Import library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
```

Load dataset

```
data = pd.read_csv('/content/obesitas.csv')
```

Eksplorasi dataset

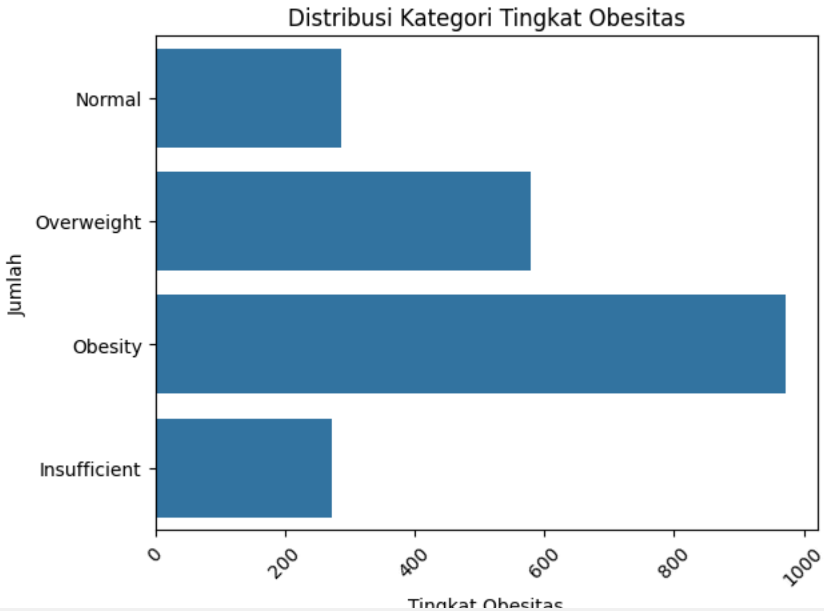
```
print("\nDataset Overview:\n", data.head())
print("\nInfo Dataset:\n")
data.info()
print("\nMissing Values:\n", data.isnull().sum())
print("\nDescriptive Statistics:\n", data.describe())
```

```
 2   Height                          2111 non-null   float64
 3   Weight                          2111 non-null   float64
 4   family_history_with_overweight  2111 non-null   object
 5   FAVC                            2111 non-null   object
 6   FCVC                            2111 non-null   float64
 7   NCP                             2111 non-null   float64
 8   CAEC                            2111 non-null   object
 9   SMOKE                           2111 non-null   object
 10  CH2O                            2111 non-null   float64
 11  SCC                             2111 non-null   object
 12  FAF                             2111 non-null   float64
 13  TUE                             2111 non-null   float64
 14  CALC                            2111 non-null   object
 15  MTRANS                          2111 non-null   object
 16  NObeyesdad                      2111 non-null   object
```

```
              CH2O          FAF          TUE
count  2111.000000  2111.000000  2111.000000
mean      2.008011     1.010298     0.657866
std       0.612953     0.850592     0.608927
min       1.000000     0.000000     0.000000
25%       1.584812     0.124505     0.000000
50%       2.000000     1.000000     0.625350
75%       2.477420     1.666678     1.000000
max       3.000000     3.000000     2.000000
```

Visualisasi data

```
sns.countplot(data['NObeyesdad'])
plt.title('Distribusi Kategori Tingkat Obesitas')
plt.xlabel('Tingkat Obesitas')
plt.ylabel('Jumlah')
plt.xticks(rotation=45)
plt.show()
```



Preprocessing Data

Encoding untuk fitur kategorikal

```
encoder = LabelEncoder()
for col in data.select_dtypes(include=['object']).columns:
    data[col] = encoder.fit_transform(data[col])
```

Pisahkan fitur dan label

```
X = data.drop(columns=['NObeyesdad'])
y = data['NObeyesdad']
```

Standarisasi fitur

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Split data untuk training dan testing

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Model 1: Decision Tree

```
print("\n=== Decision Tree Classifier ===")
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)
dt_predictions = dt_model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, dt_predictions))
print("Classification Report:\n", classification_report(y_test, dt_predictions))
```

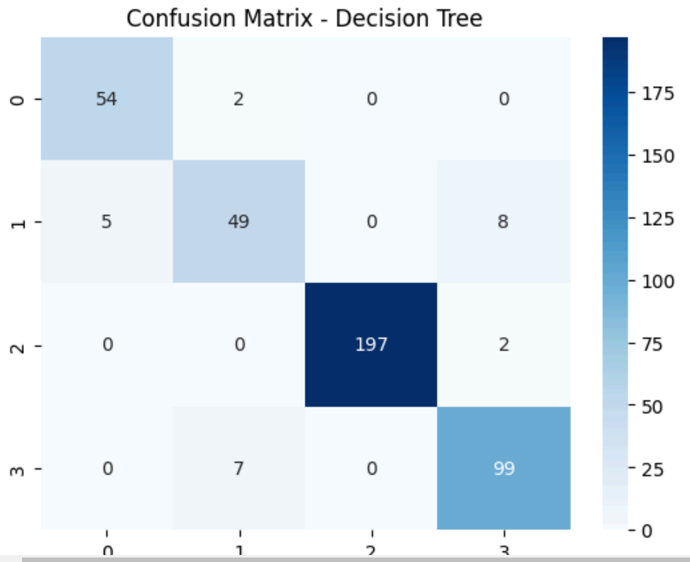
```
=== Decision Tree Classifier ===
Accuracy: 0.9432624113475178
Classification Report:
               precision    recall  f1-score   support

           0       0.92      0.96      0.94        56
           1       0.84      0.79      0.82        62
           2       1.00      0.99      0.99       199
           3       0.91      0.93      0.92       106

    accuracy                           0.94       423
   macro avg       0.92      0.92      0.92       423
weighted avg       0.94      0.94      0.94       423
```

Confusion Matrix Decision Tree

```
sns.heatmap(confusion_matrix(y_test, dt_predictions), annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Decision Tree')
plt.show()
```



Model 2: K-Nearest Neighbors

```
print("\n=== K-Nearest Neighbors (KNN) ===")
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
knn_model.fit(X_train, y_train)
knn_predictions = knn_model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, knn_predictions))
print("Classification Report:\n", classification_report(y_test, knn_predictions))
```

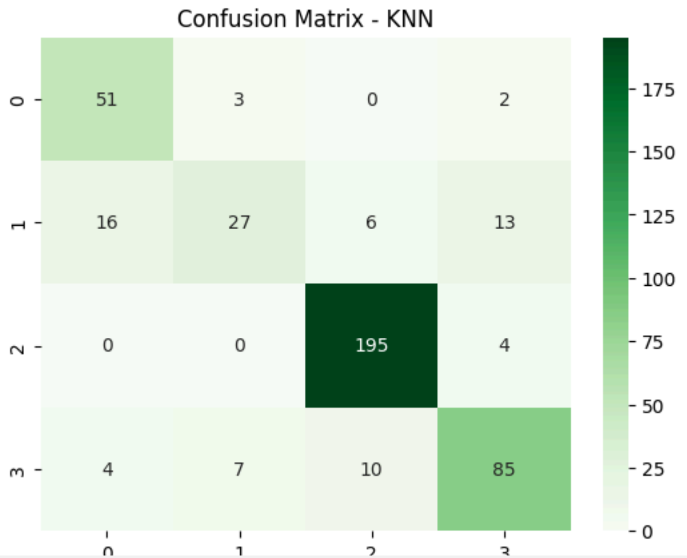
```
=== K-Nearest Neighbors (KNN) ===
Accuracy: 0.8463356973995272
Classification Report:
               precision    recall  f1-score   support

           0       0.72      0.91      0.80        56
           1       0.73      0.44      0.55        62
           2       0.92      0.98      0.95       199
           3       0.82      0.80      0.81       106

    accuracy                           0.85       423
   macro avg       0.80      0.78      0.78       423
weighted avg       0.84      0.85      0.84       423
```

Confusion Matrix KNN

```
sns.heatmap(confusion_matrix(y_test, knn_predictions), annot=True, fmt='d', cmap='Greens')
plt.title('Confusion Matrix - KNN')
plt.show()
```



Perbandingan Kinerja

```
dt_accuracy = cross_val_score(dt_model, X, y, cv=5, scoring='accuracy').mean()
knn_accuracy = cross_val_score(knn_model, X, y, cv=5, scoring='accuracy').mean()

print("\n=== Model Comparison ===")
print(f"Decision Tree Accuracy (CV): {dt_accuracy:.2f}")
print(f"KNN Accuracy (CV): {knn_accuracy:.2f}")
```

```
=== Model Comparison ===
Decision Tree Accuracy (CV): 0.93
KNN Accuracy (CV): 0.81
```

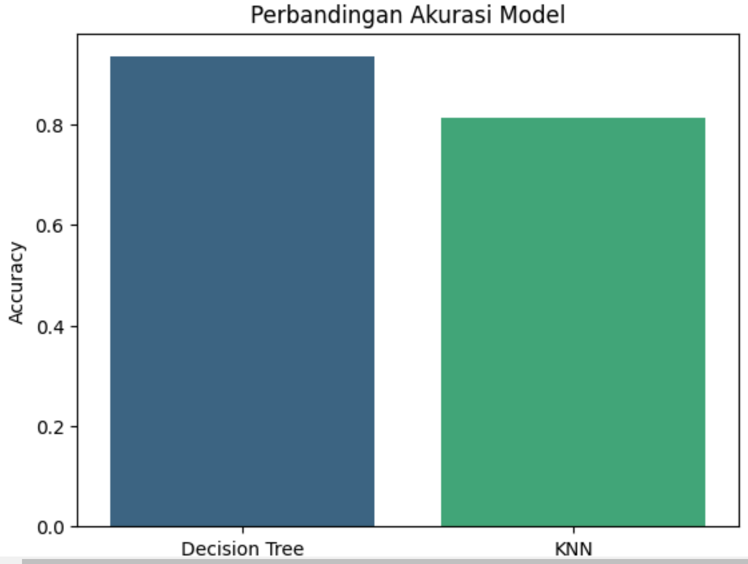
Visualisasi Perbandingan

```
models = ['Decision Tree', 'KNN']
accuracies = [dt_accuracy, knn_accuracy]
sns.barplot(x=models, y=accuracies, palette='viridis')
plt.title('Perbandingan Akurasi Model')
plt.title('Perbandingan Akurasi Model')
plt.ylabel('Accuracy')
plt.show()
```

```
<ipython-input-21-7df307d9cee7>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x=models, y=accuracies, palette='viridis')
```


Perbandingan Akurasi Model
Accuracy
Decision Tree
KNN