

Praktikum Fisika Komputasi

Modul 11

Septian Tri Laksono

1227030032

```
def Trapezoid(a,b,f):
    n = 100
    def trapezoid(f,a,b,n=100):
        h = (b - a)/n
        sum = 0.0
        for i in range (1,n):
            x = a + i * h
            sum = sum + f(x)
        integral = (h/2)*(f(a)+2*sum+f(b))
        return integral
    integral = trapezoid(f,a,b,n)
    print(a, ", ", b, ", ", round(integral,2))

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 2*x)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 2*x+2)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 2*x+4)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 4*x+6)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 6*x+8)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 8*x+10)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 10*x+12)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 12*x+14)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 14*x+12)

for i in range(0,5):
```

```

Trapezoid(i+1,i+2,lambda x: 20*x+40)

#Import library yang diperlukan
import numpy as np
import pandas as pd
from sklearn import svm
from google.colab import drive
import matplotlib.pyplot as plt

#Mount Google Drive
drive.mount('/content/drive')

#Path ke file Database.txt di Google Drive
file_path = '/content/drive/My Drive/Trapezoid.txt' # Ganti dengan path sesuai lokasi file Anda di Google Drive

#Membaca data dari file
Database = pd.read_csv(file_path, sep=",", header=0)

# x Data, y Target
X = Database[['a', 'b']] #Pastikan kolom sesuai dengan nama yang ada di file
y = Database ['Target']

#membuat dan melatih model SVH
clf = svm.SVC()
clf.fit(X.values, y)

#Melakukan prediksi
y_pred = clf.predict(X.values)

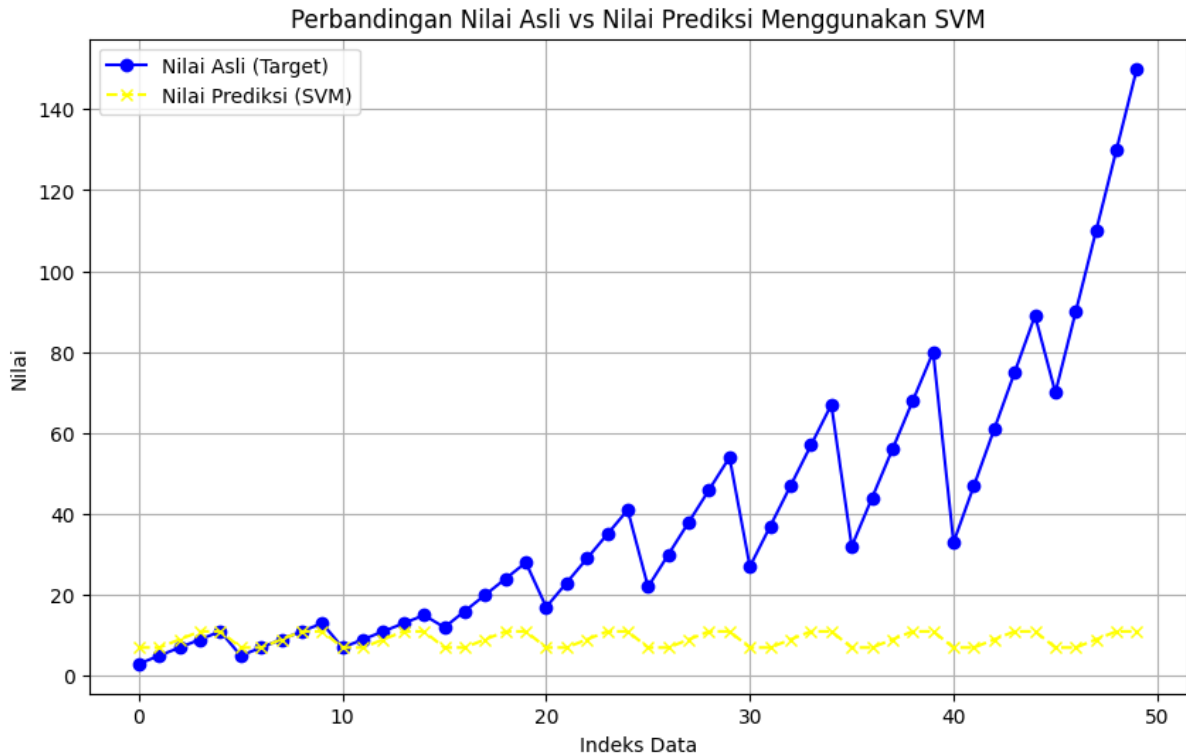
# Menampilkan hasil prediksi
print("Hasil prediksi:")
for i, pred in enumerate(y_pred):
    print(f"{X.iloc[i, 0]}, {X.iloc[i, 1]}, {pred}")

# Membuat plot perbandingan nilai asli dengan nilai prediksi
plt.figure(figsize=(10, 6))

# Periksa panjang data sebelum plotting untuk menghindari error
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)', color='blue')
plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi (SVM)', color='yellow')

# Menambahkan label dan judul
plt.xlabel('Indeks Data')
plt.ylabel('Nilai')
plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
plt.legend()
plt.grid()
plt.show()

```



Berdasarkan grafik yang dihasilkan, terdapat perbedaan pola . Nilai asli (garis biru) menunjukkan pola pertumbuhan yang non-linear, terutama setelah indeks data sekitar 30, dengan kenaikan yang semakin tajam, sedangkan nilai prediksi (garis kuning) tetap hampir konstan di sepanjang indeks data, menunjukkan bahwa model tidak dapat menangkap tren kenaikan yang ada pada data asli. Dari grafik, terlihat bahwa prediksi tidak mendekati nilai asli, terutama pada indeks data yang lebih tinggi. Hal ini kemungkinan besar disebabkan oleh persamaan yang digunakan tidak sesuai dengan sifat data dan model terlalu sederhana, sehingga tidak mampu mempelajari hubungan kompleks dalam data.

```
def Trapezoid(a,b,f):
    n = 100
    def trapezoid(f,a,b,n=100):
        h = (b - a)/n
        sum = 0.0
        for i in range (1,n):
            x = a + i * h
            sum = sum + f(x)
        integral = (h/2)*(f(a)+2*sum+f(b))
        return integral
    integral = trapezoid(f,a,b,n)
    print(a, ", ", b, ", ", round(integral,2))

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 2*x)
```

```

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 2*x+2)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 2*x+4)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 4*x+6)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 6*x+8)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 8*x+10)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 10*x+12)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 12*x+14)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 14*x+16)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambdax: 20*x+40)

#Import library yang diperlukan
import numpy as np
import pandas as pd
from sklearn import svm
from google.colab import drive
import matplotlib.pyplot as plt

#Mount Google Drive
drive.mount('/content/drive')

#Path ke file Database.txt di Google Drive
file_path = '/content/drive/My Drive/Trapezoid2.txt' # Ganti dengan path sesuai lokasi file Anda di Google Drive

#Membaca data dari file
Database = pd.read_csv(file_path, sep=";", header=0)

# x Data, y Target
X = Database[['a', 'b']] #Pastikan kolom sesuai dengan nama yang ada di file
y = Database ['Target']

#membuat dan melatih model SVH
clf = svm.SVC()

```

```

clf.fit(X.values, y)

#Melakukan prediksi
y_pred = clf.predict(X.values)

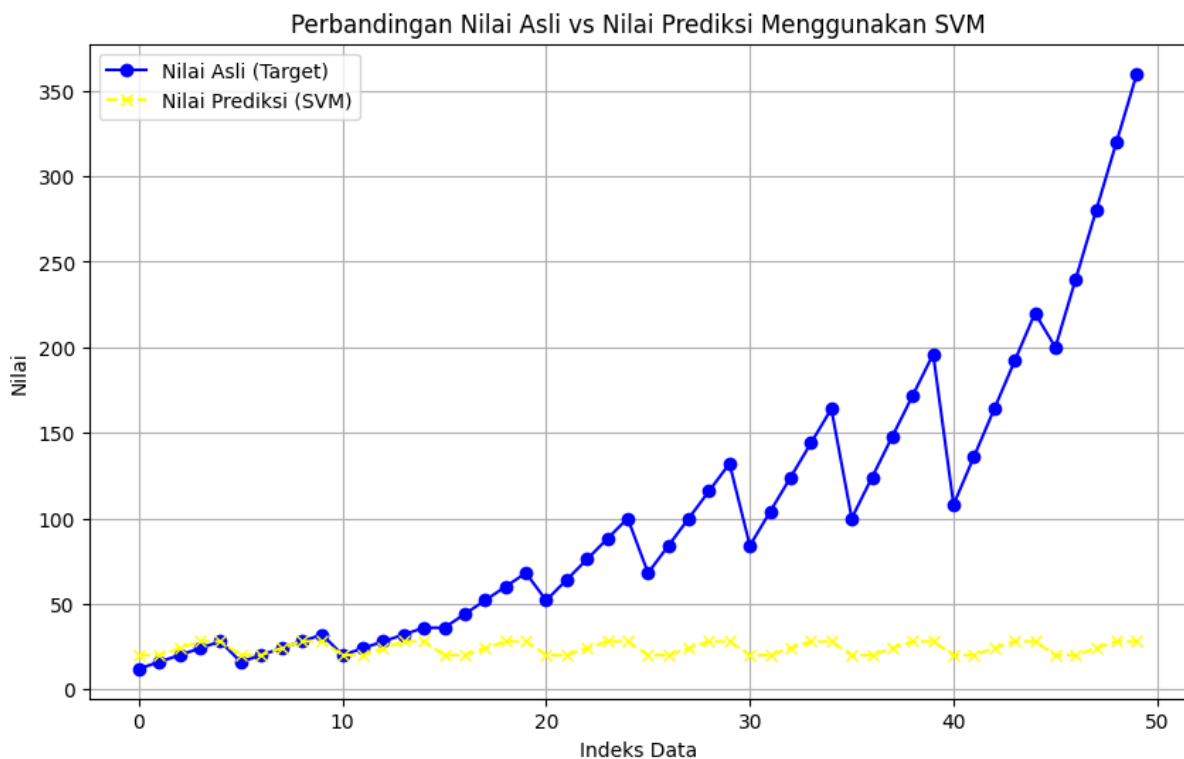
# Menampilkan hasil prediksi
print("Hasil prediksi:")
for i, pred in enumerate(y_pred):
    print(f"{X.iloc[i, 0]}, {X.iloc[i, 1]}, {pred}")

# Membuat plot perbandingan nilai asli dengan nilai prediksi
plt.figure(figsize=(10, 6))

# Periksa panjang data sebelum plotting untuk menghindari error
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)', color='blue')
plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi (SVM)', color='yellow')

# Menambahkan label dan judul
plt.xlabel('Indeks Data')
plt.ylabel('Nilai')
plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
plt.legend()
plt.grid()
plt.show()

```



Berdasarkan hasil grafik yang kedua juga didapat perbedaan yang signifikan, terlihat dari nilai asli (ditunjukkan dengan garis biru) menunjukkan pola kenaikan yang cukup tajam terutama setelah indeks data tertentu (sekitar 50 ke atas), sementara nilai prediksi (garis

kuning) tetap relatif konstan dan tidak mengikuti pola kenaikan nilai asli. Model SVM yang digunakan tampaknya tidak mampu menangkap pola atau tren dari nilai asli. Kemungkinan besar, ini disebabkan oleh ketidakcocokan antara kemampuan model dan sifat data yang diprediksi (seperti pola non-linear yang kompleks) dan mungkin model ini tidak cukup dilatih dengan parameter yang tepat, seperti pemilihan parameter regulasi, atau tidak cukup data pelatihan.

Metode trapezoid digunakan untuk menghitung integral numerik suatu fungsi $f(x)$ dalam interval $[a,b]$. Metode ini membagi area di bawah kurva menjadi segmen berbentuk trapesium, menghitung luas masing-masing trapesium, dan menjumlahkannya. Dengan mendefinisikan a: batas bawah integral. b: batas atas integral. $f(x)$: fungsi yang akan diintegrasikan. Dan n: jumlah segmen yang digunakan untuk membagi interval (100). Kemudian menghitung integral menggunakan rumus berikut.

$$\text{Integral} = \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right]$$

SVM adalah algoritma yang digunakan untuk klasifikasi atau regresi. Dalam kasus ini, SVM digunakan untuk memprediksi nilai integral berdasarkan parameter input (a dan b). Data dibaca dari file CSV menggunakan pandas yang mana kolom a dan b digunakan sebagai fitur input (x), dan kolom Target digunakan sebagai label output (y). Model SVM dibuat menggunakan sklearn dan kemudian model dilatih dengan data input x (fitur batas integral) dan target y (hasil integral trapezoid). Kemudian hasilnya berupa nilai asli (y) yang dibandingkan dengan nilai prediksi (y prediksi) menggunakan grafik garis yang mana garis biru menunjukkan nilai asli (Target) dan garis kuning menunjukkan nilai prediksi (hasil SVM).