

## Analisis Depresi Mahasiswa dengan Model *Machine Learning*

Abdullah Mubarak Maspeke<sup>1</sup> Lukman Nur Rahman<sup>2</sup>

<sup>1,2</sup> Teknik Informatika STMIK Tazkia

<sup>1</sup>[241552010001.maspeke@student.stmik.tazkia.ac.id](mailto:241552010001.maspeke@student.stmik.tazkia.ac.id)

<sup>2</sup>[241552010007.lukman@student.stmik.tazkia.ac.id](mailto:241552010007.lukman@student.stmik.tazkia.ac.id)

### Abstrak

Perhatian terhadap kondisi psikologis mahasiswa belakangan ini kian meningkat. Sayangnya, banyak sekali mahasiswa yang mengalami depresi tanpa teridentifikasi pada tahap awal. Untuk mengatasi hal ini, kami merancang sebuah program komputer cerdas yang bisa "belajar" mengenali gejala depresi dengan menganalisis data mahasiswa, termasuk usia, jurusan, dan nilai IPK mereka.

Pendekatan yang kami pakai adalah algoritma *Stochastic Gradient Descent* (SGD), sebuah metode yang melatih komputer secara bertahap untuk terus memperbaiki kemampuannya dalam membuat prediksi yang tepat. Sebagai bahan latihannya, kami memanfaatkan data dari survei publik di situs Kaggle yang telah kami sesuaikan terlebih dahulu.

Setelah diuji, program yang kami buat terbukti mampu membedakan kondisi mental mahasiswa dengan tingkat ketepatan yang baik. Hasil ini menandakan bahwa teknologi AI sangat berpotensi untuk dipakai sebagai alat deteksi awal masalah kejiwaan di dunia perkuliahan. Dengan begitu, mahasiswa yang membutuhkan bantuan bisa lebih cepat ditemukan dan ditangani.

**Kata kunci:** kesehatan mental, depresi mahasiswa, *machine learning*, klasifikasi, *stochastic gradient descent*

### Abstract

*Awareness of students' mental health has been growing in recent years. Unfortunately, many students who suffer from depression are not identified early. To help address this problem, we designed a smart computer program that can "learn" to recognize signs of depression by looking at student data such as age, major, and GPA.*

*The method we used is called the Stochastic Gradient Descent (SGD) algorithm. This is a way of training the computer step by step so it can gradually improve its ability to make accurate predictions. For training, we used survey data from the Kaggle platform, which we processed before applying it to our model.*

*When tested, our program was able to classify students' mental health conditions with good accuracy. These results show that artificial intelligence has strong potential to be used as an early detection tool for mental health problems in universities. With such a system, students who are at risk can be identified and supported more quickly.*

**Keywords:** mental health, student depression, machine learning, classification, stochastic gradient descent

## 1. Pendahuluan

Masalah kesehatan mental mahasiswa ini suda seharusnya tidak diremehkan dan sudah seharusnya menjadi urusan bersama, bukan lagi beban individu. Data pemerintah dan berita di media terus menunjukkan gambaran yang suram banyak sekali mahasiswa yang berjuang melawan tekanan psikologis akibat beratnya beban studi dan kehidupan sosial. Ini diperjelas oleh berbagai peristiwa menyedihkan yang pernah viral, seperti kasus bunuh diri akibat stres tugas akhir yang membuka mata kita tentang adanya penderitaan sunyi dan tak mampu diungkapkan di tengah kehidupan kampus yang tampak normal[1].

Mental yang sehat adalah dasar dari banyak hal bagi seorang pelajar. Jika ini rusak berbagai masalah akan muncul, mulai dari prestasi yang menurun, kehilangan minat pada segala hal, mengisolasi diri, hingga yang paling parah keinginan untuk mengakhiri hidup. Tetapi sebenarnya ada tembok penghalang besar yang datang dari rasa takut dan malu. Anggapan negatif dari lingkungan sekitar sering kali membuat mahasiswa memilih untuk memendam masalahnya sendiri daripada mencari bantuan.

Di tengah tantangan ini, kami percaya teknologi bisa menjadi bagian dari solusi. Penelitian ini mengusulkan penggunaan *machine learning* (kecerdasan buatan) sebagai sebuah terobosan. Idennya adalah melatih sebuah sistem komputer untuk mengenali tanda-tanda depresi dari data yang ada, seperti seorang detektif yang mencari pola dalam bukti. Karena bekerja secara objektif, sistem ini bisa melakukan deteksi awal tanpa menimbulkan stigma.

Penelitian ini memiliki tiga tujuan yang jelas. Satu, kami ingin membangun sebuah model yang bisa mengelompokkan mahasiswa ke dalam kategori risiko depresi. Dua, kami ingin tahu seberapa baik kinerja algoritma *Stochastic Gradient Descent (SGD)* untuk tugas tersebut. Tiga, kami ingin menunjukkan bahwa teknologi ini praktis dan bisa diandalkan sebagai metode skrining massal di universitas.

Harapannya sistem yang kami kembangkan ini dapat menjadi "asisten" bagi para konselor di kampus. Fungsinya bukan untuk mendiagnosis, melainkan untuk memberikan sinyal awal. Dengan adanya sinyal ini pihak kampus bisa lebih proaktif menjangkau mahasiswa yang membutuhkan dukungan dan memberikan bantuan yang tepat sebelum terlambat. Seluruh proses, dari data mentah hingga menjadi sebuah model yang berfungsi, akan kami coba sampaikan secara bertahap dalam laporan penelitian ini.

## 2. Metodologi

### 2.1. Sumber dan Modifikasi Data

Suber penelitian ini menggunakan data yang tersedia untuk publik (data sekunder) yang bersumber dari dataset "*Student Mental Health*" di platform Kaggle [2]. Dataset ini berisi jawaban survei dari mahasiswa mengenai berbagai aspek, termasuk status pernikahan, adanya riwayat depresi, kecemasan, jurusan, dan usia. Agar sesuai dengan tujuan penelitian, kami melakukan sedikit modifikasi pada data asli, yaitu dengan menambahkan kolom "*CGPA\_Grade*". Kolom ini berisi informasi rentang Indeks Prestasi Kumulatif (IPK) mahasiswa, sebuah faktor penting yang diduga berkaitan dengan tingkat stres dan kondisi mental mereka.

Gender	Age	Course	StudyYear	MaritalStatus	Depression	Anxiety	PanicAttack	SeekTreatment	CGPA	CGPA_Grade
Female	18	KIRKHS	year 1	FALSE	FALSE	FALSE	FALSE	FALSE	3.50 - 4.00	A
Female	24	Engineering	year 2	TRUE	TRUE	FALSE	TRUE	TRUE	2.50 - 2.99	C
Female	24	BIT	year 3	FALSE	FALSE	TRUE	FALSE	FALSE	3.00 - 3.49	B
Female	22	Engineering	year 4	FALSE	FALSE	FALSE	FALSE	FALSE	3.50 - 4.00	A
Female	20	Usuluddin	year 2	FALSE	TRUE	FALSE	FALSE	FALSE	3.00 - 3.49	B
Male		BIT	year 1	FALSE	FALSE	FALSE	FALSE	FALSE	0 - 1.99	E
Male	23	TAASL	year 2	FALSE	FALSE	FALSE	TRUE	FALSE	3.50 - 4.00	A
Male	18	BCS	year 1	FALSE	FALSE	TRUE	TRUE	FALSE	3.50 - 4.00	A
Female	19	Engineering	year 1	FALSE	FALSE	TRUE	FALSE	FALSE	3.50 - 4.00	A
Female	18	Engine	year 4	FALSE	FALSE	FALSE	FALSE	FALSE	3.50 - 4.00	A

Gambar 1: Potongan Dataset "*Student Mental Health*" setelah modifikasi

Suber data dari kaggle ini hanya memberikan data dari 101 mahasiswa/mahasiswi. Jumlah ini sejujurnya terlalu sedikit untuk membuat model yang benar-benar bisa mewakili kondisi semua mahasiswa. Ada juga kemungkinan jumlah mahasiswa yang mengalami depresi dan yang tidak depresi tidak seimbang. Hal ini bisa membuat model jadi belajar kurang adil dan lebih cenderung

ke salah satu kelompok saja. Keterbatasan lainnya ada pada jenis data yang dipakai. Data seperti usia, IPK, dan status pernikahan memang bisa memberi gambaran awal, tapi sebenarnya belum cukup untuk menunjukkan semua hal yang memengaruhi kesehatan mental. Faktor lain seperti dukungan keluarga atau teman, kondisi ekonomi, pengalaman hidup, dan tingkat stres sehari-hari juga penting untuk diperhitungkan. Hal-hal yang dapat mempengaruhi Kesehatan mental cenderung bersifat privasi jadi tidak menutup kemungkinan ada yg tidak jujur dalam mengisi kuesioner. Karena itu, penelitian dimasa depan sebaiknya memakai data yang lebih banyak dan jumlah kelas yang lebih seimbang serta variabel yang lebih beragam agar hasil prediksinya bisa lebih tepat dan bermanfaat kemudian menilai kejujuran pemberi data juga perlu ditimbnag ulang.

## 2.2. Persiapan Data

Sebelum data diproses oleh model *computer* data tersebut perlu dibersihkan dan disiapkan terlebih dahulu. Proses ini bisa diibaratkan dengan koki yang menyiapkan bahan-bahan sebelum memasak. Beberapa bahan mentah perlu dicuci, dipotong, atau diubah bentuknya agar masakan menjadi lezat.

Dalam penelitian ini, kami melakukan beberapa hal:

- **Mengubah Data Teks menjadi Angka:** Komputer tentunya lebih mudah memahami angka daripada teks. Jadi, data seperti "Ya" atau "Tidak" pada kolom depresi diubah menjadi angka 1 dan 0. Demikian pula, rentang IPK seperti "3.50 - 4.00" diubah menjadi angka representatif, yaitu 3.75 (Memudahkan dalam perhitungan dengna rumus ini).
- **Menbuat Standar Skala Angka:** Beberapa data memiliki rentang yang berbeda, misalnya usia (belasan hingga puluhan) dan IPK (0 hingga 4). Agar si komputer tetap adil dan tidak pilih kasih pada angka yang lebih besar, semua data numerik disamakan skalanya melalui proses *standardization* ini.

## 2.3. Cara Kerja Model

Algoritma utama yang kami gunakan adalah *Stochastic Gradient Descent* (SGD). Cara kerjanya bisa dianalogikan seperti seseorang yang memiliki urusan penting dan mencoba menuruni bukit berkabut tebal untuk mencari titik terendah dibukit itu. Karena tak bisa melihat jauh, orang itu akan mengambil satu langkah kecil ke arah yang paling curam, berhenti sejenak, merasakan pijakannya, lalu mengambil langkah kecil berikutnya.

Sama halnya dengan *SGD*. Model ini diajari dgn diberikan satu per satu data mahasiswa. Untuk setiap data, model akan menebak apakah mahasidwa tersebut depresi atau tidak. Jika tebakanya salah, model akan sedikit mengoreksi arahnya. Proses ini diulang ratusan bahkan mungkin ribuan kali dengan data yang berbeda-beda dan setiap kesalahan menjadi pelajaran bagi model untuk memperbaiki diri. Sedikit demi sedikit lama kelamaan model menjadi semakin pintar dalam membedakan pola antara mahasiswa yang depresi dan yang tidak.

## 2.4. Pengujian Model

Untuk mengetahui seberapa pintar model yang telah dilatih, kami menggunakan teknik yang umum dalam *machine learning*. Kami membagi seluruh data yang kami miliki menjadi dua bagian:

### 1. Data Latihan (80%)

Data Ini bisa dianggap kaya *cheat book* yang diberikan kepada model untuk belajar. Model akan mempelajari semua pola dari data ini.

#### 2. Data Ujian (20%)

Data ini dirahasiakan dari model selama proses belajar. Setelah model selesai belajar, kami mengujinya dengan data ini untuk melihat seberapa baik performanya pada data yang belum pernah ia lihat sebelumnya.

Keberhasilan model diukur dengan metrik akurasi khusus, yaitu persentase tebakan benar yang berhasil dibuat oleh model pada data ujian.

#### 2.5. Tools

Pembuatan model ini menggunakan beberapa hal penting diantaranya:

##### 1. Python

Kami menggunakan Python untuk menulis semua perintah yang harus dijalankan oleh komputer. Bahasa ini dipilih karena sangat populer dan punya banyak fitur siap pakai yang mempercepat pekerjaan. Selain itu bahasa ini sudah kami pelajari di kelas sehingga lebih familier.

##### 2. Jupyter Notebook

Sebagian besar proses kami kerjakan di Jupyter Notebook. Ini adalah sebuah lembar kerja digital di mana kami bisa menulis perintah, menjalankan perintah itu, dan langsung melihat hasilnya, baik berupa teks maupun gambar, di satu tempat. Ini juga sangat disarankan oleh dosen kami.

##### 3. Alat Bantu Tambahan (Library)

Kami juga memakai beberapa alat bantu tambahan untuk tugas-tugas spesifik:

- o **scikit-learn:** Ini adalah alat utama untuk membuat program kecerdasan buaatannya. Di sinilah semua proses "belajar" komputer terjadi.
- o **pandas:** Alat ini dipakai untuk mengelola data. Tugasnya adalah membaca dan merapikan tabel data agar siap untuk dianalisis.
- o **numpy:** Berfungsi sebagai kalkulator canggih untuk menangani semua proses hitung-hitungan yang berat.
- o **matplotlib & seaborn:** Ini adalah alat untuk menggambar. Kami menggunakannya untuk mengubah data angka menjadi grafik yang lebih mudah dilihat dan dipahami.

### 3. Hasil Dan Pembahasan

Pada bab ini, kita akan membahas hasil dari eksperimen kita. Analisis difokuskan pada perbandingan kinerja antara model *Linear Regression* dan *Stochastic Gradient Descent (SGD) Classifier*.

#### 3.1. Hasil Utama

Eksperimen dilakukan dengan membagi dataset menjadi 80 sampel untuk pelatihan (*training set*) dan 21 sampel untuk pengujian (*test set*) dengan 7 kolom dari dataset yang sudah disempurnakan untuk model ini. Evaluasi performa dilakukan pada *test set* untuk mengukur kemampuan generalisasi model terhadap data baru.

```
Data berhasil dimuat. Memulai persiapan data untuk Machine Learning...  
  
Ukuran data latih (X_train): (80, 7)  
Ukuran data uji (X_test): (21, 7)
```

Gambar 2: Pembagian Dataset untuk pelatihan dan pengujian

Hasil evaluasi menunjukkan perbedaan performa yang signifikan antara kedua model:

- Model *Linear Regression*, setelah penerapan thresholding pada nilai prediksinya mencapai akurasi sebesar 80.95%. Model ini juga mencatatkan nilai *Mean Squared Error (MSE)* sebesar 0.1867 pada prediksi mentahnya.
- Model *Stochastic Gradient Descent (SGD) Classifier* menunjukkan akurasi yang lebih rendah, yaitu 66.67%. Berdasarkan metrik akurasi dapat disimpulkan bahwa model *Linear Regression* menunjukan performa yang lebih buruk dibandingkan *SGD Classifier* dalam tugas klasifikasi ini.

### 3.2. Analisis Kinerja Model

Analisis tidak hanya terbatas pada akurasi agar lebih mudah dipahami. Metrik *Precision* dan *Recall* dari *Classification Report* dievaluasi untuk menganalisis perilaku model terhadap setiap kelas, terutama kelas target '*Depression*'.

- Analisis Kinerja *SGD Classifier***  
Model ini menunjukkan nilai Recall sebesar 0.57 untuk kelas '*Depression*'. Hal ini mengindikasikan bahwa model hanya mampu mengidentifikasi 57% dari total kasus positif aktual, sehingga terdapat tingkat *False Negative* yang signifikan (43%). Selain itu, nilai Precision sebesar 0.50 menunjukkan bahwa dari seluruh instansi yang diprediksi sebagai '*Depression*', hanya separuhnya yang benar-benar merupakan kasus positif.

```
--- Melatih Model: Stochastic Gradient Descent (SGD) Classifier ---  
  
Akurasi: 0.6667  
  
Laporan Klasifikasi (SGD Classifier):  
      precision    recall  f1-score   support  
  
      0       0.77       0.71       0.74        14  
      1       0.50       0.57       0.53         7  
  
   accuracy          0.67        0.67        21  
  macro avg       0.63       0.64       0.64        21  
 weighted avg       0.68       0.67       0.67        21
```

Gambar 3: Hasil pelatihan *SGD Classifier*

- Analisis Kinerja *Linear Regression***  
Model ini juga mencatatkan nilai Recall yang identik (0.57), yang menandakan keterbatasan serupa dalam hal sensitivitas atau kemampuan mendeteksi semua kasus positif. Namun, keunggulannya terletak pada nilai Precision yang jauh lebih tinggi, yaitu 0.80. Artinya, meskipun kedua model memiliki keterbatasan dalam menjaring seluruh kasus, prediksi positif yang dihasilkan oleh *Linear Regression* memiliki tingkat keandalan yang jauh lebih tinggi.

```
--- Melatih Model: Linear Regression ---  
  
Mean Squared Error (MSE) dari prediksi mentah: 0.1867  
Akurasi setelah thresholding (> 0.5): 0.8095  
  
Laporan Klasifikasi (Linear Regression):  
      precision    recall  f1-score   support  
  
      0         0.81      0.93      0.87         14  
      1         0.80      0.57      0.67          7  
  
   accuracy              0.81         21  
  macro avg         0.81      0.75      0.77         21  
 weighted avg         0.81      0.81      0.80         21
```

Gambar 4: Hasil pelatihan *Linear Regression*

### 3.3. Pembahasan

Hasil penelitian menunjukkan bahwa model *Linear Regression* dengan akurasi 80.95% memiliki potensi sebagai teknologi pendukung dalam identifikasi risiko depresi di kalangan mahasiswa. Namun, analisis kinerja mengungkap adanya limitasi kritis, yaitu nilai *Recall* yang rendah (57%) pada kedua model. Tingginya angka *False Negative* ini merupakan risiko substansial dalam konteks deteksi kondisi kesehatan, di mana kegagalan mengidentifikasi individu berisiko dapat menunda intervensi yang diperlukan.

Oleh karena itu, implementasi praktis dari model ini bukanlah sebagai alat diagnostik, melainkan sebagai sistem penyaringan awal (*preliminary screening tool*). Tujuannya adalah untuk mengidentifikasi individu yang mungkin memerlukan perhatian lebih lanjut dan merekomendasikan mereka untuk berkonsultasi dengan layanan konseling profesional.

Keunggulan *Linear Regression* dalam hal *Precision* (80%) menjadikannya kandidat yang lebih sesuai untuk peran ini, karena dapat meminimalkan jumlah peringatan yang keliru (*False Positives*) dan menjaga kepercayaan pengguna terhadap sistem.

### 3.4. kode python

```
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler, OneHotEncoder  
from sklearn.compose import ColumnTransformer  
from sklearn.linear_model import LinearRegression, SGDClassifier  
from sklearn.pipeline import Pipeline  
from sklearn.metrics import mean_squared_error, accuracy_score,  
classification_report, confusion_matrix
```

```
from sklearn.impute import SimpleImputer
import numpy as np

#
=====
#
# BAGIAN 1: MEMBACA DAN MEMPERSIAPKAN DATA
#
=====
#
# Membaca data dari file CSV"
try:
    df = pd.read_csv("Dataset.csv", sep=';')
except Exception as e:
    print(f"Error membaca file dengan separator ';': {e}")
    print("Mencoba membaca dengan separator ','...")
    df = pd.read_csv("Dataset.csv")

print("Data berhasil dimuat. Memulai persiapan data untuk Machine
Learning...")

# Mendefinisikan variabel target yang akan diprediksi
TARGET = 'Depression'

# Membuat pemetaan untuk kolom CGPA
cgpa_map = {
    '0 - 1.99': 1.0,
    '2.00 - 2.49': 2.25,
    '2.50 - 2.99': 2.75,
    '3.00 - 3.49': 3.25,
    '3.50 - 4.00': 3.75
}

# Mengonversi semua kolom boolean/string boolean menjadi format numerik
(1/0)
for col in ['Gender', 'MaritalStatus', 'Depression', 'Anxiety',
'PanicAttack', 'SeekTreatment']:
    if col in df.columns:
        if df[col].dtype == 'object':
            df[col] = df[col].apply(lambda x: 1 if x.strip().lower() in
['female', 'true'] else 0)
        elif df[col].dtype == 'bool':
            df[col] = df[col].astype(int)
```

```
#Menambahkan kolom CGPA_Numeric ke dataframe
df['CGPA_Numeric'] = df['CGPA'].map(cgpa_map) # Baris ini
menerjemahkan IPK dari teks ke angka
# Mengisi nilai IPK yang kosong dengan nilai rata-rata dari semua IPK
yang ada
df['CGPA_Numeric'].fillna(df['CGPA_Numeric'].mean(), inplace=True)
# Baris ini tetap sama, untuk menghapus kolom yang tidak perlu
df.drop(columns=['CGPA_Grade'], inplace=True, errors='ignore')

# Mendefinisikan fitur (X) dan target (y)
# Fitur yang tidak relevan atau menyebabkan kebocoran data akan dibuang
features_to_drop = ['Timestamp', TARGET, 'Anxiety', 'PanicAttack',
'CGPA', 'CGPA_Grade']
X = df.drop(columns=features_to_drop, errors='ignore')
y = df[TARGET]

# Mengidentifikasi kolom kategorikal dan numerik untuk preprocessing
X['StudyYear'] = X['StudyYear'].astype(str)
categorical_features = ['Course', 'StudyYear']
numerical_features = ['Age', 'CGPA_Numeric']

# Membuat pipeline untuk fitur numerik yang menangani nilai kosong
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')), # Mengisi nilai kosong
    ('scaler', StandardScaler()) # Melakukan scaling
])

#Membuat preprocessor untuk mentransformasi kolom
preprocessor = ColumnTransformer(
    transformers=[
        # Gunakan pipeline numerik yang baru
        ('num', numeric_transformer, numerical_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'),
categorical_features)
    ],
    remainder='passthrough'
)

# Memisahkan data menjadi data latih dan data uji (80% latih, 20% uji)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random state=42, stratify=y)
```



```
print(f"\nUkuran data latih (X_train): {X_train.shape}")
print(f"Ukuran data uji (X_test): {X_test.shape}")

#
=====
=====
# BAGIAN 2: ALGORITMA 1 - LINEAR REGRESSION (UNTUK KLASIFIKASI)
#
=====
=====
print("\n--- Melatih Model: Linear Regression ---")

# Membuat pipeline: Preprocessing -> Model
lr_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                              ('regressor', LinearRegression())])

# Melatih model
lr_pipeline.fit(X_train, y_train)

# Membuat prediksi (hasilnya adalah nilai kontinu)
y_pred_lr_raw = lr_pipeline.predict(X_test)

# Mengonversi hasil prediksi menjadi kelas biner (0 atau 1) dengan
threshold 0.5
y_pred_lr = (y_pred_lr_raw > 0.5).astype(int)

# Evaluasi model
mse = mean_squared_error(y_test, y_pred_lr_raw)
accuracy_lr = accuracy_score(y_test, y_pred_lr)

print(f"\nMean Squared Error (MSE) dari prediksi mentah: {mse:.4f}")
print(f"Akurasi setelah thresholding (> 0.5): {accuracy_lr:.4f}")
print("\nLaporan Klasifikasi (Linear Regression):")
print(classification_report(y_test, y_pred_lr, zero_division=0))

#
=====
=====
# BAGIAN 3: ALGORITMA 2 - STOCHASTIC GRADIENT DESCENT (SGD)
#
=====
=====
```

```
print("\n--- Melatih Model: Stochastic Gradient Descent (SGD) Classifier\n---")

# Membuat pipeline: Preprocessing -> Model
# loss='log_loss' membuat SGDClassifier bekerja seperti Regresi Logistik
sgd_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                ('classifier',
                                 SGDClassifier(loss='log_loss', random_state=42, max_iter=1000, tol=1e-3))])

# Melatih model
sgd_pipeline.fit(X_train, y_train)

# Membuat prediksi (hasilnya sudah dalam bentuk kelas 0 atau 1)
y_pred_sgd = sgd_pipeline.predict(X_test)

# Evaluasi model
accuracy_sgd = accuracy_score(y_test, y_pred_sgd)
print(f"\nAkurasi: {accuracy_sgd:.4f}")

print("\nLaporan Klasifikasi (SGD Classifier):")
print(classification_report(y_test, y_pred_sgd, zero_division=0))

print("\nConfusion Matrix (SGD Classifier):")
cm = confusion_matrix(y_test, y_pred_sgd)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No Depression', 'Depression'], yticklabels=['No Depression', 'Depression'])
plt.title('Confusion Matrix - SGD Classifier')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

#### 4. Kesimpulan

Penelitian ini menunjukkan bahwa algoritma *machine learning Stochastic Gradient Descent (SGD)* dapat digunakan untuk membuat model klasifikasi depresi pada mahasiswa dengan hasil yang cukup akurat. Data akademik dan demografis terbukti bisa menjadi dasar awal bagi teknologi untuk membantu proses *skrining* kesehatan mental sejak dini.

Dari sisi manfaat praktis, penelitian ini memberi peluang bagi perguruan tinggi untuk mengembangkan sistem pendukung keputusan yang dapat membuat layanan konseling lebih efektif dan responsif. Untuk tahap lanjutan, model ini masih bisa ditingkatkan dengan menambahkan faktor lain, seperti pola tidur, interaksi sosial, analisis teks dari media social, atau bahkan ketaatan beragama agar hasil prediksinya semakin tepat.

Seperti yang pernah dikatakan oleh psikolog dan pendidik nasional, Seto Mulyadi (Kak Seto) bahwa “mahasiswa harus bisa tumbuh dalam suasana yang menyenangkan.”. Teknologi yang dikembangkan melalui penelitian ini diharapkan dapat menjadi salah satu upaya agar tekanan akademik tidak merampas kebahagiaan sekaligus kesehatan mental para mahasiswa yang berujung pada hal-hal buruk yang tidak diinginkan sekalipun.

### Ucapan Terima Kasih

Dengan penuh rasa terima kasih dan rasa hormat, kami menyampaikan apresiasi kepada Bapak Hendri Karisma, S.Kom., M.T. secara mendalam, atas segala pengajaran, bimbingan, motivasi, serta saran yang telah membantu dalam penyusunan penelitian ini. Penulis juga menyampaikan rasa terima kasih dan rasa hormatnya kepada pihak penyedia dataset melalui platform Kaggle yang telah memfasilitasi tersedianya data publik sebagai dasar penelitian ini. Penulis juga menyampaikan rasa terima kasih ke semua pihak yang telah menyediakan artikel online dan pdf yang sangat membantu dalam penyusunan penelitian ini.

### Daftar Pustaka

- [1] Kementerian Kesehatan RI, “Orang Depresi yang Tak Tertangani Rawan Bunuh Diri,” *Sehat Negeriku*, 2019. [Online]. Tersedia: <https://kemkes.go.id/id/orang-depresi-tak-tertangani-rawan-bunuh-diri>
- [2] B. Bekci, “Student Mental Health Analysis,” Kaggle, 2023. [Online]. Tersedia: <https://www.kaggle.com/code/berkaybekci/student-mental-health-analysis>
- [3] A. F. Hidayatullah dan I. K. R. Arthana, “Prediksi Tingkat Depresi Mahasiswa Selama Pandemi Covid-19 Menggunakan Algoritma Naïve Bayes,” *Jurnal Rekayasa Sistem & Industri*, vol. 8, no. 2, hlm. 129-138, 2021. [Online]. Tersedia: <https://jrsi.sie.telkomuniversity.ac.id/index.php/jrsi/article/view/1547>
- [4] D. A. Wahyuni dan R. I. M. Zen, “Analisis Faktor-Faktor yang Mempengaruhi Tingkat Stres Mahasiswa,” *Jurnal Psikologi*, vol. 16, no. 1, hlm. 55-65, 2020. [Online]. Tersedia: <https://journal.uii.ac.id/Psikologika/article/view/14771>
- [5] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, hlm. 2825-2830, 2011. [Online]. Tersedia: <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>
- [6] World Health Organization, “Depression,” WHO, 2023. [Online]. Tersedia: <https://www.who.int/news-room/fact-sheets/detail/depression>

[7] R. Lazarus dan S. Folkman, *Stress, Appraisal, and Coping*. New York: Springer, 1984. [Online]. Tersedia: [https://www.google.co.id/books/edition/Stress\\_Appraisal\\_and\\_Coping/Gv3t\\_3lj-x4C?hl=en&gbpv=0](https://www.google.co.id/books/edition/Stress_Appraisal_and_Coping/Gv3t_3lj-x4C?hl=en&gbpv=0)

[8] UGM, “Kementerian Kesehatan Ungkap Kasus Bunuh Diri Meningkat,” Universitas Gadjah Mada, 2023. [Online]. Tersedia: <https://ugm.ac.id/id/berita/kementerian-kesehatan-ungkap-kasus-bunuh-diri-meningkat-hingga-826-kasus/>