112	61.12	7.112	50.28	0
...	...	...	...	...	...	...	...	...

### 4.2 Hasil Klasifikasi Menggunakan Algoritma *K-Nearest Neighbor (KNN)*

Untuk mengetahui serta mengukur akurasi dari klasifikasi yang menggunakan metode *KNN* peneliti mencoba meng-input nilai dengan  $k$  sebesar  $k= 3, 5$  dan  $7$ . Peneliti memasukkan jumlah  $k$  yang sering digunakan sebagai penelitian dengan algoritma klasifikasi *KNN* ini untuk mencari jumlah  $k$  yang optimal. Namun, dikarenakan apabila peneliti memasukkan berapa saja jumlah  $k$ -nya hasil akurasi tetap menunjukkan nilai akurasi yang sempurna yaitu 100%, bisa dilihat sebagai berikut:

1. Pemodelan menggunakan algoritma KNN, mencoba memasukkan jumlah  $k = 3$ , dimana artinya akan diambil 3 tetangga terdekat untuk dapat menentukan kelas mana yang paling sesuai terhadap data yang di prediksi.

**Tabel 3.** *Confusion Matrix* Model Algoritma KNN,  $k=3$

		Kelas Prediksi				
		0	1	2	3	4
Kelas Sebenarnya	0	28	0	0	0	0
	1	0	19	0	0	0
	2	0	0	22	0	0
	3	0	0	0	28	0
	4	0	0	0	0	29

Dari uraian Tabel 3 *Confusion Matrix* dapat dihitung a akurasi, recall (sensitivity), presisi, *f1-score* dengan memanfaatkan persamaan (4), (5), (6), dan (7) pada bab 2 sebagai berikut :

- *Accuracy* 
$$= \frac{TP}{(TP + TN + FP + FN)} \times 100\%$$

$$= \frac{28+19+22+28+29}{28+19+22+28+29} \times 100\%$$

$$= 100\%$$

- *Precision* 
$$= \frac{TP}{(FP + TP)} \times 100\%$$

$$= \frac{28}{0+28} \times 100\%$$

$$= 100\%$$

- *Recall* 
$$= \frac{TP}{FN + TP} \times 100\%$$

$$= \frac{28}{0+28} \times 100\%$$

$$= 100\%$$

- *F1-Score* 
$$= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$= 2 \times \frac{100 \times 100}{100 + 100}$$

$$= 100\%$$

2. Pemodelan menggunakan algoritma KNN, mencoba memasukkan jumlah  $k = 5$ , dimana artinya akan diambil 5 tetangga terdekat untuk dapat menentukan kelas mana yang paling sesuai terhadap data yang di prediksi.

**Tabel 4.** *Confusion Matrix* Model Algoritma KNN,  $k=5$

		Kelas Prediksi				
		0	1	2	3	4
Kelas Sebenarnya	0	28	0	0	0	0
	1	0	19	0	0	0
	2	0	0	22	0	0
	3	0	0	0	28	0
	4	0	0	0	0	29

Dari uraian Tabel 4 Confusion Matrix dapat dihitung akurasi, recall (sensitivity), presisi, *f1-score* dengan memanfaatkan persamaan (4), (5), (6), dan (7) pada bab 2 sebagai berikut :

$$\begin{aligned}
 \bullet \text{ Accuracy} &= \frac{TP}{(TP + TN + FP + FN)} \times 100\% \\
 &= \frac{28+19+22+28+29}{28+19+22+28+29} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 \bullet \text{ Precision} &= \frac{TP}{(FP + TP)} \times 100\% \\
 &= \frac{28}{0+28} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 \bullet \text{ Recall} &= \frac{TP}{FN + TP} \times 100\% \\
 &= \frac{28}{0+28} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 \bullet \text{ F1-Score} &= 2x \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\
 &= 2x \frac{100 \times 100}{100 + 100} \\
 &= 100\%
 \end{aligned}$$

- Pemodelan menggunakan algoritma KNN, mencoba memasukkan jumlah  $k = 7$ , dimana artinya akan diambil 7 tetangga terdekat untuk dapat menentukan kelas mana yang paling sesuai terhadap data yang di prediksi.

**Tabel 5.** *Confusion Matrix* Model Algoritma KNN,  $k=7$

		Kelas Prediksi				
		0	1	2	3	4
Kelas	0	28	0	0	0	0

Sebenarnya	1	0	19	0	0	0
	2	0	0	22	0	0
	3	0	0	0	28	0
	4	0	0	0	0	29

Dari uraian Tabel 5 Confusion Matrix dapat dihitung akurasi, recall (sensitivity), presisi, *f1-score* dengan memanfaatkan persamaan (4), (5), (6), dan (7) pada bab 2 sebagai berikut :

- $$\begin{aligned} \bullet \text{ Accuracy} &= \frac{TP}{(TP + TN + FP + FN)} \times 100\% \\ &= \frac{28+19+22+28+29}{28+19+22+28+29} \times 100\% \\ &= 100\% \end{aligned}$$
- $$\begin{aligned} \bullet \text{ Precision} &= \frac{TP}{(FP + TP)} \times 100\% \\ &= \frac{28}{0+28} \times 100\% \\ &= 100\% \end{aligned}$$
- $$\begin{aligned} \bullet \text{ Recall} &= \frac{TP}{FN + TP} \times 100\% \\ &= \frac{28}{0+28} \times 100\% \\ &= 100\% \end{aligned}$$
- $$\begin{aligned} \bullet \text{ F1-Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{100 \times 100}{100 + 100} \\ &= 100\% \end{aligned}$$

### 4.3 Hasil Klasifikasi Menggunakan Algoritma *Decision Tree*

Untuk mengetahui serta mengukur akurasi dari klasifikasi yang menggunakan metode *Decision Tree* peneliti memasukkan beberapa parameter diantaranya menggunakan *criteria='entropy'* Kriteria ini digunakan untuk mengukur kualitas pemisahan dan dihitung dengan mengambil informasi yang diberikan oleh entropi untuk menentukan akar (*root node*) dan seleksi atribut. Karena dengan menggunakan entropi nilai yang dihasilkan lebih tinggi daripada menggunakan perhitungan gini index. Kemudian, mendeklarasikan kedalaman maksimum pohon '*max\_depth=3*'. Strategi untuk memilih split terbaik memasukkan *splitter='best'* dan terakhir mendeklarasikan '*random\_state=0*'.

**Tabel 6.** *Confusion Matrix Model Algoritma Decision Tree*

		Kelas Prediksi				
		0	1	2	3	4
Kelas	0	28	0	0	0	0

Sebenarnya	1	0	19	0	0	0
	2	0	1	21	0	0
	3	0	0	0	28	0
	4	0	0	0	0	29

Dari uraian Tabel 6 Confusion Matrix dapat dihitung akurasi, recall (sensitivity), presisi, *f1-score* dengan memanfaatkan persamaan (4), (5), (6), dan (7) pada bab 2 sebagai berikut :

- Accuracy* =  $\frac{TP}{(TP + TN + FP + FN)} \times 100\%$

=  $\frac{28+19+21+28+29}{28+19+1+21+28+29} \times 100\%$

= 99%
- Precision* =  $\frac{TP}{(FP + TP)} \times 100\%$

=  $\frac{28}{0+28} \times 100\%$

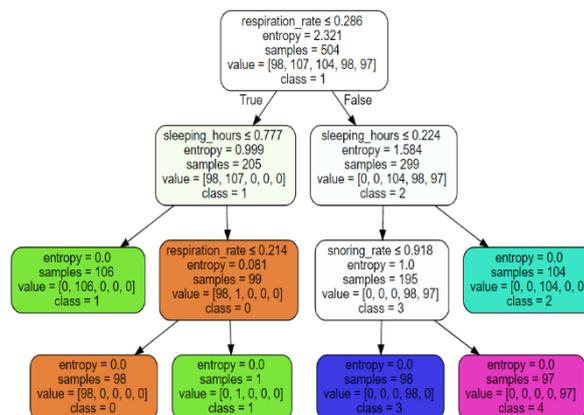
= 100%
- Recall* =  $\frac{TP}{FN + TP} \times 100\%$

=  $\frac{28}{0+28} \times 100\%$

= 100%
- F1-Score* =  $2x \frac{Precision * Recall}{Precision + Recall}$

=  $2x \frac{100 * 100}{100 + 100}$

= 100%



Gambar. 2. Hasil Graph Decision Tree

#### 4.4 Hasil Klasifikasi Menggunakan Algoritma Support Vector Machine (SVM)

Untuk mengetahui serta mengukur akurasi dari SVM untuk klasifikasi multi-kelas peneliti menggunakan metode *One Versus One (OVO) decision\_function\_shape = 'ovo'*. Pada SVM terdapat metode ini digunakan untuk mengklasifikasikan data yang memiliki lebih dari dua kelas.

**Tabel 7.** *Confusion Matrix Model Algoritma SVM*

		Kelas Prediksi				
		0	1	2	3	4
Kelas Sebenarnya	0	28	0	0	0	0
	1	0	19	0	0	0
	2	0	0	22	0	0
	3	0	0	0	28	0
	4	0	0	0	0	29

Dari uraian Tabel 7 Confusion Matrix dapat dihitung akurasi, recall (sensitivity), presisi, *f1-score* dengan memanfaatkan persamaan (4), (5), (6), dan (7) pada bab 2 sebagai berikut :

$$\begin{aligned}
 \bullet \text{ Accuracy} &= \frac{TP}{(TP + TN + FP + FN)} \times 100\% \\
 &= \frac{28+19+22+28+29}{28+19+22+28+29} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 \bullet \text{ Precision} &= \frac{TP}{(FP + TP)} \times 100\% \\
 &= \frac{28}{0+28} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 \bullet \text{ Recall} &= \frac{TP}{FN + TP} \times 100\% \\
 &= \frac{28}{0+28} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 \bullet \text{ F1-Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\
 &= 2 \times \frac{100 \times 100}{100 + 100} \\
 &= 100\%
 \end{aligned}$$

#### 4.5 Pembahasan

Dapat dilihat pada Tabel 8 setiap model memiliki hasil evaluasi masing-masing yang dilakukan terhadap performa ketiga algoritma yang digunakan didapatkan hasil *Accuracy, Precision, Recall, F1-score* dengan algoritma KNN dan SVM sebesar 100%, sedangkan *Accuracy, Precision, Recall, F1-score* menggunakan algoritma *Decision Tree* akurasinya sebesar 99%.

**Tabel 8.** Perbandingan Algoritma yang Berbeda

Algoritma	Akurasi (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	100	100	100	100
Decision Tree	99	100	100	100
SVM	100	100	100	100

## 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

Setelah proses percobaan untuk mengimplementasikan machine learning dalam mendeteksi tingkat stress manusia berdasarkan pola tidur dengan menggunakan dataset yang dimiliki dalam tugas akhir ini, mendapatkan beberapa hasil penelitian sebagai kesimpulan yakni :

1. Proses pengklasifikasian pengolahan data pada proses penelitian kami meliputi pengumpulan data, pengecekan data kemudian dilanjutkan dengan melakukan *split* atau membagi data menjadi data latih (*data train*) dan data uji (*data test*) dengan rincian *data train* dengan 80% total data, dan *data test* dengan 20% total data. Setelah proses pengujian selesai dilakukan perhitungan akurasi dari tiap-tiap algoritma yang digunakan yaitu *K-Nearest Neighbour*, *Decision Tree*, dan SVM.
2. Dalam mengimplementasikan setiap algoritma, menerapkan metode-metode yang sesuai untuk penelitian ini dari masing-masing algoritma. Pada penerapan KNN peneliti menggunakan pemodelan input nilai k sebesar k= 3, 5 dan 7. Untuk penerapan Decision Tree peneliti menggunakan criteria='entropy', dan untuk penerapan SVM peneliti menggunakan metode *multi-class SVM* yaitu dengan menerapkan One Versus One (OVO).
3. Atribut variabel yang digunakan dalam dataset SaYoPillow.csv yaitu *snoring rate*, *respiration rate*, *body temperature*, *limb movement rate*, *blood oxygen levels*, *eye movement*, *number of hours of sleep*, *heart rate* dan *stress levels*
4. Dalam penelitian ini tingkat stress diklasifikasikan menjadi 5 kelas yaitu kelas 0 dengan tingkat stress *low/normal*, kelas 1 dengan tingkat stress *medium low*, kelas 2 dengan tingkat stress *medium*, kelas 3 dengan tingkat stress *medium high*, kelas 4 dengan tingkat stress *high*.
5. Dari hasil evaluasi yang dilakukan terhadap performa ketiga algoritma yang digunakan didapatkan hasil *Accuracy*, *Precision*, *Recall*, *F1-score* dengan algoritma KNN dan SVM sebesar 1.0, sedangkan menggunakan algoritma Decision Tree *Accuracy* sebesar 0.99 dan *Precision*, *Recall*, *F1-score* sebesar 1.0

### 5.2 Saran

Setelah proses penelitian hingga evaluasi pengimplementasian machine learning yang kami lakukan, didapatkan hasil dari penerapan ketiga algoritma machine learning yaitu *K-Nearest Neighbor*, *Decision Tree*, dan *Support Vector Machine* menunjukkan hasil yang baik untuk melakukan penelitian untuk menganalisis tingkat *stress level* manusia berdasarkan kualitas tidur. Namun, diharapkan bahwa penelitian ini juga dapat dikembangkan menggunakan algoritma lainnya baik dalam lingkup *machine learning* maupun yang lebih luas. Disisi lain, untuk meningkatkan akurasi penelitian di masa mendatang dapat dilakukan dengan dataset yang memiliki jumlah lebih banyak data sehingga memiliki hasil akurasi yang lebih akurat.

## Referensi

- [1] E. Murniasih, "Apa yang Dimaksud Eustres, Distres dan Neustres?," 17 April 2020.
- [2] W. d. Nurul, 2007.
- [3] UPN "VETERAN" JAKARTA, p. 1.
- [4] F. P. F. A. B. Diva Fardiana Risa, "IMPLEMENTASI METODE NAÏVE BAYES UNTUK MENDETEKSI STRES SISWA," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 8, pp. 1301-1308, 2021.
- [5] A. L. P. R. A. N. Miftah Meinaldi Nurrochman, "IMPLEMENTASI MACHINE LEARNING UNTUK MENDETEKSI UNSUR DEPRESI," e-Proceeding of Engineering, vol. 8, pp. 6250-6256, 2021.
- [6] V. D. V. M. R. R. A. Menaka, "RECOGNITION OF HUMAN MENTAL STRES USING," JOURNAL OF ENGINEERING, COMPUTING & ARCHITECTURE, vol. 11, no. 4, pp. 82-86, 2021.
- [7] 2. Potter & Perry.
- [8] Bustami, "PENERAPAN ALGORITMA NAIVE BAYESUNTUK MENGLASIFIKASI DATA NASABAH ASURANSI," JURNAL INFORMATIKA, Vol. 8, No. 1, pp. 884-898, 2014
- [9] M.J. Sofi Defiyanti, "Integrasi Metode Klasifikasi dan Clustering dalam Data Mining," Konferensi Nasional Informatika (KNIF), pp. 39-44.
- [10] D. R. Wahyu Eko Susanto, "Komparasi Algoritma Neural Network, K-Nearest Neighbor Dan Naive Baiyes," Journal Speed – Sentra Penelitian Engineering dan Edukasi, vol. 8, p. 19, 2016.
- [11] B. U. S. o. I. System, "Support Vector Machine Algorithm," 14 Feb February 2022
- [12] Delimayanti K., Sari R., Laya M., M, R. Faisal, Pahul, "Pemanfaatan Metode Multiclass-SVM pada Model Klasifikasi Pesan Bendana Banjir di Twitter Mera Kartika," Edu Komputika Journal, 8, 2021.
- [13] M.L. Pratama, Murfi, H., "STUDI KOMPARASI METODE MULTICLASS SUPPORT VECTOR MACHINE UNTUK MASALAH ANALISIS SENTIMEN PADA TWITTER," FMIPA UI, 2014.
- [14] D. A. Ningtyas, M. Wahyudi, and N. Nurajijah, "Klasifikasi Siswa SMK Berpotensi Putus Sekolah Menggunakan Algoritma Decision Tree, Support Vector Machine Dan Naive Bayes," J. Khatulistiwa Inform., vol. 7, no. 2, pp. 85–90, 2019, doi: 10.31294/jki.v7i2.6839.
- [15] E. P. Nikkodemus, "Data mining : Konsep dan aplikasi menggunakan MATLAB," vol. xxiv, pp. 333-335, 2012.
- [16] Arthana. R. "Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning".2019. Diakses pada <https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-pecificity-serta-yang-diprioritaskan-b79ff4d77de8>