

# Analisis Perbandingan Algoritma K-Nearest Neighbor dan Decision Tree dalam Klasifikasi Tingkat Obesitas

Rizki Wahyu Yulianti\*, Fikri Budiman, Defri Kurniawan

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: <sup>1,\*</sup>111202214212@mhs.dinus.ac.id, <sup>2</sup>fikri.budiman@dsn.dinus.ac.id, <sup>3</sup>defri.kurniawan@dsn.dinus.ac.id

Email Penulis Korespondensi: 111202214212@mhs.dinus.ac.id\*

Submitted: 26/10/2025; Accepted: 26/11/2025; Published: 31/12/2025

**Abstrak**— Kasus obesitas yang terus bertambah di era digital dipengaruhi oleh perubahan gaya hidup, pola makan, aktivitas fisik, serta faktor demografis. Untuk mendukung upaya deteksi dini, penelitian ini menerapkan metode machine learning guna mengklasifikasikan tingkat obesitas berdasarkan *Obesity Levels Dataset* yang memuat 2.111 data individu dengan 17 variabel terkait perilaku dan kondisi fisik. Penelitian berfokus pada analisis perbandingan dua algoritma klasifikasi, yaitu *K-Nearest Neighbor (KNN)* dan *Decision Tree*, dengan tujuan menilai akurasi, presisi, *recall*, serta *stabilitas* performa kedua model setelah melalui proses optimasi. Tahapan penelitian dimulai dengan pembersihan data, *imputasi* nilai hilang menggunakan *median*, penghapusan *outlier* melalui metode *IQR*, *encoding* variabel kategori, serta standarisasi fitur numerik. Untuk menangani ketidakseimbangan kelas, digunakan teknik *SMOTE* sehingga setiap kategori kelas target memiliki distribusi yang seimbang. Model kemudian dilatih menggunakan data latih sebesar 80% dan dievaluasi pada 20% data uji. Optimasi *parameter* dilakukan menggunakan *GridSearchCV*. Pada model *K-Nearest Neighbor*, nilai *k* terbaik diperoleh pada *k* = 3 dengan akurasi 86,59%. Sementara itu, *Decision Tree* mencapai performa optimal pada konfigurasi *max\_depth* = 10, *min\_samples\_split* = 5, dan *min\_samples\_leaf* = 1. Hasil evaluasi menunjukkan bahwa kedua algoritma mampu melakukan klasifikasi tingkat obesitas dengan baik setelah proses tuning dan penyeimbangan data. *Decision Tree* memberikan performa yang lebih stabil dalam mengenali pola kompleks antarvariabel, sedangkan *K-Nearest Neighbor* menunjukkan performa optimal pada nilai *k* kecil ketika data telah distandarisasi dan diseimbangkan. Penelitian ini memberikan gambaran empiris mengenai efektivitas kedua algoritma dalam pemodelan klasifikasi obesitas serta dapat menjadi dasar pengembangan sistem prediksi kesehatan berbasis *machine learning* yang lebih akurat dan efisien.

**Kata Kunci:** Obesitas; Decision Tree; K-Nearest Neighbor; Hyperparameter Tuning; Machine Learning

**Abstract**— The increasing cases of obesity in the digital era are influenced by changes in lifestyle, diet, physical activity, and demographic factors. To support early detection efforts, this study applies machine learning methods to classify obesity levels based on the Obesity Levels Dataset, which contains 2,111 individual data with 17 variables related to behavior and physical condition. The study focuses on a comparative analysis of two classification algorithms, namely K-Nearest Neighbor (KNN) and Decision Tree, with the aim of assessing the accuracy, precision, recall, and stability of the performance of both models after going through an optimization process. The research stages begin with data cleaning, imputation of missing values using the median, removal of outliers using the IQR method, encoding categorical variables, and standardization of numeric features. To address class imbalance, the SMOTE technique is used so that each target class category has a balanced distribution. The model is then trained using 80% of the training data and evaluated on 20% of the test data. Parameter optimization is performed using GridSearchCV. In the K-Nearest Neighbor model, the best *k* value is obtained at *k* = 3 with an accuracy of 86.59%. Meanwhile, Decision Tree achieved optimal performance at *max\_depth* = 10, *min\_samples\_split* = 5, and *min\_samples\_leaf* = 1 configuration. The evaluation results showed that both algorithms were able to classify obesity levels well after the data tuning and balancing process. Decision Tree provided more stable performance in recognizing complex patterns between variables, while K-Nearest Neighbor showed optimal performance at small *k* values when the data had been standardized and balanced. This study provides an empirical overview of the effectiveness of both algorithms in modeling obesity classification and can be the basis for developing a more accurate and efficient machine learning-based health prediction system.

**Keywords:** Obesity; Decision Tree; K-Nearest Neighbor; Hyperparameter Tuning; Machine Learning

## 1. PENDAHULUAN

Pesatnya kemajuan teknologi informasi telah membawa banyak perubahan dalam kehidupan sehari-hari. Kini, hampir semua aktivitas bisa dilakukan dengan bantuan perangkat digital. Mulai dari kemudahan mengakses layanan, meningkatnya durasi penggunaan gawai (*screen time*), hingga kebiasaan memesan makanan lewat aplikasi[1]. Perubahan gaya hidup tersebut secara perlakuan memengaruhi pola makan dan menurunkan aktivitas fisik, sehingga berperan dalam peningkatan kasus obesitas di masyarakat.

Saat ini, obesitas menjadi salah satu permasalahan kesehatan serius di Indonesia. Berbagai penelitian, baik nasional maupun daerah, menunjukkan bahwa gaya hidup modern seperti sering mengonsumsi makanan olahan, kurang tidur, tingginya *screen time*, serta rendahnya aktivitas fisik berhubungan erat dengan risiko obesitas, baik pada remaja maupun orang dewasa[2][3]. Obesitas tidak hanya menjadi masalah penampilan fisik, tetapi juga memiliki keterkaitan yang kuat dengan berbagai penyakit kronis, seperti diabetes, hipertensi, dan gangguan jantung[4][5]. Dengan memahami dan mengenali obesitas sejak awal, langkah pencegahan dapat dijalankan dengan lebih efektif. Perilaku sehari-hari memegang peran penting dalam munculnya obesitas. Berbagai penelitian di Indonesia menunjukkan bahwa kebiasaan mengonsumsi makanan cepat saji dan produk olahan ultra-proses berkaitan dengan meningkatnya risiko kelebihan berat badan, khususnya pada remaja dan dewasa muda[6]. Kecenderungan tersebut

semakin didorong oleh mudahnya akses dan maraknya promosi di platform digital. Di luar pola makan, penelitian juga menunjukkan bahwa obesitas dipengaruhi oleh berbagai faktor lain yang saling berkaitan, seperti kondisi demografi, lingkungan, keluarga, hingga faktor genetik[7][8].

Di era digital, teknologi tidak hanya memudahkan akses informasi, tetapi juga semakin banyak dimanfaatkan di bidang kesehatan. Salah satunya melalui *machine learning*, yaitu algoritma komputer yang dapat mengenali pola dari data untuk menghasilkan prediksi dengan tingkat akurasi yang tinggi[9][10]. Dalam konteks obesitas, machine learning dapat digunakan untuk mengenali faktor risiko serta memperkirakan tingkat obesitas seseorang berdasarkan data perilaku, demografi, maupun kebiasaan hidup[11][12]. Dari sekian banyak algoritma yang tersedia, *K-Nearest Neighbor (KNN)* dan *Decision Tree* termasuk metode klasifikasi yang sering digunakan. Keduanya memiliki cara kerja yang berbeda, di mana *K-Nearest Neighbor* menentukan kelas berdasarkan kedekatan data baru dengan data lama, sedangkan *Decision Tree* membangun model berbentuk pohon keputusan dari atribut-atribut penting[13].

Dalam proses klasifikasi tingkat obesitas menggunakan algoritma *machine learning*, masih terdapat berbagai tantangan yang memengaruhi tingkat keakuratannya. Salah satu kendala utama adalah ketidakseimbangan data antara kategori berat badan normal, kelebihan berat badan, dan obesitas. Kondisi ini menyebabkan model lebih mudah mengenali pola dari kelas dominan, namun kurang optimal dalam mengklasifikasikan kelas lainnya. Selain itu, faktor penyebab obesitas yang kompleks seperti kebiasaan makan, aktivitas fisik, pola tidur, dan kondisi demografis menyulitkan algoritma untuk memahami keterkaitan antarvariabel secara tepat.

Dari sisi metode, algoritma *K-Nearest Neighbor* memiliki ketergantungan tinggi pada pemilihan nilai k dan skala data, sehingga kesalahan dalam menentukan parameter dapat menurunkan performa model[14]. Sementara itu, algoritma *Decision Tree* sering mengalami masalah *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data latih dan kehilangan kemampuan menyamaratakan terhadap data baru[15][16]. Hingga saat ini, belum banyak penelitian yang secara langsung membandingkan efektivitas *K-Nearest Neighbor* dan *Decision Tree* dalam memprediksi tingkat obesitas di Indonesia. Oleh karena itu, diperlukan analisis perbandingan kedua algoritma tersebut untuk menilai performa, kestabilan, serta tingkat akurasi dalam proses klasifikasi obesitas.

Berbagai penelitian di Indonesia telah menerapkan algoritma machine learning untuk klasifikasi tingkat obesitas. Penelitian oleh Utiarahman & Pratama [17] menunjukkan bahwa algoritma *Decision Tree* mampu mencapai akurasi hingga 99,3%, sedikit lebih tinggi dibandingkan *K-Nearest Neighbor* sebesar 99,0% dalam memprediksi tingkat obesitas berdasarkan data perilaku dan pola makan. Sementara itu, penelitian oleh Prakoso et al [18] menemukan bahwa algoritma *Decision Tree* memiliki akurasi tertinggi sebesar 95,13%, diikuti oleh *Random Forest* dengan 83,23%, dan *K-Nearest Neighbor* sebesar 83,44% dalam klasifikasi obesitas berdasarkan faktor gaya hidup. Hasil ini menunjukkan bahwa *Decision Tree* cenderung lebih unggul dalam mempelajari pola kompleks antara variabel perilaku dan demografi.

Hasil-hasil penelitian tersebut menunjukkan bahwa *K-Nearest Neighbor* dan *Decision Tree* banyak digunakan dan menghasilkan nilai akurasi yang baik, tetapi belum dilakukan optimasi perbandingan *K-Nearest Neighbor* dan *Decision Tree* pada klasifikasi tingkat obesitas, sehingga pada eksperimen ini akan dilakukan perbandingan untuk kedua algoritma tersebut. Untuk menilai keberhasilan perlu dilakukan pengujian akurasi, *recall*, *precision*, dan *F1-score* dengan metode *Confusion Matrix*, sehingga hasil pengujian mampu memberikan gambaran yang lebih menyeluruh terhadap kemampuan klasifikasi masing-masing algoritma[19]. Pada eksperimen riset ini dilakukan tahapan-tahapan sebagai berikut: dimulai dari penyeimbangan data dengan *SMOTE*, dilanjutkan dengan *Hyperparameter Tuning* untuk memperoleh konfigurasi optimal, optimasi nilai K khusus pada model *K-Nearest Neighbor*, kemudian perbandingan performa *K-Nearest Neighbor* dan *Decision Tree*, dan diakhiri dengan pengujian akhir untuk memastikan kemampuan model dalam mengklasifikasikan data secara akurat. Tujuan dari penelitian ini adalah untuk mendapatkan optimasi terbaik penggunaan algoritma antara *K-Nearest Neighbor* dan *Decision Tree* dalam Obesity Data. Dalam penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan model prediksi obesitas yang lebih akurat dan efisien berbasis algoritma *mechine learning*.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Dalam penelitian ini, menggunakan aplikasi *Google Colab* sebagai lingkungan simulasi dan pemrosesan data dengan bahasa pemrograman *Python 3.10*. *Google Colab* dipilih karena mendukung berbagai pustaka *machine learning* seperti *Scikit-learn*, *Imbalanced-learn*, *Pandas*, *NumPy*, serta dapat terintegrasi langsung dengan Google Drive untuk pengambilan dataset.

Dataset yang digunakan pada penelitian ini berasal dari *Obesity Levels Dataset* yang diunduh dari platform (<https://www.kaggle.com/datasets/juliaangladalomaeva/obesity-csv>) yaitu data obesitas berdasarkan gaya hidup dan kondisi fisik individu. Dataset yang digunakan berjudul *Obesity Data Set* dalam format .csv, dengan total 2111 baris data dan 17 atribut yang digunakan untuk mengidentifikasi tingkat obesitas seseorang.

Variabel yang digunakan dalam penelitian ini mencakup jenis kelamin (*Gender*), usia (*Age*), tinggi badan (*Height*), berat badan (*Weight*), riwayat keluarga dengan kelebihan berat badan (*family history with overweight*), serta sejumlah faktor perilaku dan kebiasaan hidup seperti konsumsi makanan berlemak (FAVC), frekuensi konsumsi

sayuran (FCVC), jumlah makanan utama per hari (NCP), kebiasaan ngemil di antara waktu makan (CAEC), serta asupan air harian (CH2O). Selain itu, terdapat variabel terkait aktivitas fisik dan kondisi tubuh, yaitu frekuensi aktivitas fisik mingguan (FAF), waktu yang dihabiskan untuk penggunaan perangkat elektronik (TUE), kebiasaan memantau asupan kalori (SCC), jenis transportasi yang digunakan sehari-hari (Mtrans), dan konsumsi alkohol (CALC).

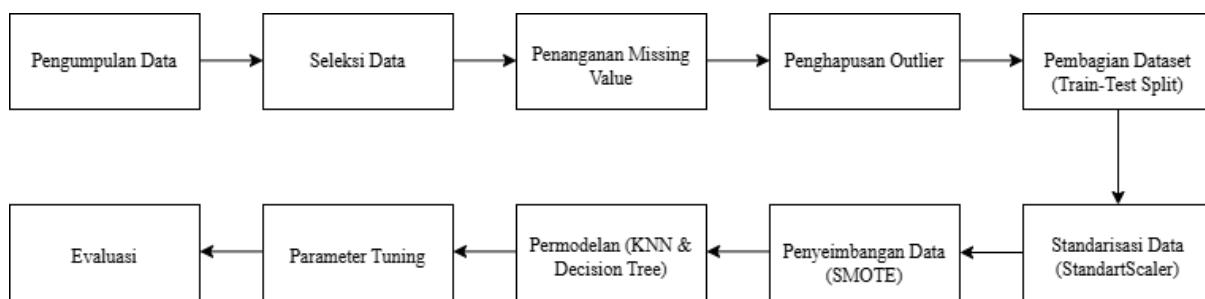
Atribut NObeyesdad berfungsi sebagai variabel target, yang mengklasifikasikan tingkat obesitas individu ke dalam beberapa kategori seperti *Insufficient Weight*, *Normal Weight*, *Overweight*, dan *Obesity*. Dataset ini merupakan data sekunder yang berasal dari sumber publik dan dapat diakses secara terbuka tanpa pengumpulan langsung dari responden.

**Tabel 1.** Variabel penelitian

Nama Variabel	Keterangan
Gender	kelamin
Age	usia
Height	tinggi badan
Weight	berat badan
family history with overweight	riwayat keluarga dengan kelebihan berat badan
FAVC	frekuensi konsumsi makanan tinggi kalori / berlemak
FCVC	frekuensi konsumsi sayuran setiap hari
NCP	Jumlah makanan utama setiap hari
CAEC	kebiasaan ngemil di antara waktu makan
CH2O	asupan air harian
FAF	frekuensi aktivitas fisik mingguan
TUE	durasi penggunaan perangkat elektronik
SCC	kebiasaan memantau asupan kalori
Mtrans	jenis transportasi sehari-hari
CALC	kebiasaan konsumsi alkohol
SMOKE	kebiasaan merokok
NObeyesdad	kategori tingkat obesitas

## 2.2 Desain dan Alur Penelitian

Penelitian ini merupakan studi *eksperiment* yang bertujuan untuk memberikan gambaran secara sistematis mengenai langkah-langkah yang dilakukan dalam proses klasifikasi tingkat obesitas dengan memanfaatkan algoritma *machine learning*. Fokus utama penelitian ini terletak pada analisis perbandingan antara algoritma *K-Nearest Neighbor* dan *Decision Tree* dalam memprediksi tingkat obesitas berdasarkan data gaya hidup dan kondisi fisik individu. Secara umum, proses penelitian ini dimulai dari tahap pengumpulan dan pemilihan dataset, dilanjutkan dengan tahap *pre-processing* yang meliputi pembersihan data, penanganan *missing value*, penghapusan *outlier*, sampai pada tahap terakhir.



**Gambar 1.** Alur Penelitian

## 2.3 Proses Analisis dan Evaluasi

Tahapan penelitian ini meliputi pra-pemrosesan data dengan menghapus *duplicasi*, mengisi *missing value* menggunakan median, menghilangkan *outlier* melalui metode *IQR* (*interquartile range*), serta mengubah variabel kategori menjadi numerik menggunakan *LabelEncoder* dan menstandarkan data dengan *StandardScaler*. Untuk menangani ketidakseimbangan kelas, digunakan metode *SMOTE* (*Synthetic Minority Oversampling Technique*) sehingga distribusi data menjadi proporsional. Dataset kemudian dibagi menjadi 80% data latih dan 20% data uji menggunakan *train\_test\_split()* dengan parameter *stratify* agar proporsi label tetap seimbang. Proses pelatihan model dilakukan menggunakan dua algoritma, yaitu *K-Nearest Neighbor* dan *Decision Tree*. Model *K-Nearest Neighbor* diuji menggunakan *GridSearchCV* dengan nilai  $k = \{3, 5, 7, 9\}$ , jarak *Manhattan* ( $p=1$ ), dan *weights = distance*. Sementara itu, model *Decision Tree* dioptimalkan dengan *hyperparameter tuning* pada *max\_depth*,

*min\_samples\_split*, dan *min\_samples\_leaf* menggunakan *3-fold cross-validation* untuk memperoleh performa terbaik.

Proses Evaluasi dilakukan dengan memanfaatkan *Confusion Matrix* untuk menghitung metrik akurasi, presisi, *recall*, dan *F1-score*, sehingga memperoleh gambaran menyeluruh mengenai kemampuan klasifikasi masing-masing algoritma.

### 3. HASIL DAN PEMBAHASAN

Pada bagian ini menyajikan hasil serta pembahasan dari penelitian klasifikasi tingkat obesitas menggunakan algoritma *Decision Tree* dan *K-Nearest Neighbor (KNN)*. Seluruh proses disusun mengikuti tahapan penelitian sebagaimana ditampilkan pada diagram alur, mulai dari pengumpulan data hingga evaluasi model setelah optimasi.

#### 3.1 Pengumpulan dan Seleksi Data

Penelitian dimulai dengan pengumpulan dataset obesitas yang berisi variabel demografis, kebiasaan makan, serta aktivitas fisik. Pada tahap seleksi awal, data dicek untuk memastikan kesesuaian struktur, format kolom, dan kelengkapan tipe fitur. Kolom numerik seperti *Age*, *Height*, *Weight*, FCVC, NCP, CH2O, FAF, dan TUE dipastikan berisi nilai valid dan dapat diproses pada tahap selanjutnya.

Tabel 2. Dataset

Age	Gender	Height	Weight	CALC	FAVC	FCVC	NPC	SCC	SMOKE	CH20	FHWO	FAF	TUE
21	Female	1.62	64	No	No	2	3	No	No	2	Yes	0	1
21	Female	1.52	56	Sometimes	No	3	3	Yes	Yes	3	Yes	3	0
23	Female	1.8	77	Frequently	No	2	3	No	No	2	Yes	2	1
27	Male	1.8	87	Frequently	No	3	3	No	No	2	No	2	0
22	Male	1.78	89.8	Sometimes	No	2	1	No	No	2	No	0	0

#### 3.2 Penanganan Missing Value dan Duplikasi

Dataset kemudian diperiksa untuk mendeteksi missing value dan duplikasi data. Seluruh nilai kosong pada fitur numerik diimputasi menggunakan median, agar distribusi data tidak berubah signifikan. Baris duplikat ditemukan dan dihapus untuk menghindari bias pada proses pelatihan model.

Tabel 3. Dataset setelah penanganan missing value dan duplikasi data

Age	Gender	Height	Weight	CALC	FAVC	FCVC	NPC	SCC	SMOKE	CH20	FHWO	FAF	TUE
21	Female	1.62	64.0	No	No	2.0	3.0	No	No	2.0	Yes	0.0	1.0
21	Female	1.52	56.0	Sometimes	No	3.0	3.0	Yes	Yes	3.0	Yes	3.0	0.0
23	Male	1.80	77.0	Frequently	No	2.0	3.0	No	No	2.0	Yes	2.0	1.0
27	Male	1.80	87.0	Frequently	No	3.0	3.0	No	No	2.0	No	2.0	0.0
22	Male	1.78	89.8	Sometimes	No	2.0	1.0	No	No	2.0	No	0.0	0.0

#### 3.3 Penghapusan Outlier

Analisis outlier dilakukan menggunakan *metode Interquartile Range (IQR)*. Nilai-nilai *ekstrem* yang berada jauh di luar rentang normal kemudian dihapus agar model tidak belajar dari pola yang tidak representatif. Langkah ini bertujuan mengurangi nilai ekstrem yang berpotensi mengganggu proses pelatihan model. Cuplikan kode berikut menunjukkan proses perhitungan IQR serta jumlah data sebelum dan sesudah penghapusan outlier.

```
print("Jumlah baris sebelum penghapusan outlier:", df.shape[0])
for col in num_cols:
    Q1, Q3 = df[col].quantile([0.25, 0.75])
    IQR = Q3 - Q1
    lower, upper = Q1 - 1.5*IQR, Q3 + 1.5*IQR
    df = df[(df[col] >= lower) & (df[col] <= upper)]
print("Jumlah baris setelah penghapusan outlier:", df.shape[0])
```

Berdasarkan hasil eksekusi kode tersebut, diperoleh bahwa jumlah data sebelum penghapusan outlier adalah 2.092 baris, dan berkurang menjadi 1.316 baris setelah proses penghapusan. menunjukkan bahwa 776 baris data ekstrem dihilangkan guna meningkatkan kualitas dataset. Ringkasan hasil ditampilkan pada Tabel 4 berikut.

Tabel 4. Distribusi data outlier

Keterangan	Jumlah Data
Sebelum penghapusan outlier	2.092
Setelah penghapusan outlier	1.316

### 3.4 Pembagian Dataset (Train–Test Split)

Dataset bersih kemudian dibagi menjadi 80% data latih dan 20% data uji menggunakan *teknik stratified sampling*. Teknik ini memastikan bahwa distribusi kelas target tetap seimbang pada kedua subset, sehingga evaluasi lebih representatif.

### 3.5 Standardisasi Data (StandardScaler)

Pada tahap selanjutnya, dilakukan proses *standarisasi* menggunakan *StandardScaler*, terutama untuk meningkatkan akurasi model KNN yang sangat bergantung pada skala jarak antar data. Setiap fitur numerik ditransformasi menjadi nilai dengan mean = 0 dan standar deviasi = 1, sehingga algoritma *machine learning* dapat beroperasi secara optimal tanpa bias akibat perbedaan skala antar fitur.

### 3.6 Penyeimbangan Data Menggunakan SMOTE

Distribusi kelas pada variabel *NObeyesdad* menunjukkan ketidakseimbangan yang cukup signifikan, di mana beberapa kategori memiliki jumlah sampel yang secara signifikan lebih sedikit dibandingkan kelas lain. Kondisi ini dapat menyebabkan model lebih berat sebelah terhadap kelas dengan data terbanyak. Untuk mengurangi ketidakseimbangan tersebut, digunakan teknik *SMOTE* (*Synthetic Minority Oversampling Technique*) yang menghasilkan data sintetis pada kelas minoritas sehingga seluruh kategori memiliki jumlah yang seimbang. Cuplikan kode berikut menunjukkan proses penerapan *SMOTE* dan pemeriksaan distribusi kelas sebelum dan sesudah penyeimbangan data:

```
# Pisahkan fitur dan label
X = df.drop('NObeyesdad', axis=1)
y = df['NObeyesdad']
# Terapkan SMOTE
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
# Cek distribusi label sebelum dan sesudah SMOTE
print("Distribusi label sebelum SMOTE:\n", y.value_counts())
print("\nDistribusi label sesudah SMOTE:\n", y_resampled.value_counts())
```

Berdasarkan hasil eksekusi kode tersebut, seluruh kategori berhasil diseimbangkan menjadi 314 data per kelas. Ringkasan distribusi kelas sebelum dan sesudah SMOTE ditampilkan pada Tabel 5.

Tabel 5. Distribusi kelas Nobeyesdad sebelum dan sesudah SMOTE

Kategori Obesitas	Sebelum SMOTE	Sesudah SMOTE
Obesity_Type_III	314	314
Obesity_Type_II	202	314
Normal_Weight	189	314
Obesity_Type_I	188	314
Overweight_Level_II	157	314
Overweight_Level_I	139	314
Insufficient_Weight	127	314

Penyeimbangan ini memastikan bahwa proses pelatihan model tidak bias terhadap kelas tertentu dan meningkatkan kemampuan model dalam mengenali seluruh kategori obesitas secara merata.

### 3.7 Penyeimbangan Data Menggunakan SMOTE

Pada tahap ini dibangun dua model klasifikasi untuk menganalisis pola pada dataset, yaitu *Decision Tree* dan *K-Nearest Neighbor (KNN)*. Model *Decision Tree* digunakan karena mampu mempelajari pola *non-linear* dan menghasilkan struktur keputusan yang mudah dipahami. Model ini membagi data berdasarkan fitur yang memberikan pemisahan terbaik, sehingga memudahkan interpretasi dan identifikasi fitur penting. Setelah *preprocessing* dan penyeimbangan data dengan *SMOTE*, model dilatih dan dievaluasi menggunakan akurasi, *precision*, *recall*, dan *confusion matrix*. Sedangkan model KNN dipilih karena bekerja efektif pada data yang telah distandarisasi, terutama setelah penyeimbangan kelas dengan SMOTE. Algoritma ini menentukan kelas berdasarkan kedekatan jarak dengan sejumlah tetangga terdekat. Beberapa nilai K dicoba untuk memperoleh

performa terbaik. Evaluasi dilakukan menggunakan metrik klasifikasi yang sama untuk membandingkan hasilnya dengan Decision Tree.

### 3.8 Parameter Tuning (GridSearchCV)

Untuk memperoleh performa model yang lebih maksimal, dilakukan proses *hyperparameter tuning* menggunakan *GridSearchCV* dengan skema *3-fold cross-validation*. Pendekatan ini memastikan setiap kombinasi parameter diuji secara menyeluruh pada tiga bagian data yang berbeda, sehingga hasil yang diperoleh lebih stabil dan tidak bergantung pada satu pembagian data saja.

Proses *tuning* pada *K-Nearest Neighbor (KNN)* difokuskan pada pencarian jumlah tetangga terbaik, jenis pembobotan jarak, serta tipe *metrik* yang digunakan untuk menghitung jarak antar titik data. Parameter yang diuji meliputi: *n\_neighbors* dengan nilai {3, 5, 7, 9} untuk menentukan jumlah tetangga terdekat, *weights* dengan opsi *uniform* dan *distance* untuk melihat perbedaan antara bobot yang sama dengan bobot berdasarkan jarak, *p* dengan nilai {1, 2} yang mewakili metrik jarak *Manhattan* (*p* = 1) dan *Euclidean* (*p* = 2).

Cuplikan kode berikut memperlihatkan proses pengujian nilai **k** serta perhitungan akurasi pada setiap konfigurasi:

```
# Daftar nilai k yang akan diuji
k_values = [3, 5, 7, 9]
# List untuk menyimpan hasil
results = []
for k in k_values:
    # Buat model dengan nilai k tertentu
    knn = KNeighborsClassifier(n_neighbors=k)
    # Latih model
    knn.fit(X_train, y_train)
    # Prediksi data test
    y_pred = knn.predict(X_test)
    # Hitung akurasi
    acc = accuracy_score(y_test, y_pred)
    # Simpan hasil
    results.append({'k': k, 'accuracy': acc})
# Tampilkan hasil
import pandas as pd
df_results = pd.DataFrame(results)
print(df_results)
```

Hasil pengujian akurasi dari masing-masing nilai K dapat dilihat pada Tabel 6 berikut

Tabel 6. Hasil pengujian akurasi model KNN dengan Variasi nilai K

Nilai K	Accuracy
3	0.865909
5	0.847727
7	0.831818
9	0.815909

Berdasarkan hasil yang tercantum pada Tabel 4, dapat dilihat bahwa hasil nilai K=3 memberikan akurasi tertinggi sebesar 86.59%, sehingga dipilih sebagai parameter terbaik untuk model *K-Nearest Neighbor*.

Pada *Tuning Decision Tree*, *tuning* dilakukan untuk mengatur kompleksitas pohon agar mampu belajar pola data dengan baik tanpa mengalami *overfitting*. Parameter yang diseleksi yaitu: *max\_depth*, yang mengatur seberapa dalam pohon keputusan boleh berkembang, *min\_samples\_split*, yang menentukan jumlah minimum sampel pada node sebelum dilakukan pemisahan, *min\_samples\_leaf*, yaitu jumlah minimum sampel yang harus terdapat pada node daun. Melalui proses pencarian menggunakan *GridSearchCV*, diperoleh konfigurasi parameter terbaik, yaitu: *max\_depth* = 10, *min\_samples\_split* = 5, *min\_samples\_leaf* = 1. Kombinasi parameter ini memberikan keseimbangan antara kedalaman pohon dan kemampuan generalisasi, sehingga model dapat mempelajari pola data secara optimal tanpa menghasilkan struktur yang terlalu kompleks.

### 3.9 Evaluasi Model

Setelah mendapatkan model terbaik dari hasil tuning kedua algoritma, dilakukan proses evaluasi ulang untuk melakukan perbandingan terhadap kinerja setiap model. Pengujian dilakukan menggunakan *metrik accuracy*, *precision*, *recall*, dan *F1-score*, yang menunjukkan bahwa baik *Decision Tree* maupun *K-Nearest Neighbor* mengalami peningkatan kinerja setelah dilakukan *hyperparameter tuning*. Hasil evaluasi memperlihatkan bahwa kedua model dapat menghasilkan prediksi yang lebih konsisten dan memiliki tingkat akurasi yang tinggi dalam mengklasifikasikan kategori obesitas. Secara keseluruhan, proses optimasi *hyperparameter* melalui *GridSearchCV*

terbukti mampu meningkatkan kualitas prediksi dan memperkuat konsistensi hasil klasifikasi pada dataset obesitas. Hasil evaluasi disusun dalam bentuk tabel perbandingan sebagai berikut:

```
columns = ["Accuracy", "Precision", "Recall", "F1 Score"]
df_results = pd.DataFrame([
    ["Decision Tree (Before)"] + dt_before, ["KNN (Before)"] + knn_before,
    ["Decision Tree (Tuned)"] + dt_after, ["KNN (Tuned)"] + knn_after
], columns=["Model"] + columns)
df_results
```

Hasil evaluasi dapat dilihat pada Tabel 7 berikut:

Tabel 7. Classification report

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	94%	0.94	0.94	0.94
KNN	84%	0.84	0.84	0.84
Decision Tree(Tuned)	95%	0.95	0.95	0.95
KNN(Tuned)	91%	0.91	0.91	0.91

Berdasarkan Tabel 5, *classification report* menunjukkan perbandingan hasil performa model sebelum dan sesudah dilakukan *hyperparameter tuning* pada algoritma *Decision Tree* dan *K-Nearest Neighbors* dalam memprediksi kategori obesitas. Model *Decision Tree* sebelum *tuning* memiliki akurasi sebesar 95%, dengan nilai *precision*, *recall*, dan *F1-score* masing-masing 0. 952055, 0. 952273, dan 0. 951995. Setelah dilakukan *tuning*, performanya meningkat menjadi 95%, dengan nilai metrik lainnya juga naik menjadi 0. 951950, 0. 952273, dan 0. 951941, yang menunjukkan model menjadi lebih akurat dan stabil.

Sementara itu, model *K-Nearest Neighbor* sebelum *tuning* memiliki akurasi 84%, dan nilai *precision*, *recall*, serta *F1-score* sebesar 0. 847179, 0. 847727, dan 0. 845862. Setelah dilakukan *tuning*, performanya juga meningkat menjadi 91%, dengan *precision* 0. 914282, *recall* 0. 913636, dan *F1-score* 0. 913372. Secara keseluruhan, hasil ini menunjukkan bahwa proses *hyperparameter tuning* mampu meningkatkan kinerja kedua model, membuat hasil klasifikasi menjadi lebih akurat dan konsisten dalam mengenali kategori obesitas.

### 3.10 Analisis Confusion Matrix

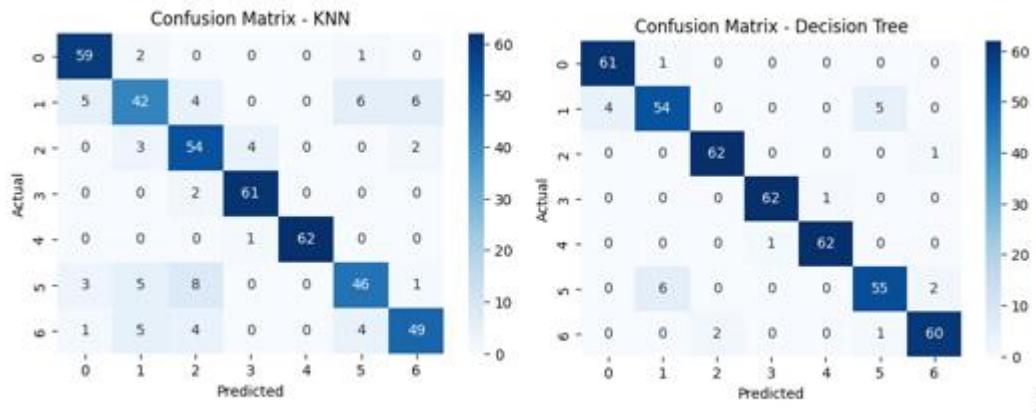
*Confusion matrix* merupakan alat evaluasi penting dalam klasifikasi karena menampilkan jumlah prediksi benar dan salah untuk setiap kelas secara terperinci. Matriks ini terdiri atas empat komponen utama: *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Dengan melihat distribusi nilai-nilai ini, performa model dapat dianalisis lebih mendalam, tidak hanya berdasarkan akurasi saja, tetapi juga kemampuan model membedakan kelas yang saling berdekatan.

Dalam konteks klasifikasi tingkat obesitas, *confusion matrix* membantu mengidentifikasi kelas mana saja yang sering salah diprediksi, misalnya kelas dengan karakteristik yang mirip sehingga rawan tertukar. Melalui visualisasi heatmap, pola kesalahan dan keberhasilan model dapat diamati dengan jelas, sehingga mempermudah interpretasi hasil klasifikasi. Berikut bagian kode pada Google Colab yang digunakan untuk menghasilkan *confusion matrix* pada setiap model klasifikasi yang diuji:

```
model_names = [ 'Decision Tree', 'KNN' ]
models = [dt, knn]
results = []
for model, name in zip(models, model_names):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, average='weighted', zero_division=0)
    rec = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    results.append([name, acc, prec, rec, f1])
# Confusion Matrix
plt.figure(figsize=(6, 4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title(f'Confusion Matrix - {name}')
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

```
plt.show()
```

Beedasarkan hasil kode tersebut menghasilkan visualisasi confusion matrix untuk masing-masing model, sebagaimana ditunjukkan pada Gambar 2. Melalui tampilan tersebut, distribusi prediksi benar dan salah dapat diamati secara lebih rinci pada setiap kelas kategori obesitas. Visualisasi ini juga membantu dalam memahami sejauh mana kemampuan kedua algoritma dalam membedakan kelas yang memiliki karakteristik serupa.



**Gambar 2. Confusion matrix**

Gambar 2 memperlihatkan *confusion matrix* dari hasil klasifikasi tingkat obesitas menggunakan dua model, yaitu *K-Nearest Neighbor (KNN)* dan *Decision Tree*. Secara umum, model *Decision Tree* memperlihatkan hasil yang lebih baik dibandingkan KNN, dengan jumlah prediksi benar yang lebih banyak di hampir seluruh kelas. Terlihat bahwa sebagian besar data diklasifikasikan dengan tepat pada diagonal utama, menandakan kemampuan model dalam mengenali dan mengklasifikasi pola kategori obesitas secara akurat. Meskipun begitu, masih terdapat beberapa kesalahan klasifikasi pada kelas tertentu, seperti pada kelas 5 dan 6, di mana muncul nilai *False Positive (FP)* dan *False Negative (FN)* yang menandakan adanya sedikit ketidaktepatan prediksi pada kategori yang berdekatan.

Sementara itu, model *K-Nearest Neighbor* menunjukkan kinerja yang sedikit lebih rendah. Beberapa kelas mengalami peningkatan jumlah kesalahan prediksi, terutama pada kategori 5 dan 6, di mana model sulit membedakan individu dengan tingkat obesitas yang mirip. Hal ini bisa disebabkan oleh sensitivitas *K-Nearest Neighbor* terhadap jarak antar data dan pengaruh parameter *k*, yang meskipun telah dioptimalkan, tetap memiliki keterbatasan dalam menangkap variasi kompleks pada data.

Secara umum, hasil pada gambar tersebut menunjukkan bahwa *Decision Tree* memiliki kestabilan dan tingkat akurasi yang lebih unggul dibandingkan *K-Nearest Neighbor*, meskipun keduanya masih menunjukkan sedikit kesalahan pada kelas tertentu.

### 3.11 Pembahasan Akhir

Hasil penelitian ini mengindikasi bahwa kombinasi algoritma *Decision Tree* dan *K-Nearest Neighbor* dengan tahapan *preprosesing* yang sistematis serta optimasi parameter yang tepat sehingga mampu menghasilkan kinerja klasifikasi obesitas yang akurat dan stabil. Sebelum pelatihan model, data melalui tahap *pre-processing* meliputi penghapusan *duplicasi*, menangani *missing value* menggunakan median, penghapusan *outlier* dengan metode *Interquartile Range (IQR)*, serta standarisasi fitur menggunakan *StandardScaler*. Selain itu, ketidakseimbangan kelas diatasi menggunakan *Synthetic Minority Oversampling Technique (SMOTE)* agar distribusi data lebih proporsional. Langkah-langkah ini terbukti meningkatkan stabilitas model, khususnya pada *K-Nearest Neighbor* yang sensitif terhadap perbedaan skala data.

Optimasi parameter menggunakan *GridSearchCV* menghasilkan konfigurasi terbaik pada KNN dengan *k=3*, jarak Manhattan (*p=1*), dan *weights = distance*, sedangkan *Decision Tree* optimal pada *max\_depth=10*, *min\_samples\_split=2*, dan *min\_samples\_leaf=1*. Berdasarkan hasil pengujian, model *Decision Tree* menunjukkan peningkatan kinerja yang cukup signifikan setelah melalui proses *hyperparameter tuning*. Akurasi awal sebesar 94% meningkat menjadi 95%, disertai peningkatan pada nilai *precision* dan *recall* masing-masing menjadi 0.94. Sementara itu, model *K-Nearest Neighbor* juga mengalami perbaikan performa, dengan akurasi meningkat dari 84% menjadi 91% setelah dilakukan optimasi parameter. Meskipun peningkatan pada *K-Nearest Neighbor* tidak sebesar *Decision Tree*, hasil tersebut menunjukkan bahwa penyesuaian parameter berperan penting dalam menyeimbangkan bias dan variansi, serta meningkatkan konsistensi hasil prediksi pada kedua model.

Visualisasi melalui *Confusion Matrix* pada Gambar 2 mengonfirmasi hasil tersebut. Model *Decision Tree* mampu memberikan prediksi yang lebih akurat pada sebagian besar kelas obesitas, terutama pada kelas 1 hingga 4.

Kesalahan klasifikasi (*False Positive* dan *False Negative*) masih muncul pada kelas 5 dan 6, namun jumlahnya jauh lebih sedikit dibandingkan sebelum tuning. Di sisi lain, *K-Nearest Neighbor* menunjukkan kecenderungan kesalahan pada kelas yang berdekatan karena algoritma ini sangat bergantung pada jarak antar data. Meski demikian, setelah dilakukan standarisasi dan penyesuaian parameter, jumlah kesalahan tersebut berhasil ditekan secara signifikan, menandakan bahwa *preprocessing* dan optimasi berperan besar dalam memperbaiki kinerja model.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa *Decision Tree* memiliki performa yang lebih unggul dibandingkan *K-Nearest Neighbor* dalam mengklasifikasikan tingkat obesitas, baik dari segi akurasi maupun stabilitas prediksi. Namun, kedua model menunjukkan peningkatan nyata setelah melalui tahap *preprocessing*, *feature scaling*, dan *hyperparameter tuning*. Hal ini membuktikan bahwa kombinasi langkah-langkah tersebut efektif dalam meningkatkan kualitas pembelajaran mesin serta menghasilkan model klasifikasi obesitas yang lebih akurat, konsisten, dan andal untuk diterapkan pada data dengan karakteristik serupa.

Dibandingkan penelitian sebelumnya yang hanya menggunakan *Decision Tree* atau *K-Nearest Neighbor* secara terpisah[20][21], penelitian ini menggabungkan tahapan *preprocessing*, penyeimbangan data, dan optimasi *hyperparameter* untuk meningkatkan hasil klasifikasi. Pendekatan tersebut terbukti mampu memperbaiki akurasi dan kestabilan kedua model dalam mengenali berbagai kategori obesitas.

Secara keseluruhan, kombinasi teknik *preprocessing*, standarisasi fitur, dan *tuning parameter* menjadikan *Decision Tree* dan *K-Nearest Neighbor* bekerja lebih optimal serta menghasilkan prediksi yang lebih konsisten. Meski demikian, penelitian ini masih terbatas pada satu dataset publik. Untuk penelitian selanjutnya, disarankan untuk menguji generalisasi model pada dataset yang lebih beragam atau menerapkan metode stacking ensemble dan deep learning agar hasil prediksi lebih adaptif dan akurat terhadap variasi data yang kompleks.

## 4. KESIMPULAN

Berdasarkan hasil analisis dan uraian pembahasan, dapat disimpulkan bahwa algoritma *Decision Tree (DT)* memiliki performa yang lebih unggul dibandingkan dengan algoritma *K-Nearest Neighbor (KNN)* dalam melakukan klasifikasi tingkat obesitas berdasarkan data terkait gaya hidup dan kondisi fisik individu. Tahapan *preprocessing* yang meliputi penghapusan data *duplicasi*, penanganan *missing value* menggunakan median, penghapusan *outlier* dengan menggunakan metode (*Interquartile Range*) *IQR*, standarisasi fitur dengan *StandardScaler*, serta penyeimbangan data menggunakan *SMOTE* terbukti berpengaruh besar untuk membantu meningkatkan kualitas data dan akurasi pada model. Setelah dilakukan *hyperparameter tuning* menggunakan *GridSearchCV*, akurasi algoritma *Decision Tree* meningkat dari 94% menjadi 95%, sedangkan algoritma *K-Nearest Neighbor* naik dari 84% menjadi 91%. Selain akurasi, nilai *precision*, *recall*, dan *F1-score* juga ikut membaik. Hasil ini menunjukkan bahwa proses optimasi parameter berpengaruh terhadap kestabilan model dan hasil prediksi. Secara keseluruhan, *Decision Tree* lebih mampu mengenali hubungan kompleks antarvariabel dibanding *K-Nearest Neighbor* yang lebih sederhana namun tetap cukup baik sebagai pembanding. Keterbatasan dalam penelitian ini adalah hanya penggunaan satu dataset dengan jumlah sampel yang terbatas. Penelitian selanjutnya disarankan untuk memanfaatkan dataset dengan ukuran yang lebih besar serta mencoba menerapkan metode lain yang mampu menangkap pola data yang lebih kompleks, sehingga hasil klasifikasi dapat menjadi lebih akurat dan memiliki penerapan yang lebih luas.

## REFERENCES

- [1] G. P. Maraya, “Analisis Hubungan Penggunaan Layanan Pesan Antar Makanan Online Dengan Perubahan Pola Makan Generasi Z Di Jakarta Saat Pandemi Covid-19,” *Univ. Binawan*, pp. 18–23, 2022.
- [2] R. Suraya, A. S. V. Nababan, A. Siagian, and Z. Lubis, “Pengaruh Konsumsi Makanan Jajanan, Aktivitas Fisik, Screen Time, dan Durasi Tidur Terhadap Obesitas Pada Remaja,” *J. Dunia Gizi*, vol. 3, no. 2, pp. 80–87, 2021, doi: 10.33085/jdg.v3i2.4732.
- [3] T. Kristiana, D. Hermawan, U. Febriani, and A. Farich, “Hubungan Antara Pola Tidur Dan Kebiasaan Makan Junk Food Dengan Kejadian Obesitas Pada Mahasiswa Universitas Malahayati Tahun 2019,” *Hum. Care J.*, vol. 5, no. 3, p. 750, 2020, doi: 10.32883/hcj.v5i3.758.
- [4] S. Suharsono, S. Suyanta, A. Sugiyarto, Y. Yulistanti, and L. Handayani, “Upaya Pencegahan Penyakit Tidak Menular Melalui Penyuluhan Dan Deteksi Dini,” *BESIRU J. Pengabd. Masy.*, vol. 2, no. 6, pp. 530–541, 2025, doi: 10.62335/besiru.v2i6.1343.
- [5] N. Rachmawati and D. W. Minarsih, “Deteksi Dini dan Edukasi Obesitas Sebagai Upaya Pencegahan Diabetes Mellitus,” vol. I, no. April, pp. 48–56, 2025.
- [6] F. Diba, “Makanan Ultra-prosesed, Inovasi dalam Industri Makanan Modern Ultra-processed Foods, Inovation in the Modern Food Industry,” *J. Kedokt. dan Kesehatan-Fakultas Kedokt. Univ. Islam Sumatera Utara*, vol. 24, no. 1, pp. 191–201, 2025, [Online]. Available: <https://www.jurnal.fk.uisu.ac.id/index.php/ibnusina/article/view/798>
- [7] S. Riskia, “Faktor Resiko Penyebab Obesitas Pada Remaja : Literature Review,” *J. Kesehat. bertuah Indones.*, vol. 1, no. 2, pp. 53–64, 2024.
- [8] Nourmayansa Vidya, Ritanti, and D. Ratnawati, “Genetik Mempengaruhi Kejadian Obesitas Pada Anak Usia Sekolah,” *J. Heal. Educ. Lit.*, vol. 5, no. 1, pp. 1–7, 2022, doi: 10.31605/j-healt.v5i1.1495.

- [9] R. G. Wardhana, G. Wang, and F. Sibuea, "Penerapan Machine Learning Dalam Prediksi Tingkat Kasus Penyakit Di Indonesia," *J. Inf. Syst. Manag.*, vol. 5, no. 1, pp. 40–45, 2023, doi: 10.24076/joism.2023v5i1.1136.
- [10] S. M. K. Yahya, *Data Mining*. CV Jejak, anggota IKAPI, 2022.
- [11] Mellyia P & Erlin, "Prediksi Penyakit Stroke Menggunakan Machine Learning Dengan Algoritma Random Forest," *J. Infomedia Tek. Inform.*, vol. 9, no. 2, 2024.
- [12] E. Hasibuan *et al.*, "Implementasi Machine Learning untuk Prediksi Harga Mobil Bekas dengan Algoritma Regresi Linear berbasis Web," *J. Ilm. Komputasi*, vol. 21, no. 4, pp. 595–602, 2022, doi: 10.32409/jikstik.21.4.3327.
- [13] G. S. W. Prabuningrat, D. P. Hostiadi, and N. L. P. Srinadi, "Klasifikasi Deteksi Anomali Menggunakan Metode Machine Learning," *Pros. Semin. Has. Penelit. Inform. dan Komput.*, vol. 1, no. 2, pp. 845–850, 2024.
- [14] Mufliah Ihza Rifatama, Mohammad Reza Faisal, Rudy Herteno, Irwan Budiman, and Muhammad Itqan Mazdadi, "Optimasi Algoritma K-Nearest Neighbor Dengan Seleksi Fitur Menggunakan Xgboost," *J. Inform. dan Rekayasa Elektron.*, vol. 6, no. 1, pp. 64–72, 2023, doi: 10.36595/jire.v6i1.723.
- [15] S. Zacksavira, C. A. Br Sebayang, J. Imanuel, and G. Br Sibarani, "Implementasi Algoritma Decision Tree untuk Deteksi Depresi Pada Pelajar," *Innov. J. Soc. Sci. Res.*, vol. 5, no. 3, pp. 5915–5934, 2025, doi: 10.31004/innovative.v5i3.19793.
- [16] D. Kiswanto, F. Ramadhani, N. M. Surbakti, and N. A. Nasution, "Pengembangan dan Implementasi Sistem Deteksi Serangan DDoS Berbasis Algoritma Random Forest," vol. 6, no. 3, 2025, doi: 10.47065/bit.v6i2.2203.
- [17] S. A. Utiarahman and A. M. M. Pratama, "Penerapan Support Vector Machine dan Random Forest Classifier Untuk Klasifikasi Tingkat Obesitas," *J. Fasilkom*, vol. 14, no. 3, pp. 754–760, 2024, doi: 10.37859/jf.v14i3.8104.
- [18] R. N. Prakoso, S. I. Rochim, A. Subarnas, and M. E. Kurniawan, "Perbandingan Algoritma Naïve Bayes Dan Random Forest Dalam Klasifikasi Obesitas Berdasarkan Faktor Gaya Hidup," *J. Inf. Eng. Educ. Technol.*, vol. 9, no. 1, pp. 11–18, 2025, doi: 10.26740/jeet.v9n1.p11-18.
- [19] S. Lonang, A. Yudhana, and M. K. Biddinika, "Analisis Komparatif Kinerja Algoritma Machine Learning untuk Deteksi Stunting," *J. Media Inform. Budidarma*, vol. 7, no. 4, p. 2109, 2023, doi: 10.30865/mib.v7i4.6553.
- [20] H. T. Santoso, F. A. Felmid, A. N. Fadila, A. Ristyawan, and E. Daniati, "Analisis Kinerja Algoritma Data Mining pada Klasifikasi Tingkat Obesitas dengan K-Fold Cross Validation dan AUC," *Agustus*, vol. 8, pp. 2549–7952, 2024.
- [21] S. Y. Sibi and A. R. Widiarti, "Klasifikasi Tingkat Obesitas Mempergunakan Algoritma KNN," *Semin. Nas. Corisindo*, vol. 7, no. 2, pp. 370–375, 2022.