

Laporan Praktikum Fisika Komputasi

Tugas 11 : Senin, 02 Desember 2024

Disusun oleh : Najlah Rupaiah

NIM : 1227030025

1. Hasil prediksi integral trapezoid menggunakan *Support Vector Machine* pada beberapa persamaan:

Hasil prediksi:

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0

X: 5, 6 -> 11.0

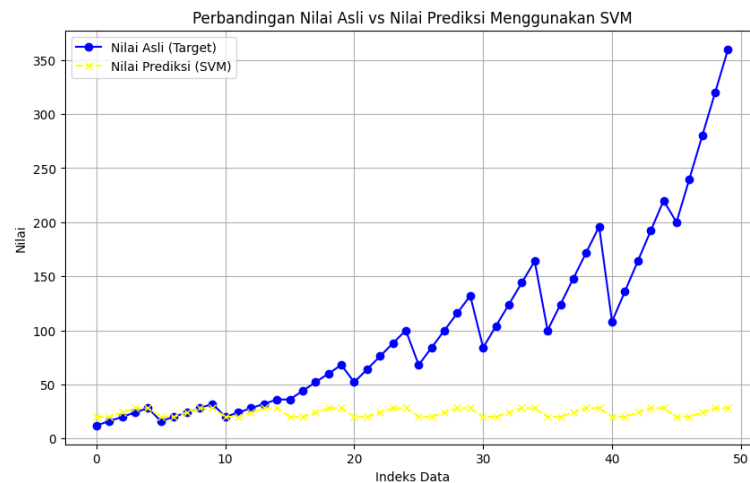
X: 1, 2 -> 7.0

X: 2, 3 -> 7.0

X: 3, 4 -> 9.0

X: 4, 5 -> 11.0
 X: 5, 6 -> 11.0
 X: 1, 2 -> 7.0
 X: 2, 3 -> 7.0
 X: 3, 4 -> 9.0
 X: 4, 5 -> 11.0
 X: 5, 6 -> 11.0
 X: 1, 2 -> 7.0
 X: 2, 3 -> 7.0
 X: 3, 4 -> 9.0
 X: 4, 5 -> 11.0
 X: 5, 6 -> 11.0

2. Berikut grafik perbandingan nilai asli dan nilai prediksi pada soal 1:



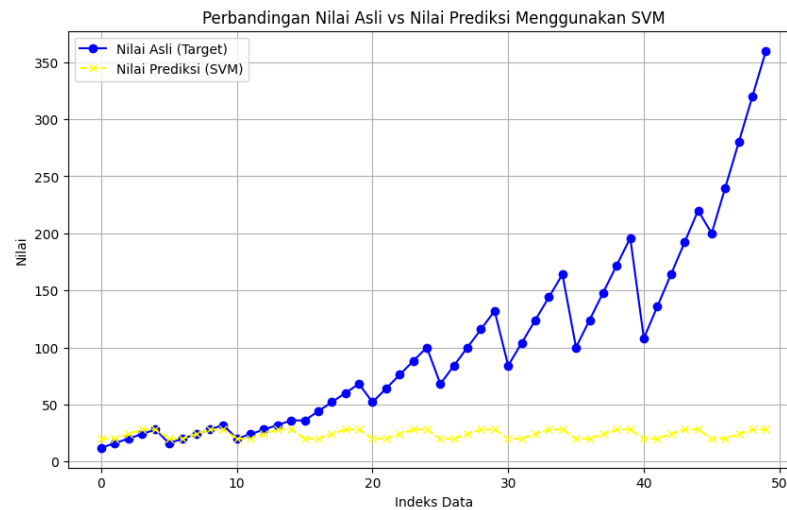
Dari grafik ini, terlihat bahwa nilai asli (target) yang digambarkan dengan garis biru menunjukkan tren yang meningkat secara tajam, terutama di bagian akhir data. Sementara itu, nilai prediksi yang dihasilkan oleh model SVM, ditunjukkan dengan garis kuning, terlihat stabil dan tidak mengikuti pola peningkatan dari nilai asli. Perbedaan antara kedua garis ini semakin besar seiring bertambahnya indeks data, yang menunjukkan bahwa model SVM gagal menangkap pola kenaikan yang ada di nilai asli. Hal ini mengindikasikan bahwa prediksi model kurang akurat.

3. Hasil prediksi integral trapezoid menggunakan *Support Vector Machine* pada beberapa persamaan dengan nilai $a = i+2$ dan $b = i+4$:

Hasil prediksi:
 X: 2, 4 -> 20.0
 X: 3, 5 -> 20.0
 X: 4, 6 -> 24.0
 X: 5, 7 -> 28.0
 X: 6, 8 -> 28.0
 X: 2, 4 -> 20.0
 X: 3, 5 -> 20.0
 X: 4, 6 -> 24.0
 X: 5, 7 -> 28.0
 X: 6, 8 -> 28.0

X: 6, 8 -> 28.0

4. Berikut grafik perbandingan nilai asli dan nilai prediksi pada soal 3:



Kurva yang tergambar pada grafik ini, sama seperti pada grafik no 2, terlihat bahwa nilai asli (target) yang digambarkan dengan garis biru menunjukkan tren yang meningkat secara tajam, terutama di bagian akhir data. Sementara itu, nilai prediksi yang dihasilkan oleh model SVM, ditunjukkan dengan garis kuning, terlihat stabil dan tidak mengikuti pola peningkatan dari nilai asli. Perbedaan antara kedua garis ini semakin besar seiring bertambahnya indeks data, yang menunjukkan bahwa model SVM gagal menangkap pola kenaikan yang ada di nilai asli. Hal ini mengindikasikan bahwa prediksi model kurang akurat.

5. Untuk menentukan integral trapezoid dan prediksi nilai integral trapezoid menggunakan metode *Support Vector Machine*, terdapat 2 kode program yang di gunakan:

```
def Trapezoid(a,b,f):
    """
    Fungsi untuk mencari Integral Trapezoid dengan mengganti nilai
    a = batas atas
    dan
    b = batas bawah,
    serta
    f = yang akan diintegralkan
    """
    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)) #Rumus Trapezoid
        return integral
    integral = trapezoid(f,a,b,n)
    print(a,",",b,",",round(integral,2))
```

```

# Melakukan looping untuk membuat database dari beberapa soal integral
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)

```

Program ini dibuat untuk menghitung nilai integral secara numerik menggunakan metode Trapezoid pada berbagai fungsi linear. Di dalam program, terdapat fungsi bernama Trapezoid yang menerima tiga masukan: batas bawah a, batas atas b, dan fungsi matematika f yang ingin diintegrasikan. Di dalam fungsi ini, terdapat fungsi tambahan bernama trapezoid, yang bertugas melakukan perhitungan integral menggunakan rumus metode Trapezoid dengan membagi interval antara a dan b menjadi beberapa segmen kecil, sebanyak n.

Rumus Trapezoid digunakan dengan menghitung panjang segmen kecil h, lalu menjumlahkan nilai fungsi di titik-titik tertentu dalam segmen tersebut. Nilai di titik tengah dari semua segmen dijumlahkan, kemudian ditambahkan dengan nilai fungsi di batas awal dan akhir. Setelah semua dihitung, nilai integral diperoleh sesuai rumus metode Trapezoid dan hasilnya dikembalikan.

Program juga memiliki beberapa perulangan untuk menghitung integral dari berbagai fungsi linear yang berbeda. Setiap fungsi memiliki bentuk seperti $2x$, $2x+2$, $2x+4$, dan seterusnya. Nilai batas bawah (a) dan batas atas (b) diatur agar meningkat pada setiap iterasi, dimulai dari 1 hingga 6 untuk setiap fungsi. Hasil integral dari setiap kombinasi fungsi dan batas dihitung, dibulatkan hingga dua desimal, dan ditampilkan.

```

# 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/Modul 11/Trapezoid 1.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 SVM
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: {X.iloc[i, 0]}, {X.iloc[i, 1]} -> {pred}")

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

```

Program ini dibuat untuk membaca data dari file, melatih model SVM, lalu membandingkan hasil prediksi model dengan nilai aslinya menggunakan grafik. Awalnya, program ini mengimpor beberapa pustaka penting seperti numpy, pandas, sklearn untuk model SVM, dan matplotlib untuk membuat grafik. Karena datanya disimpan di Google Drive, program melakukan mounting ke Google Drive supaya file bisa diakses.

Data yang ada di file bernama Trapezoid 1.txt dibaca ke dalam DataFrame menggunakan pandas. Data ini kemudian dipisah menjadi dua bagian: bagian input (X), yaitu kolom a dan b, serta bagian target (y), yaitu nilai yang mau diprediksi. Setelah itu, model SVM dibuat menggunakan fungsi dari sklearn, lalu model ini dilatih dengan data X dan y.

Setelah model selesai dilatih, program menggunakan data yang sama untuk menghasilkan prediksi. Hasil prediksi ini kemudian ditampilkan satu per satu bersama dengan nilai inputnya supaya mudah dilihat.

Terakhir, program membuat grafik untuk membandingkan nilai asli (target) dengan nilai prediksi yang dihasilkan oleh model SVM. Nilai asli digambarkan dengan garis biru, sedangkan hasil prediksi ditunjukkan dengan garis kuning yang memiliki tanda silang. Grafik ini diberi label, judul, dan grid supaya lebih jelas. Inti dari program ini adalah melihat apakah model SVM sudah bisa memprediksi nilai dengan baik berdasarkan data yang tersedia.