

# Klasifikasi Kesehatan Mental Mahasiswa Model TMAS dengan Algoritma Decision Tree, Logistic Regression, dan Random Forest

Anindya Ananda Hapsari<sup>1\*</sup>, Ahmad Syafei Nursuwanda<sup>2</sup>, Halimatuz Zuhriyah<sup>3</sup>,  
Devan Junesco Vresdian<sup>4</sup>

<sup>1-3</sup>Teknik Informatika , Universitas Global Jakarta, Depok 16412, Indonesia

<sup>4</sup>Teknik Elektro , Universitas Global Jakarta, Depok 16412, Indonesia

aninananda@jgu.ac.id, ahmadsyafei@jgu.ac.id, halimatuz@jgu.ac.id, devan@jgu.ac.id

## Abstrak

Di Indonesia, 6,1% penduduk berusia 15 tahun ke atas mengalami gangguan kesehatan mental, dengan 12,69% mahasiswa menunjukkan kerentanan akibat tekanan hidup dan ketidakstabilan mental. Kesehatan mental sangat penting bagi mahasiswa, karena berpengaruh langsung pada kesuksesan akademik. Penelitian ini menganalisis data kesehatan mental mahasiswa dengan dataset yang diambil berdasarkan hasil survey data mahasiswa di Universitas Global Jakarta, dengan hasil survey mengikuti metode TMAS yang direkomendasikan oleh *expert* psikolog untuk kemudian diolah menggunakan algoritma *Decision Tree*, *Random Forest*, dan *Logistic Regression* dengan Python. Hasilnya menunjukkan bahwa *Logistic Regression* mencapai akurasi tertinggi sebesar 90%, diikuti oleh *Decision Tree* dan *Random Forest* yang masing-masing memiliki akurasi 80%. Selain itu, model *Logistic Regression* memberikan prediksi kecemasan paling akurat berdasarkan TMAS dengan akurasi 90%, dibandingkan 80% untuk *Decision Tree* dan *Random Forest*.

**Kata kunci:** Kesehatan Mental, Python, Klasifikasi, Algoritma, Data Science

## Abstract

*In Indonesia, 6.1% of the population aged 15 and older experience mental health disorders, with 12.69% of students showing vulnerability due to life stressors and mental instability. Mental health is crucial for students, as it directly impacts their academic success. This study analyzes student mental health data using a dataset derived from surveys conducted at Global University Jakarta, with the survey following TMAS methods recommended by psychological experts. The data is processed using Decision Tree, Random Forest, and Logistic Regression algorithms in Python. The results indicate that Logistic Regression achieves the highest accuracy at 90%, followed by Decision Tree and Random Forest, both with an accuracy of 80%. Furthermore, the Logistic Regression model provides the most accurate anxiety predictions based on TMAS, with an accuracy of 90%, compared to 80% for Decision Tree and Random Forest.*

**Keywords:** Mental Health, Python, Classification, Algorithm, Data Science

## 1. PENDAHULUAN

Di Indonesia, gangguan kesehatan mental mulai menyerang individu berusia 15 tahun ke atas, dengan prevalensi mencapai 6,1%. Di antara mahasiswa, risiko ini lebih tinggi, terutama bagi mereka yang berada di fase remaja akhir dan dewasa awal. Dalam kondisi mental yang tidak stabil dan di tengah konflik hidup, sekitar 12,69% mahasiswa dilaporkan mengalami masalah kesehatan mental (Salvia, 2021). Kesehatan mental sangat penting bagi mahasiswa, karena berdampak langsung pada

kesuksesan akademik mereka. Masalah mental dapat memengaruhi kemampuan komunikasi, sehingga mengganggu aktivitas sehari-hari (Dwi Putra et al., 2023) Sejumlah penelitian telah menerapkan metode *data science* untuk menganalisis masalah kesehatan mental. Metode ini memungkinkan deteksi dini, yang penting untuk mencegah kondisi memburuk (Thenata & Suryadi, 2022). Penelitian sebelumnya menguji potensi algoritma pembelajaran mesin untuk mendiagnosis dan memprediksi gangguan mental, serta aplikasinya dalam pendidikan dan kesehatan (Muzumdar et al., 2022). Salah satu

penelitian oleh (Dwi Putra et al., 2023) membandingkan algoritma *Naive Bayes Classifier* dan *Support Vector Machine*, menunjukkan bahwa *Support Vector Machine* memiliki akurasi 94,37%. Penelitian lain oleh (Wibowo et al., 2023) menemukan bahwa algoritma *Decision Tree* dan *Random Forest* mencapai akurasi 100% dalam mendeteksi stres mahasiswa.

Berdasarkan penelitian ini, penggunaan algoritma klasifikasi seperti *Decision Tree*, *Random Forest*, dan *Logistic Regression* dapat efektif untuk menganalisis data kesehatan mental mahasiswa. Penelitian ini akan menerapkan algoritma tersebut pada dataset kesehatan mental mahasiswa menggunakan Python, yang dikenal sebagai salah satu bahasa pemrograman teratas dalam *data science* (Syamsu, 2021). Fokus utama akan berada pada nilai akurasi sebagai tolok ukur performa, bersama metrik tambahan seperti *precision*, *recall*, dan *F1 score*, yang akan divalidasi melalui *cross-validation*.

Hasil dari model algoritma ini akan dibandingkan dengan metode psikologi untuk menilai ketepatan prediksi. Penelitian ini bertujuan untuk menentukan algoritma terbaik dalam klasifikasi kesehatan mental mahasiswa antara *Decision Tree*, *Logistic Regression*, dan *Random Forest*, serta membandingkan hasil prediksi ketiga algoritma dalam mengukur tingkat kecemasan menggunakan TMAS, dengan evaluasi melalui metrik *accuracy*, *precision*, *recall*, dan *F1-score* serta teknik validasi *cross-validation*.

## 2. METODE PENELITIAN

Berdasarkan penelitian sebelumnya yang telah dilakukan review dan menjadi latar belakang penelitian ini, terdapat kesamaan dan perbedaan dalam metode, objek, dan hasil penelitian. Beberapa studi menggunakan data dari sumber yang berbeda tetapi menerapkan metode dan algoritma yang bervariasi, sementara yang lain menggunakan metode dan algoritma yang sama dengan data yang berbeda.

Penelitian ini merupakan gabungan dan pengembangan dari penelitian sebelumnya, dengan fokus pada dua objek data yang akan dianalisis. Peneliti akan membandingkan hasil pengolahan data menggunakan algoritma *Decision Tree*, *Logistic Regression*, dan *Random Forest* untuk menentukan algoritma mana yang menghasilkan performa tertinggi.

### 2.1 Data Science

*Data Science* adalah disiplin ilmu yang memfokuskan pada analisis data, baik yang terstruktur maupun tidak terstruktur, terutama data numerik. Bidang ini mencakup pengumpulan, pengolahan, pengelolaan, klasifikasi, dan pengarsipan data. *Data Science* bertujuan untuk menyajikan dan mendistribusikan data, serta mengubahnya menjadi informasi yang mudah dipahami oleh semua orang (Aditya, M. A., et al., 2020).

### 2.2 Metode Klasifikasi

Salah satu metode dalam *machine learning* yang termasuk dalam kategori *supervised learning* adalah klasifikasi. Teknik ini bertujuan untuk memprediksi kategori dari data input, dengan fokus pada mencapai tingkat akurasi tertinggi dalam menentukan kelas target. Algoritma klasifikasi menganalisis hubungan antara atribut input dan output untuk membangun model yang akan digunakan dalam pengujian. Sebelum melakukan prediksi, diperlukan tahap pembelajaran yang menggunakan data latih (*training data*), sedangkan data uji (*testing data*) digunakan untuk proses prediksi itu sendiri (Charbuty & Abdulazeez, 2021).

### 2.3 Algoritma Decision Tree

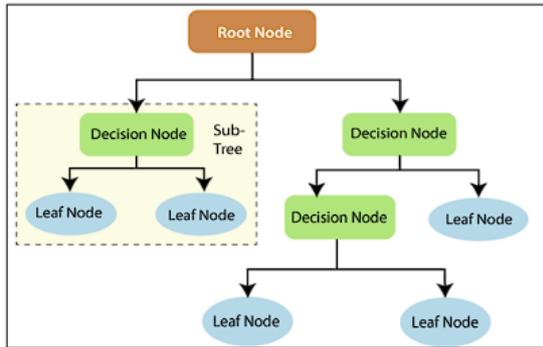
*Decision Tree* adalah algoritma *machine learning* yang digunakan untuk klasifikasi dengan menggambarkan proses pengambilan keputusan dalam bentuk pohon. Pada struktur ini, setiap node mewakili keputusan, cabang menunjukkan pengukuran fitur berdasarkan parameter input, dan node daun mencerminkan hasil klasifikasi atau nilai akhir (Pratama et al., 2023).

Algoritma ini berfungsi sebagai sistem pendukung keputusan, di mana node internal melakukan pengujian terhadap atribut, cabang mencerminkan hasil dari pengujian tersebut, dan node daun mengidentifikasi kelas yang ditentukan. Bentuk visual dari algoritma *Decision Tree* dapat dilihat pada gambar 1.

### 2.4 Algoritma Logistic Regression

*Logistic Regression* adalah suatu algoritma yang digunakan untuk memprediksi probabilitas keberhasilan atau kegagalan suatu peristiwa dengan menghasilkan suatu hasil sebagai probabilitas antara 0 dan 1. Algoritma ini menggunakan variabel independen (atribut) dan

dependen (tujuan) dengan menggunakan fungsi logika (Pratama et al., 2023).



Gambar 1. Algoritma *Decision Tree* (Charbuty & Abdulazeez, 2021)

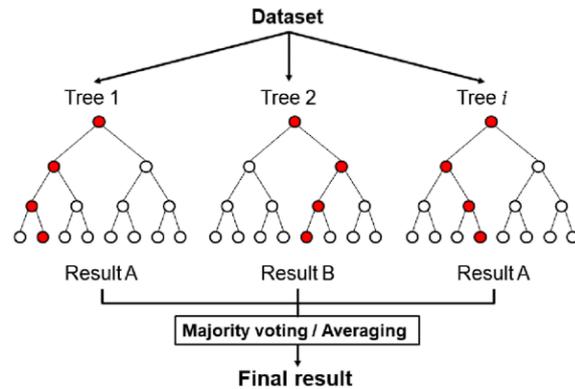
### 2.5 Algoritma Random Forest

Adalah metode yang digunakan untuk klasifikasi, regresi, dan operasi lainnya berdasarkan teknik *clustering* dalam pembelajaran data. Proses seleksi dibagi menjadi dua bagian, dan subset atribut diambil secara acak dari atribut terbaik. Dengan demikian, *Random Forest* menghasilkan informasi acak dan mengidentifikasi atribut kunci untuk membangun pohon keputusan. Dalam klasifikasi untuk data yang tidak teramati, keputusan diambil berdasarkan suara terbanyak dari setiap pohon, yang memungkinkan metode ini memberikan kinerja yang baik, terutama dalam mengatasi berbagai masalah kinerja seperti *overfitting*. Selain itu, *Random Forest* juga mampu menangani *noise* dan *outlier* dengan efektif dan relatif mudah diterapkan (Wibowo et al., 2023). Bentuk visual dari *Random Forest* dapat dilihat pada gambar 2.

### 2.6 Anxiety

*Anxiety* adalah gangguan yang ditandai oleh ketakutan atau kecemasan yang tidak terkendali, sering kali dipicu oleh tekanan dan stres yang tinggi, terutama di kalangan mahasiswa. Gejala kecemasan mencakup kesulitan mengendalikan emosi, khawatir berlebihan tentang hal-hal kecil, dan kecemasan yang mengganggu aktivitas serta istirahat. Hal ini dapat memengaruhi kemampuan siswa untuk mencapai tujuan akademik (Lintang et al., 2023). Penderita anxiety sering menghadapi kesulitan dalam menjalani aktivitas sehari-hari dan dapat mengalami efek fisik, seperti sakit dada dan hipertensi. Tingkat kecemasan yang tinggi dapat merusak prestasi akademik, sehingga deteksi dini sangat penting untuk mendapatkan

pengobatan yang tepat dan mencegah kondisi semakin memburuk (Thenata & Suryadi, 2022).



Gambar 2. Algoritma *Random Forest* (Ghosh, 2024)

### 2.7 Taylor Minnesota Anxiety Scale

*Taylor Minnesota Anxiety Scale* (TMAS) adalah alat yang digunakan untuk mengukur tingkat kecemasan individu. TMAS mengevaluasi kesehatan mental dengan menyajikan pernyataan yang mencakup tiga aspek: fisiologis, kognitif, dan afektif, yang dijawab dengan dua pilihan, yaitu "ya" atau "tidak." Hingga saat ini, TMAS masih banyak digunakan dan diadaptasi dalam berbagai penelitian yang berkaitan dengan kecemasan (Kumar Barman & Kumar, 2022)

### 2.8 Simple Random Sampling

Salah satu metode pengambilan sampel yang umum digunakan dalam penelitian ilmiah adalah simple random sampling. Dalam metode ini, anggota sampel dipilih secara acak, sehingga setiap individu memiliki peluang yang sama untuk terpilih. Simple random sampling diakui sebagai "metode paling sederhana dan paling umum untuk pemilihan sampel," di mana unit sampel dipilih satu per satu dengan probabilitas pemilihan yang sama pada setiap pengundian. Salah satu keunggulan dari metode ini adalah kesederhanaannya, yang membuatnya mudah dipahami dan diterapkan. Selain itu, metode ini bersifat non-bias karena memberikan kesempatan yang sama bagi setiap individu untuk terpilih, sehingga mengurangi potensi bias dalam pemilihan sampel. Dengan ukuran sampel yang cukup besar, simple random sampling cenderung menghasilkan sampel yang representatif, meskipun sampel yang lebih kecil mungkin menghadapi risiko variasi yang lebih besar (Golzar & Noor, 2022).

### 2.9 Slovin

Rumus Slovin adalah metode populer dalam penelitian kuantitatif untuk menentukan jumlah sampel yang representatif, memungkinkan hasil penelitian digeneralisasikan tanpa memerlukan tabel jumlah sampel. Dengan rumus yang sederhana, Slovin memungkinkan perhitungan sampel antara 10% hingga 20% dari total populasi, dengan nilai e yang digunakan adalah 0,1 (10%) dan 0,2 (20%) (Santoso, 2023).

$$n = \frac{N}{1 + N(e)^2}$$

Keterangan:

n = Ukuran sampel/jumlah responden

N = Ukuran populasi

E = Persentase kelonggaran ketelitian kesalahan pengambilan sampel yang masih bisa ditolerir;  
e= 0,1

### 2.10 Pengembangan Model Algoritma

Pada tahap ini peneliti mulai mengembangkan model algoritma yang akan digunakan sebagai inti dari penelitian ini. Hal-hal atau tools yang digunakan dalam penelitian ini adalah *Google Collab* sebagai media tempat model dibangun dan menggunakan bahasa pemrograman Python. Gambar 3 adalah langkah-langkah yang dilakukan dalam pengembangan model.

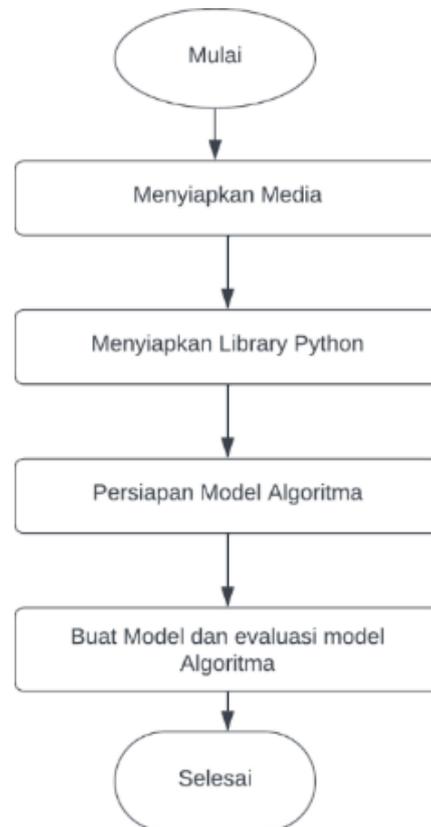
Pada penelitian ini menggunakan *Google Collab* sebagai media *training* data dan membentuk model. Selanjutnya menyiapkan *library* Python yang akan digunakan. Setelah media atau toolsnya sudah siap disini peneliti akan memasukkan *library* Python dalam model algoritma *library* yang digunakan dalam Python adalah Numpy, Pandas, Matplotlib, Seaborn, Plotly, dan Scipy.

Setelah semua *library* tersedia, langkah berikutnya adalah membuat tab untuk persiapan sebelum pengujian model. Ini mencakup "*reading dataset*" untuk membaca dan mengekspor data, "*preprocessing data*" untuk membersihkan dan menyiapkan data, serta "*training-test data*" untuk membagi data menjadi set data latih dan data uji.

### 2.11 Dataset

Untuk dataset yang digunakan pada penelitian ini diawali dengan proses pengumpulan data yang dilakukan dengan menyebarkan kuisioner melalui *Google Form* kepada mahasiswa

Universitas Global Jakarta yang bersedia mengisi kuisioner tersebut.



Gambar 3. Diagram alur pengembangan model

✓ Import Library

```

[ ] import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from scipy import stats
from scipy.stats import randint

# prep
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV, train_test_split
from sklearn import preprocessing
from sklearn.datasets import make_classification
from sklearn.preprocessing import binarize, LabelEncoder, MinMaxScaler

# models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus
from sklearn import tree
import six
import sys
import pickle
sys.modules['sklearn.externals.six'] = six

# Validation libraries
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import f1_score, precision_score, recall_score, confusion_matrix, accuracy_score
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.model_selection import cross_val_score
    
```

Gambar 4. Library Python yang digunakan

Metode pengambilan sampel yang digunakan adalah *Simple Random Sampling*. Untuk menentukan jumlah minimal sampel yang akan dijadikan responden, peneliti menerapkan rumus Slovin. Variabel pertanyaan dalam kuisioner diambil dari penelitian sebelumnya yang telah terbukti efektif dan reliabel dalam memprediksi kecema-

san kesehatan mental mahasiswa. Selain itu, beberapa pertanyaan juga dirumuskan berdasarkan saran dari pakar melalui wawancara langsung dan referensi dari penelitian terdahulu, pertanyaan juga mengikuti dari pakar untuk pengambilan dengan model pertanyaan TMAS.

### 2.12 Pre-Processing Data

Pada tahap ini, data yang diperoleh sebelumnya akan melalui proses *pre-processing* terlebih dahulu. *Library* Python yang digunakan adalah NumPy dan Pandas. NumPy adalah library yang umum digunakan untuk perhitungan ilmiah dan mendukung data multidimensi serta matriks besar.



Gambar 5. Library Python NumPy dan Pandas

Sementara itu, Pandas menyediakan fungsi untuk menghapus kolom atau baris, mengisi nilai yang hilang, dan melakukan berbagai operasi lainnya pada *DataFrame*. Dalam tahap ini, data akan dibersihkan, termasuk membersihkan nilai yang hilang, menghapus kolom atau fitur yang tidak diperlukan, hingga data siap untuk masuk ke tahap berikutnya. Pada tahap *pre-processing* dilakukan, *cleaning* data, *rename* dan *encoder* data, normalisasi data,

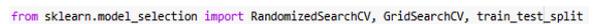
Setelah data dinormalisasi, langkah berikutnya adalah menambahkan target untuk model algoritma. Target ini ditentukan melalui perhitungan poin fitur yang mengukur tingkat kecemasan menggunakan metode TMAS. Hasil perhitungan tersebut akan menjadi target untuk model *Decision Tree*, *Random Forest*, dan *Logistic Regression*. Prosesnya meliputi pembuatan kolom baru berisi poin dari pertanyaan TMAS, yang dinamakan "*anxiety predict*," serta kolom "*kategori*" untuk mengklasifikasikan data sebagai "*anxiety*" atau "*normal*." Selanjutnya, data dalam kolom "*kategori*" diubah menjadi format numerik.



Gambar 6. Proses penambahan target

### 2.13 Pengujian Model Algoritma

Pada tahap ini, data yang telah bersih dari proses *pre-processing* akan dimasukkan ke dalam model algoritma *Decision Tree*, *Random Forest*, dan *Logistic Regression* untuk mengevaluasi akurasi. Sebelum itu, perlu dilakukan penentuan target dan fitur, serta membagi data menjadi dua bagian: data latihan (*training*) dan data uji (*testing*) menggunakan *library train\_test\_split* dari *sklearn*. Pada proses ini, peneliti akan mengubah label menjadi data X untuk fitur dan Y untuk target. Langkah-langkahnya meliputi memasukkan fitur ke dalam "*feature\_cols*" untuk menghasilkan nilai X, serta menambahkan target ke dalam data Y untuk dilatih dan diuji. Berikut gambar 7 dan 8 adalah kode untuk proses ini.



Gambar 7. Library *train\_test\_split* yang digunakan



Gambar 8. Proses split data sebelum masuk ke dalam model

Kedua dataset akan dilatih dan diuji menggunakan perbandingan yang dianggap sesuai oleh peneliti. Peneliti menggunakan dua ukuran: 70% untuk data *training* dan 30% untuk data *testing*, serta 80% untuk data *training* dan 20% untuk data *testing*, untuk mengevaluasi performa model pada ukuran *training* yang berbeda. Setelah pembagian data, keduanya akan dimasukkan ke dalam berbagai model algoritma untuk diolah, diprediksi, dan dievaluasi hasilnya.

## 3. HASIL DAN PEMBAHASAN

Hasil dari pengumpulan data di proses dengan menyebarkan kuesioner melalui metode *Simple Random Sampling*. Populasi yang diteliti adalah 1199 mahasiswa Universitas Global Jakarta. Un-

tuk menentukan jumlah minimal sampel responden, peneliti menggunakan rumus Slovin dengan margin of error 15%. Sehingga dalam dataset uji didapatkan 50 responden.

Gambar 9. Data yang dikumpulkan

Data yang didapatkan kemudian diolah dengan metode *Pre-processing*, yaitu *cleaning* data, *Re-name* dan *Encoder* data, normalisasi data, dan klasifikasi untuk mendapatkan *anxiety predict*. Setelah model dibentuk dilakukan pengujian model algoritma. Setelah data dinormalisasi, langkah selanjutnya adalah menambahkan target sebagai nilai Y untuk pengujian dan pelatihan model algoritma. Peneliti menggunakan perhitungan TMAS dari kuesioner untuk menentukan target yang dimasukkan ke dalam model. Gambar 10 – 12 adalah tabel data sebelum dan sesudah penambahan kolom target.

...	Khawatir Tanpa Alasan	Sedikit Ketakutan	Rasanya Tidak Berguna di saat tertentu	Sulit Fokus	Rasanya Canggung	Mengambil Sesuatu dengan Berat	Rasanya Tidak Berguna	Kurang Kepercayaan Diri	Hampir Hancur	Percaya Diri
...	0	1	0	0	0	1	0	0	0	0
...	1	0	0	1	0	0	1	1	1	1
...	1	0	0	1	1	1	1	1	1	0
...	1	1	0	1	1	1	1	1	1	1
...	0	0	0	0	0	0	0	0	0	1

Gambar 10. Data sebelum ditambahkan target

...	Rasanya Tidak Berguna di saat tertentu	Sulit Fokus	Rasanya Canggung	Mengambil Sesuatu dengan Berat	Rasanya Tidak Berguna	Kurang Kepercayaan Diri	Hampir Hancur	Percaya Diri	anxiety predict	kategori
...	0	0	1	1	0	0	0	0	13	normal
...	1	0	0	0	1	1	1	1	25	anxiety
...	1	1	1	1	1	1	0	1	25	anxiety
...	0	1	1	1	1	1	1	1	30	anxiety
...	0	0	0	0	0	0	1	1	17	normal

Gambar 11. Data setelah ditambahkan target

...	Rasanya Tidak Berguna di saat tertentu	Sulit Fokus	Rasanya Canggung	Mengambil Sesuatu dengan Berat	Rasanya Tidak Berguna	Kurang Kepercayaan Diri	Hampir Hancur	Percaya Diri	anxiety predict	kategori
...	0	0	1	1	0	0	0	0	13	0
...	1	0	0	0	1	1	1	1	25	1
...	1	1	1	1	1	1	0	1	25	1
...	0	1	1	1	1	1	1	1	30	1
...	0	0	0	0	0	0	1	1	17	0

Gambar 12. Data setelah target diencoder

Seperti terlihat pada gambar 10 yang sebelumnya hanya memiliki fitur kini telah ditambahkan kolom "*anxiety predict*" beserta kategorinya, seperti pada gambar 11. Setelah itu, target di *encoding* agar menjadi format numerik, seperti yang ditunjukkan pada gambar 12. Kolom ini diperoleh melalui perhitungan poin fitur dengan metode TMAS. Kolom "kategori" akan digunakan sebagai target dalam model algoritma

untuk memprediksi apakah seseorang terklasifikasi sebagai "*anxiety*" atau "*normal*".

### 3.1. Pengujian Model Algoritma

Pada tahap pengujian algoritma, peneliti mendapatkan hasil yang bervariasi dari setiap percobaan. Sebelum pengujian model, data dibagi menjadi data latih dan data uji dengan dua proporsi: 70-30 (70% untuk pelatihan dan 30% untuk pengujian) dan 80-20 (80% untuk pelatihan dan 20% untuk pengujian).

### 3.2 Decision Tree

Berikut gambar 3 adalah hasil dari pengujian model algoritma *Decision Tree* setelah dimasukkan data uji dan data latih kedalamnya untuk setiap *Train Test Size* yang digunakan.

#### 1. Train Test Size 70-30

```

Decision Tree Accuracy: 0.7333333333333333
Decision Tree Confusion Matrix:
[[6 1]
 [3 5]]
Decision Tree Classification Report:
              precision    recall  f1-score   support

   0           0.67         0.86         0.75         7
   1           0.83         0.62         0.71         8

 accuracy          0.73         0.73         0.73         15
 macro avg         0.75         0.74         0.73         15
 weighted avg      0.76         0.73         0.73         15
    
```

Gambar 13. Hasil uji model *Decision Tree* train test size 70-30

Hasil dari pengujian model algoritma *Decision Tree* dengan *Train Test Size* 70-30 mendapatkan tingkat akurasi 0.73 atau sekitar 73%. Berikut tabel 1 hasil *Classification report* yang didapatkan.

Tabel 1. *Classification report* DT Train Test Size 70-30

	Precision	Recall	F1-score
	83%	62%	71%
Accuracy			73%

#### 2. Train Test Size 80-20

```

Decision Tree Accuracy: 0.5
Decision Tree Confusion Matrix:
[[3 0]
 [5 2]]
Decision Tree Classification Report:
              precision    recall  f1-score   support

   0           0.38         1.00         0.55         3
   1           1.00         0.29         0.44         7

 accuracy          0.50         0.64         0.49         10
 macro avg         0.69         0.64         0.49         10
 weighted avg      0.81         0.50         0.47         10
    
```

Gambar 14. Hasil uji model *Decision Tree* rasio train test size 80-20

Hasil dari pengujian model algoritma *Decision Tree* dengan *Train Test Size* 80-20 mendapatkan tingkat akurasi yang lebih tinggi dari sebelumnya yakni 0.5 atau sekitar 50%. Berikut tabel 2 hasil *Classification report* yang didapatkan.

Tabel 2. *Classification report* DT Train Test Size 80-20

<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
100%	29%	44%
Accuracy		50%

### 3.3 Random Forest

Berikut gambar 15 adalah hasil dari pengujian model algoritma *Random Forest* setelah dimasukkan data uji dan data latih kedalamnya untuk setiap *Train Test Size* yang digunakan.

#### 1. Train Test Size 70-30

```

Random Forest Accuracy: 0.8
Random Forest Confusion Matrix:
[[6 1]
 [2 6]]
Random Forest Classification Report:
precision recall f1-score support
0 0.75 0.86 0.80 7
1 0.86 0.75 0.80 8
accuracy 0.80 0.80 0.80 15
macro avg 0.80 0.80 0.80 15
weighted avg 0.81 0.80 0.80 15
    
```

Gambar 15. Hasil uji model *random forest train test size* 70-30

Dari gambar 15 dapat terlihat bahwa hasil pengujian model algoritma *Random Forest* terhadap *Train Test Size* 70-30 mendapatkan hasil akurasi 0.80 atau sekitar 80%. Berikut tabel 3 hasil *Classification report* yang didapatkan.

Tabel 3. *Classification report* RF Train Test Size 70-30

<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
86%	75%	80%
Accuracy		80%

#### 2. Train Test Size 80-20

```

Random Forest Accuracy: 0.8
Random Forest Confusion Matrix:
[[3 0]
 [2 5]]
Random Forest Classification Report:
precision recall f1-score support
0 0.60 1.00 0.75 3
1 1.00 0.71 0.83 7
accuracy 0.80 0.86 0.80 10
macro avg 0.80 0.86 0.79 10
weighted avg 0.88 0.80 0.81 10
    
```

Gambar 16. Hasil uji model *random forest train test size* 80-20

Pada gambar 16 terlihat bahwa nilai akurasi untuk model algoritma *Random Forest* tidak berubah pada *Train Test Size* 80-20 yakni 0.80 atau sekitar 80%. Berikut tabel 4 hasil *Classification report* yang didapatkan.

Tabel 4. *Classification report* RF Train Test Size 80-20

<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
100%	71%	83%
Accuracy		80%

### 3.4 Logistic Regression

Berikut gambar 17 adalah hasil dari pengujian model algoritma *Logistic Regression* setelah dimasukkan data uji dan data latih kedalamnya untuk setiap *Train Test Size* yang digunakan.

#### 1. Train Test Size 70-30

```

Logistic Regression Accuracy: 0.8
Logistic Regression Confusion Matrix:
[[5 2]
 [1 7]]
Logistic Regression Classification Report:
precision recall f1-score support
0 0.83 0.71 0.77 7
1 0.78 0.88 0.82 8
accuracy 0.80 0.80 0.80 15
macro avg 0.81 0.79 0.80 15
weighted avg 0.80 0.80 0.80 15
    
```

Gambar 17. Hasil uji model *Logistic Regression train test size* 70-30

Hasil dari model algoritma *Logistic Regression* dapat dilihat dari gambar 17 dimana akurasi yang dihasilkan dengan *Train Test Size* 70-30 adalah 0.80 atau sekitar 80%. Berikut tabel 5 hasil *Classification report* yang didapatkan.

Tabel 5. *Classification report* LG Train Test Size 70-30

<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
78%	88%	82%
Accuracy		80%

#### 2. Train Test Size 80-20

Dari hasil model algoritma *Logistic Regression* pada *Train Test Size* 80-20 dapat dilihat dari gambar 18 dimana hasil akurasi yang didapat adalah 0.90 atau sekitar 90% lebih tinggi dari pada hasil model algoritma pada *Train Test Size* sebelumnya. Berikut tabel 6 hasil *Classification report* yang didapatkan.

```

Logistic Regression Accuracy: 0.9
Logistic Regression Confusion Matrix:
[[3 0]
 [1 6]]
Logistic Regression Classification Report:

```

	precision	recall	f1-score	support
0	0.75	1.00	0.86	3
1	1.00	0.86	0.92	7
accuracy			0.90	10
macro avg	0.88	0.93	0.89	10
weighted avg	0.93	0.90	0.90	10

Gambar 18 hasil uji model *logistic regression* train test size 80-20

Tabel 6. Classification report RF Train Test Size 80-20

<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
100%	86%	92%
Accuracy		90%

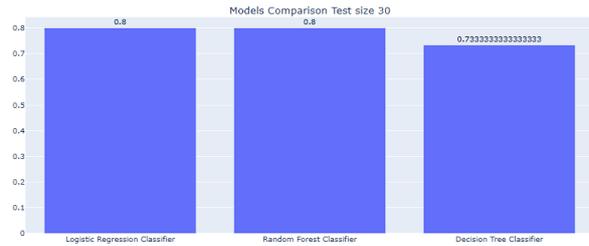
### 3.5 Analisa Hasil

Setelah dilakukan pengujian dan menganalisis hasil dari performa model algoritma *Decision Tree*, *Random Forest*, dan *Logistic Regression*. Didapatkan hasil yang dapat dijadikan acuan model mana yang performanya paling baik dalam mengolah dan memprediksi data kesehatan mental mahasiswa terkhusus pada tingkat *anxiety* mahasiswa. Berikut adalah tabel 7 perbandingan hasil akurasi dari model algoritma *Decision Tree*, *Random Forest*, dan *Logistic Regression* untuk tiap Train Test Size yang digunakan.

Dari tabel 7 bisa melihat untuk tiap akurasi dari model algoritma *Decision Tree*, *Random Forest*, dan *Logistic Regression*. Agar lebih mudah untuk melihat model algoritma mana yang performanya paling baik dalam setiap *Train Test Size*. Berikut gambar 19 bagan untuk melihat performa akurasi dari model algoritma *Decision Tree*, *Random Forest*, dan *Logistic Regression*.

Tabel 7. Perbandingan *accuracy* model algoritma

	<i>Decision tree</i>	<i>Random forest</i>	<i>Logistic regression</i>
Train Test Size 70-30	73%	80%	80%
Train Test Size 80-20	50%	80%	90%



Gambar 19. Bagan perbandingan akurasi model algoritma rasio *train test size* 70-30



Gambar 20. Bagan perbandingan akurasi model algoritma rasio *train test size* 80-20

Dari diagram gambar 19 – 20 dan tabel 7, terlihat bahwa model algoritma Regresi Logistik menunjukkan kinerja yang lebih baik dibandingkan dengan kedua model lainnya, yaitu *Decision Tree* dan *Random Forest*. Selain akurasi, peneliti juga memperhatikan faktor-faktor lain seperti *recall*, *precision*, dan *f1-score* dalam evaluasi model.

#### 1. Recall

Tabel 8. Perbandingan *recall* model algoritma

	<i>Decision Tree</i>	<i>Random Forest</i>	<i>Logistic Regression</i>
Train Test Size 70-30	62%	75%	88%
Train Test Size 80-20	29%	71%	92%

Dari tabel 8 dapat dilihat bahwa model algoritma *Logistic Regression* memiliki nilai *recall* yang paling tinggi dibandingkan model algoritma *Decision Tree* dan *Random Forest*.

#### 2. Precision

Tabel 9. Tabel perbandingan *precision* model algoritma

	<i>Decision Tree</i>	<i>Random Forest</i>	<i>Logistic Regression</i>
Train Test Size 70-30	83%	86%	78%
Train Test Size 80-20	100%	100%	100%

Pada tabel 9 diatas didapatkan nilai *precision* untuk model algoritma *Random Forest* lebih unggul dibandingkan model algoritma *Decision Tree* dan *Logistic Regression*.

### 3. F1-score

Tabel 10 tabel perbandingan *f1-score* model algoritma

	<i>Decision tree</i>	<i>Random forest</i>	<i>Logistic regression</i>
Train Test Size 70-30	71%	80%	82%
Train Test Size 80-20	44%	83%	92%

Dapat dilihat dari tabel 10 yang menunjukkan nilai *f1-score* diatas bahwa model algoritma *Logistic Regression* memiliki performa *f1-score* yang lebih baik dibandingkan dengan model algoritma *Decision Tree* dan *Random Forest*. Setelah menganalisis kinerja dari tiga model algoritma *Decision Tree*, *Random Forest*, dan *Regresi Logistic* peneliti dapat menyimpulkan model mana yang paling superior. Dalam pengambilan keputusan mengenai model algoritma yang memiliki performa terbaik, menyajikan tabel perbandingan yang menunjukkan hasil dari setiap model. Tabel ini mengilustrasikan seberapa sering masing-masing model memberikan hasil yang lebih baik secara keseluruhan dibandingkan dengan yang lain, sehingga memungkinkan peneliti untuk dengan jelas melihat keunggulan dari masing-masing algoritma dalam konteks kinerja yang diukur.

### 4. KESIMPULAN

Kesimpulan penelitian ini menunjukkan bahwa model algoritma *Regresi Logistik* memiliki performa terbaik, dengan akurasi mencapai 90%, *recall* 86%, *precision* 100%, dan *f1-score* 92%. Sebagai perbandingan, model *Decision Tree* hanya meraih akurasi 50% dengan *recall* 29%, *precision* 100%, dan *f1-score* 44%. Sementara itu, model *Random Forest* memperoleh akurasi 80%, *recall* 71%, *precision* 100%, dan *f1-score* 83%. Dalam prediksi kecemasan menggunakan metode TMAS, *Regresi Logistik* juga menampilkan hasil yang paling akurat, berhasil memprediksi 4 dari 5 data sebagai kecemasan, sedangkan *Decision Tree* dan *Random Forest* masing-masing hanya mampu memprediksi 3 data dengan akurasi 50% dan 80%. Hal ini mengindikasikan bahwa *Regresi Logistik* lebih efektif da-

lam mendeteksi kecemasan dibandingkan dengan kedua model lainnya.

### REKOMENDASI

Berdasarkan hasil penelitian yang menunjukkan keunggulan model regresi logistik dalam memprediksi kecemasan, disarankan agar penelitian selanjutnya mempertimbangkan penggunaan dan pengembangan lebih lanjut dari algoritma ini dalam konteks kesehatan mental. Peneliti juga dapat mengeksplorasi kombinasi model *Regresi Logistik* dengan teknik *ensemble* atau metode lain untuk meningkatkan akurasi dan kemampuan prediksi. Selain itu, disarankan untuk melakukan penelitian dengan sampel yang lebih besar dan beragam guna menguji ketahanan model ini dalam berbagai populasi. Penggunaan variabel tambahan yang relevan juga dapat membantu meningkatkan kinerja model dan memberikan wawasan yang lebih mendalam mengenai faktor-faktor yang berkontribusi terhadap kecemasan. Algoritma terbaik juga dapat dikembangkan kemudian untuk membuat aplikasi pendeteksi mental dengan *Android* atau *website* dengan *expert system* atau *machine learning*.

### UCAPAN TERIMA KASIH

Terima kasih kepada LPPM Universitas Global Jakarta dan program studi Teknik Informatika atas dukungan pada riset ini.

### DAFTAR PUSTAKA

- Aditya, M. A., Mulyana, R. D., Eka, I. P., & Widiyanto, S. R. (2020, February). Penggabungan Teknologi Untuk Analisa Data Berbasis Data Science. In Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) (Vol. 1, No. 1, pp. 51-56). Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 2(01), 20-28. <https://doi.org/10.38094/jastt20165>
- Putra, H. D., Khairani, L., & Hastari, D. (2023, August). Comparison of Naive Bayes Classifier and Support Vector Machine Algorithms for Classifying Student Mental Health Data: Perbandingan Algoritma Na-

- ive Bayes Classifier dan Support Vector Machine untuk Klasifikasi Data Kesehatan Mental Mahasiswa. In SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat (pp. 120-125).
- Ghosh, S. (2024). Comparing Regular Random Forest Model with Weighted Random Forest Model for Classification Problem. *International Journal of Statistics and Applications*, 2024(1), 7–12. <https://doi.org/10.5923/j.statistics.20241401.02>
- Golzar, J., & Noor, S. (2022). Simple Random Sampling. In *IJELS* (Issue 2).
- Kumar Barman, T., & Kumar, H. (2022). *Manifest Anxiety of University Students as related to their Sex and Educational Qualification*. 10. <https://doi.org/10.25215/1003.097>
- Lintang, L., Raras1, H., Daffa Pratama2, M., Nadhifa3, M., Nabilla, A., & Larasati4, S. E. (2023). *Hubungan Gangguan Kecemasan (Anxiety Disorder) dengan Prestasi Belajar Mahasiswa*.
- Muzumdar, P., Prasad Basyal, G., & Vyas, P. (2022). An Empirical Comparison of Machine Learning Models for Student's Mental Health Illness Assessment. In *Asian Journal of Computer and Information Systems* (Vol. 10, Issue 1). [www.ajouronline.com](http://www.ajouronline.com)
- Pratama, Y. A., Budiman, F., Winarno, S., & Kurniawan, D. (2023). *Analisis Optimasi Algoritma Decision Tree, Logistic Regression dan SVM Menggunakan Soft Voting*. 7, 1908–1919. <https://doi.org/10.30865/mib.v7i4.6856>
- Syamsu, M. (2021). *Peran Data Science dan Data Scientist Untuk Mentransformasi Data Dalam Industri 4.0* (Vol. 2, Issue 1).
- Aloysius, S., & Salvia, N. (2021). Analisis Kesehatan Mental Mahasiswa Perguruan Tinggi X Pada Awal Terjangkitnya Covid-19 di Indonesia. *Jurnal Citizenship Virtues*, 1(2), 83-97.
- Thenata, A. P., & Suryadi, M. (2022). Machine Learning Prediction of Anxiety Levels in the Society of Academicians During the Covid-19 Pandemic. *Jurnal Varian*, 6(1), 81–88. <https://doi.org/10.30812/varian.v6i1.2149>
- Wibowo, M., Rizieq, M., & Djafar, F. (2023). *Perbandingan Metode Klasifikasi Untuk Deteksi Stress Pada Mahasiswa di Perguruan Tinggi*. <https://doi.org/10.30865/mib.v7i1.5182>