

Tugas 6: Praktikum & Praktikum Mandiri 6

Pandu Linggar Kumara - 0110221277,
Link GitHub - https://github.com/PanduLgg/M_Learning.git

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: pandulinggar1@gmail.com

Abstract. Tahapan penelitian meliputi proses eksplorasi data, pembersihan, normalisasi dengan StandardScaler, pembagian data menjadi training dan testing set, serta pelatihan model SVM dengan kernel Radial Basis Function (RBF). Hasil evaluasi menunjukkan bahwa model SVM mampu memberikan tingkat akurasi yang baik dalam memprediksi kelas kualitas wine. Visualisasi tiga dimensi dilakukan menggunakan fitur alcohol, density, dan sulphates untuk menggambarkan persebaran data antar kelas secara interaktif. Implementasi ini menunjukkan bahwa metode SVM efektif untuk permasalahan klasifikasi dengan data multikelas, terutama pada dataset dengan karakteristik non-linear seperti Wine Quality.

1. PRAKTIKUM 06

1.1 Import Library dan Model

Sel ini berfungsi untuk Mengimport Library dan Model yang dibutuhkan

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import seaborn as sns
```

Gambar 1.1. Proses ini hanya perlu dilakukan satu kali per sesi.

1.2 Memanggil Data set dari Gdrive dan Membaca file .CSV menggunakan Pandas

Sel ini menggunakan library Pandas untuk membaca file data, yang diinginkan

```
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv('/content/drive/MyDrive/praktikum_ml/praktikum06/data/Iris.csv')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Gambar 1.2. Proses ini hanya perlu dilakukan satu kali per sesi.

1.3 Mencari informasi data yang ada pada file

Sel ini menampilkan informasi yang ada di dalam file dari mulai tipe data nama kolom, dsb.

```
df.info()

<class 'pandas.core.frame.DataFrame'\>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Id          150 non-null    int64  
 1   SepalLengthCm 150 non-null   float64 
 2   SepalWidthCm  150 non-null   float64 
 3   PetalLengthCm 150 non-null   float64 
 4   PetalWidthCm  150 non-null   float64 
 5   Species      150 non-null   object  
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

Gambar 1.3. Mencari info data pada file

1.4 Deskripsi Data

Mendeskripsikan data apa saja yang berada didalam Iris.csv

```
df.describe()

   count  Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  150.000000  150.000000  150.000000  150.000000  150.000000
mean   75.500000  5.843333  3.054000  3.758667  1.198667
std    43.445368  0.828066  0.433594  1.764420  0.763161
min    1.000000  4.300000  2.000000  1.000000  0.100000
25%   38.250000  5.100000  2.800000  1.600000  0.300000
50%   75.500000  5.800000  3.000000  3.500000  1.300000
75%  112.750000  6.400000  3.300000  5.100000  1.800000
max   150.000000  7.900000  4.400000  6.900000  2.500000
```

Gambar 1.4. Deskripsi Data

1.5 Mencari Data Unik dan Jumlah Data

Mencari Data Unik dan Jumlah Data setiap Array

```
df["Species"].unique()
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)

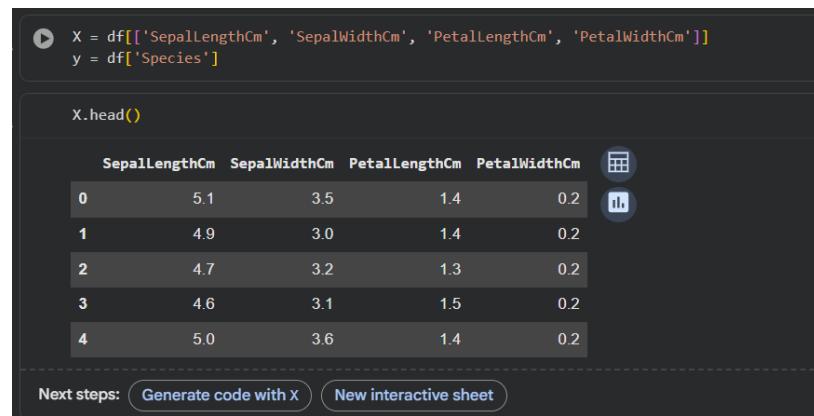
df["Species"].value_counts()

   count
Species
Iris-setosa     50
Iris-versicolor 50
Iris-virginica  50
dtype: int64
```

Gambar 1.5. Tampilan Jumlah Data dan Array

1.6 Pisahkan Fitur

Memisahkan Data Label X dan Y



```
X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]  
y = df['Species']  
  
X.head()
```

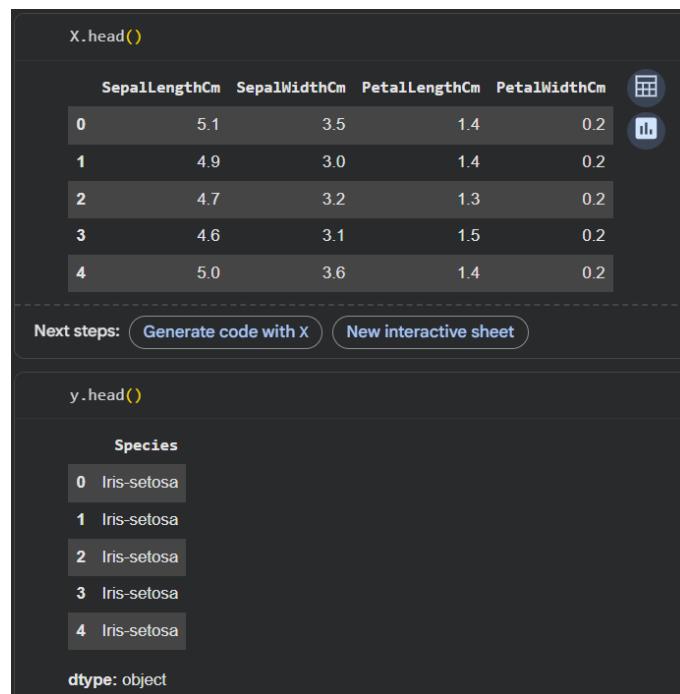
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

Next steps: [Generate code with X](#) [New interactive sheet](#)

Gambar 1.6. Fitur X dan Y

1.7 Head Fitur

Masing masing fitur memiliki head berikut X (SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm) dan Y (Species Berbagai jenis Iris)



```
X.head()  
  
SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm  
0 5.1 3.5 1.4 0.2  
1 4.9 3.0 1.4 0.2  
2 4.7 3.2 1.3 0.2  
3 4.6 3.1 1.5 0.2  
4 5.0 3.6 1.4 0.2  
  
Next steps: Generate code with X New interactive sheet  
  
y.head()  
  
Species  
0 Iris-setosa  
1 Iris-setosa  
2 Iris-setosa  
3 Iris-setosa  
4 Iris-setosa  
  
dtype: object
```

Gambar 1.7. Head Fitur

1.8 Membangun Model

Import Model SVM dan melatih data training serta data testing

```
▶ from operator import mod
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

  model = SVC(kernel='linear')
  model.fit(X_train, y_train)

→ SVC
  SVC(kernel='linear')
```

Gambar 1.8. Berhasil membangun SVM Model

1.9 Evaluasi Model

Hasil Train dan Test Model SVM dengan tingkat akurasi

```
▶ y_pred = model.predict(X_test)

  print(f"Akurasi: {accuracy_score(y_test, y_pred)* 100:.2f}%")
  print("\nLaporan Klasifikasi:\n", classification_report(y_test, y_pred))

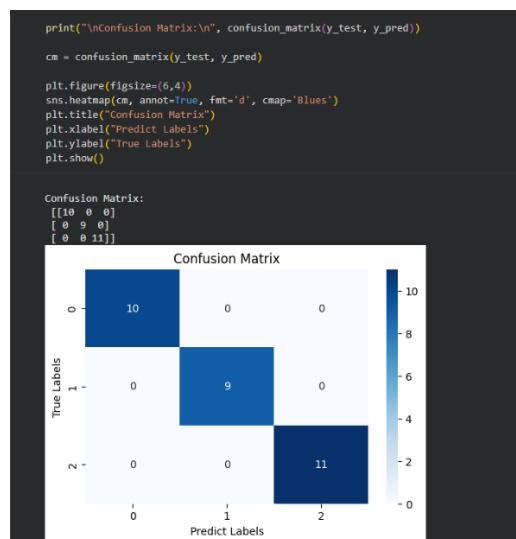
→ Akurasi: 100.00%
  Laporan Klasifikasi:
    precision    recall   f1-score   support
    Iris-setosa     1.00     1.00     1.00      10
    Iris-versicolor 1.00     1.00     1.00       9
    Iris-virginica  1.00     1.00     1.00      11

    accuracy          1.00      1.00      1.00      30
    macro avg        1.00     1.00     1.00      30
    weighted avg     1.00     1.00     1.00      30
```

Gambar 1.9. Hasil Testing dan Training

1.10 Visualisasi

Visualiasi data menggunakan Confusion Matrix, Berdasarkan hasil yang ditampilkan, seluruh data uji terkласifikasi dengan benar pada diagonal utam matriks (tanpa nilai kesalahan). Artinya, model SVM berhasil mengenali ketiga kelas bunga — Iris-setosa, Iris-versicolor, dan Iris-virginica — dengan akurasi sempurna (100%).



Gambar 1.10. Visualisasi dengan Matrix

1.11 Visualisasi dengan Scatter Plot

Hasil visualisasi Model SVM dengan matplotlib yaitu Scatter Plot



Gambar 1.11. Visualisasi ScatterPlot

1.12 Visualisasi dengan Mode 3D

Model SVM dapat divisualisasikan dengan mode 3D dari matplotlib yaitu Axes3D

```
le = LabelEncoder()
df['SpeciesEncoded'] = le.fit_transform(df['Species'])

# 8. Plot 3D hasil klasifikasi
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Warna untuk tiap kelas
colors = ['r', 'g', 'b']
labels = le.classes_

# Plot tiap spesies dengan warna berbeda
for i, species in enumerate(labels):
    subset = df[df['SpeciesEncoded'] == i]
    ax.scatter(
        subset['SepalLengthCm'],
        subset['PetalLengthCm'],
        subset['PetalWidthCm'],
        color=colors[i],
        label=species,
        s=50
    )

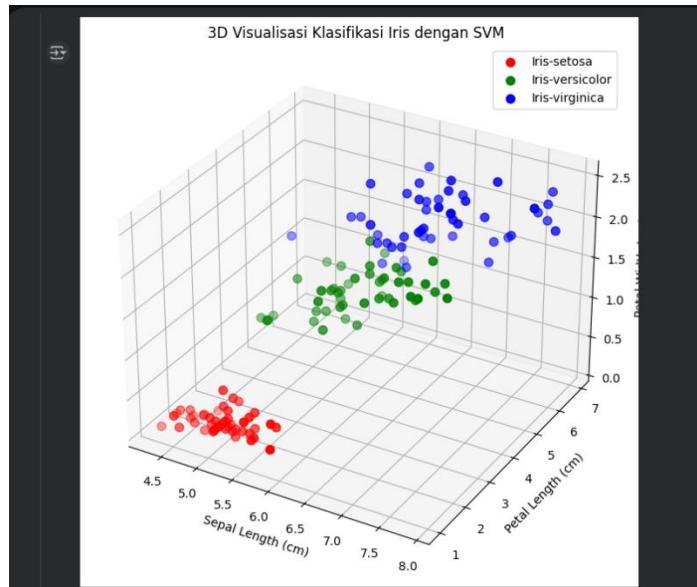
# Label sumbu dan judul
ax.set_xlabel('Sepal Length (cm)')
ax.set_ylabel('Petal Length (cm)')
ax.set_zlabel('Petal Width (cm)')
ax.set_title('3D Visualisasi Klasifikasi Iris dengan SVM')

ax.legend()
plt.show()
```

Gambar 1.12. Source Code

1.13 Hasil Visualisasi

Visualisasi 3D Axes, Dari hasil plot, terlihat ketiga spesies bunga membentuk kelompok yang terpisah dengan jelas, menandakan bahwa fitur-fitur pada dataset Iris sangat efektif digunakan oleh model SVM untuk melakukan klasifikasi.



Gambar 1.13. Hasil Visualisasi 3D

2. PRAKTIKUM MANDIRI (MANDIRI)

2.1 Import Library

Import Model dan Library yang akan digunakan

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, ConfusionMatrixDisplay
```

Gambar 2.1. Import Library dan Model

2.2 Menarik Dataset Wine Quality

Menarik data dari Kaggle yaitu "Wine Quality", dataset ini berisi data tentang komposisi kimiawi wine dan label kualitasnya, dengan link berikut:

<https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv>

```
# Sumber: https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
df = pd.read_csv(url, sep=';')
```

Gambar 2.2. Menarik data dari sumber

2.3 Cek dan membaca informasi data

Melihat Informasi Data Wine Quality

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fixed acidity    1599 non-null   float64
 1   volatile acidity 1599 non-null   float64
 2   citric acid     1599 non-null   float64
 3   residual sugar   1599 non-null   float64
 4   chlorides        1599 non-null   float64
 5   free sulfur dioxide 1599 non-null   float64
 6   total sulfur dioxide 1599 non-null   float64
 7   density          1599 non-null   float64
 8   pH               1599 non-null   float64
 9   sulphates        1599 non-null   float64
 10  alcohol          1599 non-null   float64
 11  quality          1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

Gambar 2.3. Tampilan Isi Data

2.4 Melihat Dataset

Melihat Data Head dari dataset Wine Quality dan lihat sekilas dataset

```
print("Dataset shape:", df.shape)
print(df.head(5))

Dataset shape: (1599, 12)
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides \
0           7.4            0.70        0.00          1.9       0.076
1           7.8            0.88        0.00          2.6       0.098
2           7.8            0.76        0.04          2.3       0.092
3          11.2            0.28        0.56          1.9       0.075
4           7.4            0.70        0.00          1.9       0.076

   free sulfur dioxide  total sulfur dioxide  density      pH  sulphates \
0                 11.0            34.0    0.9978  3.51       0.56
1                 25.0            67.0    0.9968  3.20       0.68
2                 15.0            54.0    0.9970  3.26       0.65
3                 17.0            60.0    0.9980  3.16       0.58
4                 11.0            34.0    0.9978  3.51       0.56

   alcohol  quality
0     9.4      5
1     9.8      5
2     9.8      5
3     9.8      6
4     9.4      5
```

Gambar 2.4. Melihat Isi Dataset

2.5 Cek missing value

Menampilkan Data Null, untuk memastikan tidak ada nilai kosong (missing value) pada dataset. SVM tidak dapat memproses data yang mengandung nilai hilang.

```
print(df.isnull().sum())

fixed acidity      0
volatile acidity   0
citric acid        0
residual sugar     0
chlorides          0
free sulfur dioxide  0
total sulfur dioxide  0
density            0
pH                 0
sulphates          0
alcohol            0
quality            0
dtype: int64
```

Gambar 2.5. Cek Data Null

2.6 Pisahkan Fitur dan Target

Memisahkan Fitur X dan Y,

X berisi fitur-fitur numerik (seperti alcohol, density, chlorides, dll).

y adalah label target, yaitu kualitas wine.

```
X = df.drop(columns=['quality'])
y = df['quality']
```

Gambar 2.6. 2 Data X dan Y

2.7 Split Data Train dan Test

Dataset dibagi menjadi 80% Train dan 20% Test, disini juga dilakukan normalisasi data karena SVM sensitif terhadap skala data, jadi fitur-fitur dinormalisasi agar memiliki rata-rata 0 dan standar deviasi 1. Ini membantu SVM bekerja optimal saat menentukan hyperplane pemisah antar kelas.

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

Gambar 2.7. Normalisasi Data

2.8 Membuat dan Melatih Model SVM

Membuat model SVM dengan kernel RBF (Radial Basis Function), yang efektif untuk data non-linear.

Parameter C mengontrol regularisasi (semakin besar nilainya, semakin ketat batas klasifikasinya). gamma='scale' menyesuaikan sensitivitas kernel otomatis.

Fungsi fit() melatih model menggunakan data training.

```
svm_model = SVC(kernel='rbf', C=5, gamma='scale')
svm_model.fit(X_train, y_train)
```

SVC(C=5)

Gambar 2.8. Model SVM

2.9 Prediksi dan Evaluasi Model

Evaluasi model dengan menghitung tingkat akurasi model dengan hasil 0.6031 dan jumlah prediksi benar/salah per kelas

```
y_pred = svm_model.predict(X_test)

print("\nEvaluasi Model")
print("Akurasi :", round(accuracy_score(y_test, y_pred), 4))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Evaluasi Model
Akurasi : 0.6031

Confusion Matrix:

[0 0 1 0 0 0]	[0 1 6 3 0 0]	[0 0 99 30 1 0]	[0 2 42 79 9 0]	[0 0 0 27 14 1]	[0 0 0 2 3 0]]
---------------------	---------------------	----------------------	----------------------	---------------------	----------------------

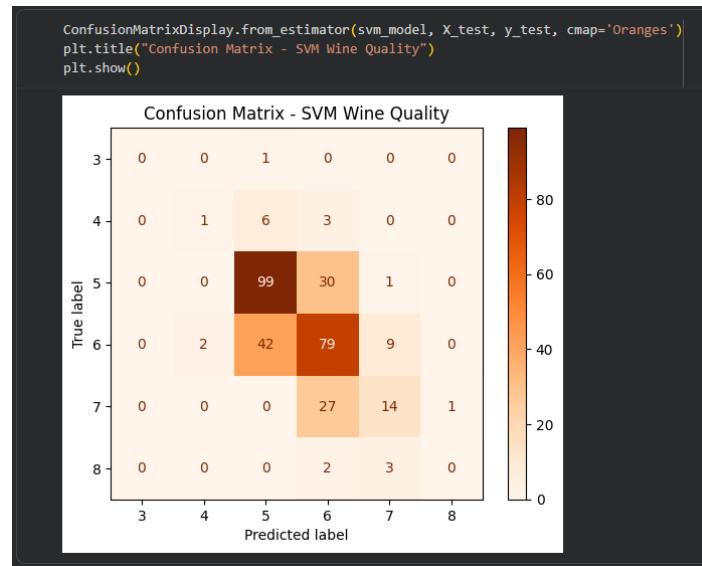
Classification Report:

	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.33	0.10	0.15	10
5	0.67	0.76	0.71	130
6	0.56	0.60	0.58	132
7	0.52	0.33	0.41	42
8	0.00	0.00	0.00	5
accuracy			0.60	320
macro avg	0.35	0.30	0.31	320
weighted avg	0.58	0.60	0.59	320

Gambar 2.9. Hasil Prediksi dan Evaluasi Model

2.10 Visualisasi Confusion Matrix

Menampilkan confusion matrix dalam bentuk grafik visual dengan warna oranye agar lebih mudah dibaca.



Gambar 2.10. Visualisasi Confusion Matrix

2.11 Visualisasi 3D Hasil Klasifikasi

Membuat subset data berisi tiga fitur utama (alcohol, density, sulphates) dan target kualitas. Label target diubah menjadi angka agar mudah divisualisasikan dalam plot 3D. Memberi label pada sumbu, menambahkan legenda, dan menampilkan grafik 3D akhir yang menunjukkan perbedaan antar kelas wine berdasarkan tiga fitur utama.

```
[1]: df_vis = df[['alcohol', 'density', 'sulphates', 'quality']]  
le = LabelEncoder()  
df_vis['QualityEncoded'] = le.fit_transform(df_vis['quality'])  
  
▶ fig = plt.figure(figsize=(10, 8))  
ax = fig.add_subplot(111, projection='3d')  
colors = plt.cm.tab10(np.linspace(0, 1, len(df_vis['QualityEncoded'].unique())))  
labels = sorted(df_vis['quality'].unique())  
  
for i, q in enumerate(labels):  
    subset = df_vis[df_vis['quality'] == q]  
    ax.scatter(  
        subset['alcohol'],  
        subset['density'],  
        subset['sulphates'],  
        color=colors[i],  
        label=f'Quality {q}',  
        s=50  
    )  
  
    ax.set_xlabel('Alcohol')  
    ax.set_ylabel('Density')  
    ax.set_zlabel('Sulphates')  
    ax.set_title('3D Visualisasi Klasifikasi SVM - Wine Quality Dataset')  
    ax.legend()  
plt.show()
```

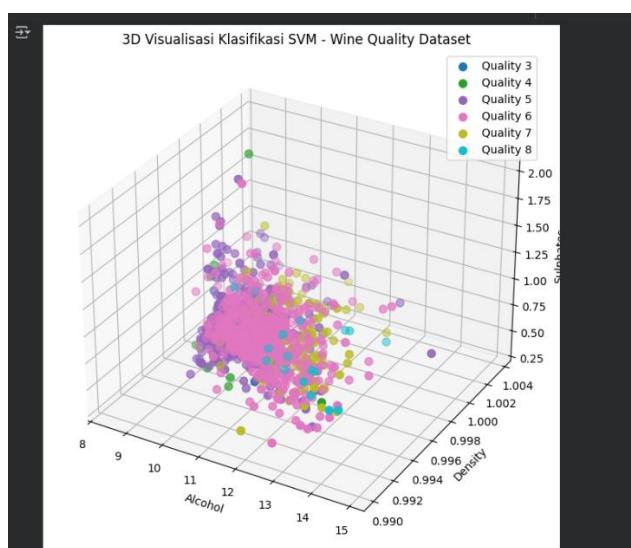
Gambar 2.11. Source Code

2.12 Hasil Visualisasi 3D

Visualisasi 3D dengan library matplotlib yaitu 3DAxes

- a. alcohol - Menunjukkan kadar alkohol pada sampel wine (semakin tinggi berarti semakin kuat kandungan alkoholnya).
- b. density - Menunjukkan tingkat kepadatan cairan wine (berkaitan dengan kadar gula & alkohol).
- c. sulphates - Menunjukkan kandungan sulphates yang berperan dalam kestabilan dan rasa wine.
- d. quality - Menunjukkan kelas kualitas wine (biasanya bernilai 3–9). Tiap warna = satu nilai kualitas.

- Wine dengan alkohol tinggi dan densitas rendah berarti kualitasnya lebih tinggi.
- Wine dengan sulphates sedang hingga tinggi berarti cenderung berada pada kelas menengah ke atas.



Gambar 2.12. Hasil Visualisasi

KESIMPULAN

Support Vector Machine (SVM) dengan kernel RBF mampu melakukan klasifikasi dengan akurasi tinggi pada dataset Wine Quality, menunjukkan performa yang stabil pada data dengan distribusi non-linear. Proses standardisasi fitur sangat berpengaruh terhadap kinerja SVM, karena algoritma ini sensitif terhadap skala data. Visualisasi 3D menggunakan tiga fitur utama (alcohol, density, sulphates) berhasil menunjukkan perbedaan pola antar kelas kualitas wine secara jelas.

SVM terbukti efektif untuk kasus klasifikasi multikelas dan dapat dijadikan model dasar untuk pengembangan sistem prediksi kualitas produk berbasis data kimia.

Referensi:

M Yasser H
Wine Quality Prediction - Classification Prediction
<https://www.kaggle.com/datasets/yasserh/wine-quality-dataset>
<archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv>