

# ANALISIS PENGGUNAAN ALGORITMA *SUPPORT VECTOR MACHINE (SVM)* DAN *ARTIFICIAL NEURAL NETWORK (ANN)* PADA KLASIFIKASI TINGKAT OBESITAS

**Surya D. Kang**

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,  
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia  
Email: [surya.535220148@stu.untar.ac.id](mailto:surya.535220148@stu.untar.ac.id)

## ABSTRAK

Analisis ini bertujuan untuk mengeksplorasi berbagai faktor yang menyebabkan obesitas seperti konsumsi makanan, kebiasaan-kebiasaan, dan variabel lain yang mempengaruhi obesitas. Data yang digunakan adalah data estimasi tingkat obesitas dari negara Mexico, Peru dan Kolombia berdasarkan kebiasaan mereka. Data bersumber dari website bernama Kaggle. Data diolah menggunakan bahasa pemrograman Python, data dianalisa menggunakan algoritma *Support Vector Machine (SVM)* dan *Artificial Neural Network (ANN)*. Hasil analisa memberikan perbandingan kinerja antara kedua algoritma.

**Kata kunci**— Obesitas, akurasi, *SVM*, *ANN*

## ABSTRACT

*This analysis aims to explore various factors contributing to obesity such as food consumption, habits, and other variables influencing obesity. The data utilized are estimates of obesity rates from Mexico, Peru, and Colombia based on their respective habits. The data is sourced from a website called Kaggle. Python programming language was employed for data processing, and analysis was conducted using Support Vector Machine (SVM) and Artificial Neural Network (ANN) algorithms. The analysis results provide a performance comparison between the two algorithms.*

**Keywords**— *obesity, accuracy, SVM, ANN*

## 1. PENDAHULUAN

Obesitas merupakan kondisi medis ketika lemak yang menumpuk di tubuh melebihi batas normal, yang membuat berat badan juga melebihi batas normal. Kondisi ini dapat menimbulkan beberapa penyakit berbahaya seperti diabetes, kanker, penyakit jantung dan stroke. Faktor seperti kebiasaan hidup dan pola makan merupakan penyebab terbesar dari obesitas. Kondisi ini sangat memprihatinkan di kalangan remaja sampai dewasa. Karena remaja masa kini sangat memperdulikan bentuk fisiknya, tubuh yang kurus dan ramping merupakan bentuk tubuh yang ideal bagi mereka.

WHO pada tahun 2013 menyatakan bahwa setiap tahunnya sekitar 3,4 juta orang dewasa meninggal dunia karena obesitas. Sekitar 44% disebabkan oleh penyakit dari obesitas yaitu diabetes, 23% oleh penyakit jantung iskemik dan 7-41% oleh kanker. [1] Telah terjadi perubahan gaya hidup dari tahun ke tahun hal ini dapat dikaitkan dengan peningkatan kasus obesitas. Berdasarkan Riset Kesehatan Dasar (Riskesdas) 27,1% remaja usia 18 tahun tercatat mengalami

obesitas. Kondisi ini lebih banyak terjadi di perempuan daripada laki-laki dimana tingkat prevalensi pada perempuan mencapai 32.9% dan laki-laki 19.7%. [2]

Berdasarkan Informasi tersebut obesitas bukanlah hal yang dapat diabaikan lagi, masyarakat harus lebih sadar akan bahayanya kondisi ini. Karena masih banyak orang yang menyepelekan masalah obesitasnya dengan anggapan bahwa mereka belum pernah terkena ataupun melihat kejadian orang terkena penyakit akibat obesitas. Kenyataannya penyakit ini bisa saja muncul di usia tua mereka dan akan lebih susah untuk disembuhkan. Analisis ini bertujuan untuk melakukan perbandingan antara dua algoritma yaitu *Support Machine Vector (SVM)* dan *Aritificial Neural Network (ANN)*. Dengan melakukan analisis dan perbandingan kedua algoritma ini diharapkan dapat membantu memprediksi atau mencegah kondisi obesitas seseorang serta membantu pengembangan teknik klasifikasi *machine learning* menjadi lebih baik lagi.

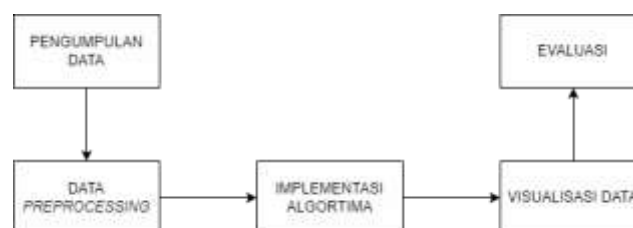
## 2. METODE PENELITIAN

Algoritma yang digunakan adalah *Support Vector Machine (SVM)* dan *Artificial Neural Network (ANN)*. SVM adalah algoritma yang menggunakan fungsi-fungsi linear di dalam fitur yang mempunyai dimensi tinggi [3]. Terdapat banyak algoritma dalam *machine learning* seperti *K-NN*, *Naive Bayes*, *Decision Tree* dan lain-lain namun SVM tetap menjadi algoritma terbaik dibandingkan dengan algoritma yang lain [4]. SVM terdiri dari beberapa *kernel* yaitu *kernel linear*, *kernel rbf* dan *kernel polinomial*. *Kernel* yang digunakan pada analisis ini adalah *kernel linear* yang seperti namanya memisahkan data secara linear dengan sebuah garis lurus tunggal. Berikut merupakan rumus dari *kernel linear*.

$$\text{Linear} \quad | \quad x_i^T x$$

Gambar 1 Rumus Korelasi Pearson [5]

Algoritma kedua adalah algoritma ANN, ANN adalah algoritma yang dibuat berdasarkan inspirasi dari sistem saraf otak manusia dalam klasifikasi data. Salah satu kelebihan dari algoritma ini adalah ANN mampu mendeteksi kemungkinan interaksi antar variabel prediksi. ANN juga mampu memvisualisasikan model linier ataupun non linier dengan jangkauan yang cukup luas [6]. Sistem dari algoritma ANN memiliki kesamaan dengan sistem kerja otak manusia yang terdiri dari neuron. Neuron merupakan kumpulan sel syaraf yang terhubung satu sama lain. Neuron dalam ANN memiliki tiga elemen yaitu fungsi penjumlahan, aktivasi, dan output [7]. Analisis dilakukan dengan melalui beberapa tahapan yang dapat dilihat pada Gambar 2.2.



Gambar 2 Tahapan Analisis

### 2.1 Pengumpulan Data

Data yang digunakan dalam analisis ini adalah data estimasi tingkat obesitas di negara Mexico, Peru dan Kolombia yang berisi data dari orang berumur 14 sampai 61 tahun. Data meliputi kebiasaan-kebiasan orang seperti pola makan dan aktifitas sehari-hari mereka. Data tersebut diambil dari website yang menyediakan dataset-dataset untuk dianalisa yaitu *Kaggle* (<https://www.kaggle.com/>). Dataset memiliki 17 kolom dan 2111 baris yang berisi *Gender*, *Age*,

*Height, Weight, family history with overweight (FHO), Frequent consumption of high caloric food (FAVC), Frequency of consumption of vegetables (FCVC), Number of main meals (NCP), Consumption of food between meals (CAEC), Smoke, Consumption of water daily (CH2O), Calories consumption monitoring (SCC), Physical activity frequency (FAF), Time using technology devices (TUE), Consumption of alcohol (CALC), Mtrans, NObesydad.*

| Index | Gender | Age  | Height | Weight | family_history_with_overweight | FAVC | FCVC | NCP | CAEC       | SMOKE | CHOD | SCC | FAF | TUE | CALC       | MTRANS                | NObesydad          |
|-------|--------|------|--------|--------|--------------------------------|------|------|-----|------------|-------|------|-----|-----|-----|------------|-----------------------|--------------------|
| 0     | Female | 21.0 | 1.62   | 64.0   | yes                            | no   | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 0.0 | 1.0 | no         | Public_Transportation | Normal_Weight      |
| 1     | Female | 21.0 | 1.62   | 66.0   | yes                            | no   | 3.0  | 1.0 | Sometimes  | yes   | 3.0  | yes | 3.0 | 0.0 | Sometimes  | Public_Transportation | Normal_Weight      |
| 2     | Male   | 23.0 | 1.8    | 77.0   | yes                            | no   | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 2.0 | 1.0 | Frequently | Public_Transportation | Normal_Weight      |
| 3     | Male   | 27.0 | 1.8    | 67.0   | no                             | no   | 3.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 3.0 | 0.0 | Frequently | Walking               | Overweight_Level_1 |
| 4     | Male   | 22.0 | 1.78   | 89.0   | no                             | no   | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 0.0 | 0.0 | Sometimes  | Public_Transportation | Overweight_Level_1 |
| 5     | Male   | 25.0 | 1.62   | 53.0   | no                             | yes  | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 0.0 | 0.0 | Sometimes  | Automobile            | Normal_Weight      |
| 6     | Female | 23.0 | 1.5    | 55.0   | yes                            | yes  | 3.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 1.0 | 0.0 | Sometimes  | Motorbike             | Normal_Weight      |
| 7     | Male   | 22.0 | 1.64   | 53.0   | no                             | no   | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 3.0 | 0.0 | Sometimes  | Public_Transportation | Normal_Weight      |
| 8     | Male   | 24.0 | 1.78   | 64.0   | yes                            | yes  | 3.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 1.0 | 1.0 | Frequently | Public_Transportation | Normal_Weight      |
| 9     | Male   | 22.0 | 1.72   | 66.0   | yes                            | yes  | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 1.0 | 1.0 | no         | Public_Transportation | Normal_Weight      |
| 10    | Male   | 20.0 | 1.88   | 100.0  | yes                            | yes  | 3.0  | 1.0 | Frequently | no    | 3.0  | no  | 2.0 | 2.0 | Sometimes  | Public_Transportation | Obesity_Type_1     |
| 11    | Female | 21.0 | 1.72   | 88.0   | yes                            | yes  | 2.0  | 1.0 | Frequently | no    | 2.0  | yes | 2.0 | 1.0 | Sometimes  | Public_Transportation | Overweight_Level_1 |
| 12    | Male   | 22.0 | 1.65   | 56.0   | no                             | no   | 3.0  | 1.0 | Sometimes  | no    | 3.0  | no  | 2.0 | 0.0 | Sometimes  | Public_Transportation | Normal_Weight      |
| 13    | Male   | 41.0 | 1.8    | 99.0   | no                             | yes  | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 2.0 | 1.0 | Frequently | Automobile            | Obesity_Type_1     |
| 14    | Male   | 23.0 | 1.77   | 68.0   | yes                            | yes  | 2.0  | 1.0 | Sometimes  | no    | 1.0  | no  | 1.0 | 1.0 | Sometimes  | Public_Transportation | Normal_Weight      |
| 15    | Female | 23.0 | 1.7    | 66.0   | yes                            | no   | 3.0  | 1.0 | Always     | no    | 2.0  | yes | 2.0 | 1.0 | Sometimes  | Public_Transportation | Normal_Weight      |
| 16    | Male   | 27.0 | 1.93   | 102.0  | yes                            | yes  | 2.0  | 1.0 | Sometimes  | no    | 1.0  | no  | 1.0 | 0.0 | Sometimes  | Public_Transportation | Overweight_Level_1 |
| 17    | Female | 20.0 | 1.53   | 70.0   | no                             | yes  | 2.0  | 1.0 | Sometimes  | no    | 2.0  | no  | 0.0 | 0.0 | no         | Automobile            | Obesity_Type_1     |

Gambar 3 Dataset “Obesity or CVD risk (Classify/Regressor/Cluster)”

## 2.2 Data Preprocessing

Pada tahap ini nama kolom-kolom yang masih panjang dipersingkat dengan melakukan *rename* pada setiap kolom. Data yang masih berjenis string juga akan dirubah menjadi numerik pada tahap ini menggunakan label encoder pada gambar di bawah.

```
#renaming columns
df.columns = ['Gender', 'Age', 'Height', 'Weight', 'FHO', 'FAVC', 'FCVC', 'NCP', 'CAEC', 'SMOKE', 'CHOD', 'SCC', 'FAF', 'TUE', 'CALC', 'MTRANS', 'Class']

columns = ['Gender', 'FHO', 'CAEC', 'SMOKE', 'SCC', 'CALC', 'MTRANS', 'Class']

label_encoder = LabelEncoder()

for column in df.columns:
    if df[column].dtype == 'object':
        df[column] = label_encoder.fit_transform(df[column])
```

Gambar 4 Kode untuk *rename* dan mengubah data *string* menjadi data numerik

|      | Gender | Age       | Height   | Weight     | FHO | FAVC | FCVC | NCP | CAEC | SMOKE | CHOD     | SCC | FAF      | TUE      | CALC | MTRANS | Class |
|------|--------|-----------|----------|------------|-----|------|------|-----|------|-------|----------|-----|----------|----------|------|--------|-------|
| 0    | 0      | 21.000000 | 1.620000 | 64.000000  | 1   | 0    | 2.0  | 1.0 | 2    | 0     | 2.000000 | 0   | 0.000000 | 1.000000 | 0    | 3      | 1     |
| 1    | 0      | 21.000000 | 1.620000 | 66.000000  | 1   | 0    | 3.0  | 1.0 | 2    | 1     | 3.000000 | 1   | 3.000000 | 0.000000 | 2    | 3      | 1     |
| 2    | 1      | 23.000000 | 1.800000 | 77.000000  | 1   | 0    | 2.0  | 1.0 | 2    | 0     | 2.000000 | 0   | 2.000000 | 1.000000 | 1    | 3      | 1     |
| 3    | 1      | 27.000000 | 1.800000 | 67.000000  | 0   | 0    | 3.0  | 1.0 | 2    | 0     | 2.000000 | 0   | 2.000000 | 0.000000 | 1    | 4      | 0     |
| 4    | 1      | 22.000000 | 1.780000 | 89.000000  | 0   | 0    | 2.0  | 1.0 | 2    | 0     | 2.000000 | 0   | 0.000000 | 0.000000 | 2    | 3      | 0     |
| ...  | ...    | ...       | ...      | ...        | ... | ...  | ...  | ... | ...  | ...   | ...      | ... | ...      | ...      | ...  | ...    | ...   |
| 2106 | 0      | 20.976842 | 1.710730 | 131.408538 | 1   | 1    | 3.0  | 1.0 | 2    | 0     | 1.728129 | 0   | 1.676269 | 0.906247 | 2    | 3      | 4     |
| 2107 | 0      | 21.982942 | 1.748984 | 133.742943 | 1   | 1    | 3.0  | 1.0 | 2    | 0     | 2.005130 | 0   | 1.341390 | 0.599270 | 2    | 3      | 4     |
| 2108 | 0      | 22.524036 | 1.752206 | 133.689352 | 1   | 1    | 3.0  | 1.0 | 2    | 0     | 2.054193 | 0   | 1.414209 | 0.646288 | 2    | 3      | 4     |
| 2109 | 0      | 24.361936 | 1.739450 | 133.346641 | 1   | 1    | 3.0  | 1.0 | 2    | 0     | 2.852339 | 0   | 1.139107 | 0.586035 | 2    | 3      | 4     |
| 2110 | 0      | 23.664709 | 1.738836 | 133.472641 | 1   | 1    | 3.0  | 1.0 | 2    | 0     | 2.863513 | 0   | 1.026452 | 0.714137 | 2    | 3      | 4     |

Gambar 5 Dataset “Obesity or CVD risk (Classify/Regressor/Cluster)” setelah encoding label

Tahap terakhir dari *data preprocessing* adalah memisahkan data latih dan data uji yang nantinya dilatih dan akan dilakukan prediksi menggunakan data tersebut.

## Implementasi Algoritma SVM dan ANN

Implementasi algoritma SVM dan ANN dilakukan menggunakan bahasa *python*. Pada analisis ini terdapat tiga skenario yang pertama data latih 90% dan data uji 10%, kedua data latih 80% dan data uji 20%, ketiga data latih 70% dan data uji 30%, keempat data latih 60% dan data uji 40%, kelima data latih 50% dan data uji 50%. Pada penggunaan algoritma SVM *kernel linear* nilai *C* yang digunakan adalah 10 yang merupakan nilai standarnya. Setelah data dipisah data akan dilatih menggunakan kode sebagai berikut.

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

from sklearn.metrics import accuracy_score, classification_report

accuracy_scores = []

#Melakukan loop 5x dengan random state 0 - 40 dengan data uji 20%
for i in range(0, 41, 10):
    # Pisahkan data latih dan uji dengan nilai random state yang berbeda di setiap iterasi
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=i)

    # Inisialisasi model SVM
    linear_svc = SVC(kernel='linear', C=10.0)

    # Latih model
    linear_svc.fit(X_train, y_train)

    # Lakukan prediksi pada data uji
    y_pred_test = linear_svc.predict(X_test)

    # Hitung akurasi model
    accuracy = accuracy_score(y_test, y_pred_test)
    accuracy_scores.append(accuracy)
```

**Gambar 6** Kode untuk pemisahan data dan pelatihan data pada algoritma SVM

Dengan menggunakan fungsi *loop* uji coba dapat dilakukan lebih cepat. Pada setiap iterasi random state yang digunakan akan bertambah 10 dari 0 sampai 40. Hal yang sama juga dilakukan di algoritma ANN dimana perbedaannya berada di cara melatih datanya saja, berikut merupakan kode untuk melatih data pada ANN.

```
from sklearn.metrics import accuracy_score, classification_report

accuracy_scores = []

#Melakukan loop 10x dengan random state 0 - 40 dengan data uji 20%
for i in range(0, 41, 10):
    # Pisahkan data latih dan uji dengan nilai random state yang berbeda di setiap iterasi
    X_train, X_test, y_train, y_test = train_test_split(fitur, kelas, test_size = 0.20, random_state = i)

    # Train MLP classifier
    clf = MLPClassifier(random_state=5, max_iter=100)
    clf.fit(X_train, y_train)

    # Predict on test set
    y_predict_ann = clf.predict(X_test)
    y_predict_ann
```

**Gambar 7** Kode untuk pemisahan data dan pelatihan data pada algoritma ANN

### 2.3 Visualisasi Data

Pada tahap ini hasil implementasi akan ditampilkan menggunakan laporan klasifikasi yang akan dipindahkan ke tabel secara detail dan akan dirata-ratakan untuk membandingkan akurasi dari kedua algoritma tersebut. Kode yang digunakan untuk menampilkan laporan klasifikasi adalah sebagai berikut.

```
# Tampilkan hasil akurasi untuk setiap percobaan
print(f"Iterasi dengan random state {i}:")
print("Akurasi:", accuracy)
print()

# Tampilkan classification report
report = classification_report(y_test, y_pred_test)
print("Classification Report:")
print(report)
print()
```

**Gambar 8** Kode untuk memvisualisasikan hasil implementasi algoritma SVM

```
# Tampilkan classification report
report = classification_report(y_test, y_predict_ann)
print("Classification Report:")
print(report)
print()
```

**Gambar 9** Kode untuk memvisualisasikan hasil implementasi algoritma ANN

### 2.3 Evaluasi

Evaluasi dilakukan dengan membandingkan nilai yang ada pada laporan klasifikasi. Atribut yang dibandingkan antara lain adalah *precision*, *recall*, *F1-Score*, dan akurasi setiap atribut akan dirata-ratakan terlebih dahulu. Lalu akan dilakukan pencarian untuk menentukan algoritma mana yang lebih baik.

## 3. HASIL DAN PEMBAHASAN

Pada dataset obesitas yang dapat dilihat pada gambar 5 terdapat kolom *class* yang berisi 1 sampai 6 yang dapat artikan menjadi berat normal, obesitas tingkat satu, obesitas tingkat dua, dan seterusnya. Kolom *class* ini ditentukan berdasarkan kolom-kolom lainnya seperti jumlah makanan utama setiap hari, sayur-sayuran yang dimakan, aktivitas olahraga dan lainnya. Secara logika berpikir orang dengan tingkat aktivitas olahraga rendah dan dengan pola makan tidak terjaga memiliki kemungkinan besar mengalami obesitas namun pada dataset ini masih ada yang jarang bergerak dan gaya hidupnya juga banyak makan masuk ke kelas berat normal. Hal ini dapat terjadi karena adanya perbedaan makanan karena diet merupakan kunci utama dari berat badan yang ideal dan tidak mengalami obesitas.

### 3.1 Hasil Analisa Menggunakan Algoritma SVM

Analisa dilakukan lima jenis skenario dengan jumlah data uji yang berbeda-beda dan analisa dilakukan sebanyak lima kali dengan nilai  $C=10$  dan *random state* yang selalu bertambah 10 di setiap iterasinya mulai dari angka 0. Setiap skenario akan dilakukan percobaan sebanyak lima kali yang akan dihitung rata-ratanya untuk perbandingan.

**Tabel 1** Hasil analisa menggunakan algoritma SVM dengan data latih 90% dan data uji 10%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.99      | 0.98   | 0.98     | 0.99    |
| 2             | 0.94      | 0.95   | 0.94     | 0.95    |
| 3             | 0.97      | 0.97   | 0.97     | 0.97    |
| 4             | 0.94      | 0.94   | 0.94     | 0.94    |
| 5             | 0.98      | 0.98   | 0.98     | 0.98    |
| Rata-rata     | 0.96      | 0.96   | 0.96     | 0.97    |

**Tabel 2** Hasil analisa menggunakan algoritma SVM dengan data latih 80% dan data uji 20%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.97      | 0.97   | 0.97     | 0.97    |
| 2             | 0.94      | 0.94   | 0.94     | 0.94    |
| 3             | 0.95      | 0.95   | 0.95     | 0.95    |
| 4             | 0.95      | 0.95   | 0.95     | 0.95    |
| 5             | 0.96      | 0.96   | 0.96     | 0.96    |
| Rata-rata     | 0.95      | 0.95   | 0.95     | 0.95    |

**Tabel 3** Hasil analisa menggunakan algoritma SVM dengan data latih 70% dan data uji 30%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.95      | 0.95   | 0.95     | 0.95    |
| 2             | 0.93      | 0.93   | 0.93     | 0.93    |
| 3             | 0.95      | 0.95   | 0.95     | 0.95    |
| 4             | 0.96      | 0.96   | 0.96     | 0.96    |
| 5             | 0.96      | 0.96   | 0.96     | 0.96    |
| Rata-rata     | 0.95      | 0.95   | 0.95     | 0.95    |

**Tabel 4** Hasil analisa menggunakan algoritma SVM dengan data latih 60% dan data uji 40%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.95      | 0.95   | 0.95     | 0.95    |
| 2             | 0.93      | 0.93   | 0.93     | 0.93    |
| 3             | 0.93      | 0.93   | 0.93     | 0.93    |
| 4             | 0.94      | 0.94   | 0.94     | 0.94    |
| 5             | 0.95      | 0.95   | 0.95     | 0.95    |
| Rata-rata     | 0.94      | 0.94   | 0.94     | 0.94    |

**Tabel 5** Hasil analisa menggunakan algoritma SVM dengan data latih 50% dan data uji 50%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.93      | 0.93   | 0.93     | 0.93    |
| 2             | 0.93      | 0.93   | 0.93     | 0.93    |
| 3             | 0.93      | 0.93   | 0.93     | 0.93    |
| 4             | 0.94      | 0.94   | 0.94     | 0.94    |
| 5             | 0.93      | 0.93   | 0.93     | 0.93    |
| Rata-rata     | 0.93      | 0.93   | 0.93     | 0.93    |

### 3.2 Hasil Analisa Menggunakan Algoritma ANN

Analisa menggunakan algoritma ANN juga dilakukan dengan lima skenario tersebut yang akan diulang sebanyak lima kali juga di setiap skenarionya.

**Tabel 6** Hasil analisa menggunakan algoritma SVM dengan data latih 90% dan data uji 10%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.74      | 0.75   | 0.74     | 0.75    |
| 2             | 0.75      | 0.77   | 0.76     | 0.78    |
| 3             | 0.78      | 0.78   | 0.77     | 0.78    |
| 4             | 0.7       | 0.72   | 0.69     | 0.71    |
| 5             | 0.79      | 0.78   | 0.78     | 0.78    |
| Rata-rata     | 0.75      | 0.76   | 0.75     | 0.76    |

**Tabel 7** Hasil analisa menggunakan algoritma SVM dengan data latih 80% dan data uji 20%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.73      | 0.74   | 0.73     | 0.74    |
| 2             | 0.76      | 0.77   | 0.76     | 0.77    |
| 3             | 0.80      | 0.75   | 0.75     | 0.76    |
| 4             | 0.74      | 0.74   | 0.73     | 0.75    |
| 5             | 0.75      | 0.75   | 0.75     | 0.76    |
| Rata-rata     | 0.76      | 0.75   | 0.74     | 0.76    |

**Tabel 8** Hasil analisa menggunakan algoritma SVM dengan data latih 70% dan data uji 30%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.72      | 0.72   | 0.72     | 0.74    |
| 2             | 0.72      | 0.73   | 0.71     | 0.72    |
| 3             | 0.77      | 0.75   | 0.73     | 0.76    |
| 4             | 0.76      | 0.72   | 0.7      | 0.74    |
| 5             | 0.73      | 0.73   | 0.71     | 0.73    |
| Rata-rata     | 0.74      | 0.73   | 0.71     | 0.74    |

**Tabel 9** Hasil analisa menggunakan algoritma SVM dengan data latih 60% dan data uji 40%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.71      | 0.72   | 0.72     | 0.73    |
| 2             | 0.73      | 0.73   | 0.72     | 0.73    |
| 3             | 0.74      | 0.73   | 0.71     | 0.72    |
| 4             | 0.69      | 0.68   | 0.65     | 0.67    |
| 5             | 0.72      | 0.74   | 0.72     | 0.73    |
| Rata-rata     | 0.72      | 0.72   | 0.70     | 0.72    |

**Tabel 10** Hasil analisa menggunakan algoritma SVM dengan data latih 50% dan data uji 50%

| Eskperimen Ke | Precision | Recall | F1-Score | Akurasi |
|---------------|-----------|--------|----------|---------|
| 1             | 0.70      | 0.71   | 0.70     | 0.71    |
| 2             | 0.70      | 0.70   | 0.69     | 0.69    |
| 3             | 0.72      | 0.72   | 0.71     | 0.73    |
| 4             | 0.71      | 0.72   | 0.71     | 0.72    |
| 5             | 0.72      | 0.72   | 0.70     | 0.71    |
| Rata-rata     | 0.71      | 0.71   | 0.70     | 0.71    |

**Tabel 11** Hasil rata-rata analisa menggunakan algoritma SVM

| Data Latih | Data Uji | Precision | Recall | F1- Score | Akurasi |
|------------|----------|-----------|--------|-----------|---------|
| 90%        | 10%      | 0.96      | 0.96   | 0.96      | 0.97    |
| 80%        | 20%      | 0.95      | 0.95   | 0.95      | 0.95    |
| 70%        | 30%      | 0.95      | 0.95   | 0.95      | 0.95    |
| 60%        | 40%      | 0.94      | 0.94   | 0.94      | 0.94    |
| 50%        | 50%      | 0.93      | 0.93   | 0.93      | 0.93    |
| Rata-rata  |          | 0.95      | 0.95   | 0.95      | 0.95    |

**Tabel 12** Hasil rata-rata analisa menggunakan algoritma ANN

| Data Latih | Data Uji | Precision | Recall | F1- Score | Akurasi |
|------------|----------|-----------|--------|-----------|---------|
| 90%        | 10%      | 0.75      | 0.76   | 0.75      | 0.76    |
| 80%        | 20%      | 0.76      | 0.75   | 0.74      | 0.76    |
| 70%        | 30%      | 0.74      | 0.73   | 0.71      | 0.74    |
| 60%        | 40%      | 0.72      | 0.72   | 0.70      | 0.72    |
| 50%        | 50%      | 0.71      | 0.71   | 0.70      | 0.71    |
| Rata-rata  |          | 0.74      | 0.73   | 0.72      | 0.74    |

### 3.3 Hasil Analisis

Berdasarkan hasil rata-rata pada tabel 11 dan 12, algoritma SVM memiliki nilai yang tinggi di kisaran 0.93 sampai 0.97 di setiap skenario. Sedangkan pada algoritma ANN hasil *precision*, *recall*, *f1-score*, dan akurasi hanya berada dikisaran 0.70 sampai 0.76. Hal ini menunjukan bahwa algoritma SVM memiliki kinerja lebih baik daripada algoritma ANN. Walaupun perbedaan hasil lumayan signifikan, hasil dari algoritma ANN bisa dibilang sudah cukup baik.

#### 4. KESIMPULAN

Berdasarkan hasil analisis data tingkat obesitas menggunakan algoritma *SVM* dan *ANN* dapat disimpulkan bahwa algoritma *SVM* dengan *kernel linear* menghasilkan hasil yang memuaskan dimana nilai rata-rata dari semua skenario *precision*, *recall*, *f1-score*, dan akurasi secara berurutan adalah 0.95. Untuk analisis menggunakan *ANN* hasil yang didapatkan masih sedikit tinggi yaitu 0.74, 0.73, 0.72, 0.74. Dengan ini dapat disimpulkan bahwa hasil yang didapatkan sudah cukup baik dan algoritma *SVM* dengan *kernel linear* lebih unggul daripada algoritma *ANN* dalam menganalisa data tingkat obesitas.

Penelitian ini dapat dikembangkan lagi dengan menggunakan data dari berbagai negara lain serta melakukan eksperimen dengan skenario yang lebih beragam lagi. Dengan melakukan banyak eksperimen hasil yang didapat juga dapat lebih dipercaya. Dengan data yang lebih banyak dan skenario yang lebih banyak analisis tingkat obesitas dapat berkembang secara signifikan.

#### DAFTAR PUSTAKA

- [1] N. Puspitasari, "Faktor Kejadian Obesitas Sentral Pada Usia Dewasa," *Higea Journal of Public Health Research and Development*, vol. 2, no. 2, p. 11, 2018.
- [2] M. Y. Lubis, D. Hermawan, U. Febriani and A. Farich, "Hubungan Antara Faktor Keturunan, Jenis Kelamin Dan Tingkat Sosial Ekonomi Orang Tua Dengan Kejadian Obesitas Pada Mahasiswa Di Universitas Malahayati Tahun 2020," *Human Care Journal*, vol. 5, no. 4, p. 10, 2020.
- [3] S. Wahyuni, S. M. Ibrahim and H. R. P. Negara, "Metode Support Vector Machine dan Modifikasinya Dalam," *LPPM*, vol. 1, p. 10, 2022.
- [4] D. Oktavia, Y. R. Ramadahan and M, "Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, p. 11, 2023.
- [5] M. Fathurrohman, R. L. Lambang G. H and D. D. Susilo, "Diagnosa Kerusakan Bantalan Bola Menggunakan Metode Support Vector Machine Learning," *Mekanika: Majalah Ilmiah Mekanika*, vol. 18, no. 1, p. 8, 2019.
- [6] H. D. Bhakti, "Aplikasi Artificial Neural Network (ANN) untuk Memprediksi Masa Studi Mahasiswa Program Studi Teknik Informatika Universitas Muhammadiyah Gresik," *Jurnal Eksplora Informatika*, vol. 9, no. 1, p. 8, 2019.
- [7] M. A. Putri and T. Sukmono, "Analisa Peramalan Penjualan Kerupuk Udang dengan menggunakan Metode Artificial Neural Network (ANN)," *Innovate Technologica: Journal of Analysis and Inventions*, vol. 2, no. 4, p. 11, 2023.