

PRAKTIKUM FISIKA KOMPUTASI SUPPORT VECTOR MACHINE (SVM)

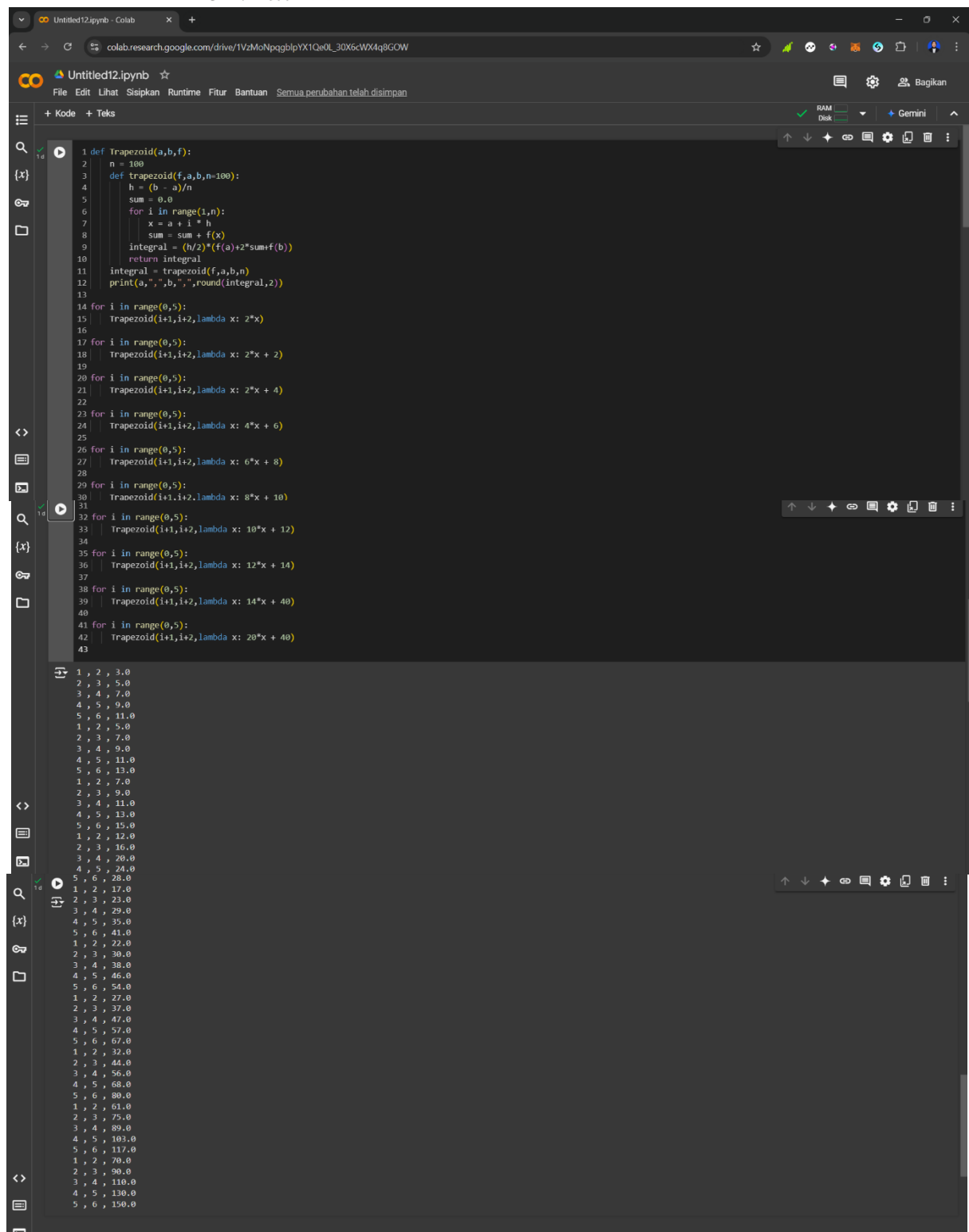
Oleh:

Eneng Yulia Pebryanti

NIM 1227030013

1. Buat prediksi integral trapezoid menggunakan Support Vector Machine pada persamaan berikut ini:

Fungsi pertama: $f(x) = 2x$
Fungsi kedua: $f(x) = 2x + 2$
Fungsi ketiga: $f(x) = 2x + 4$
Fungsi keempat: $f(x) = 4x + 6$
Fungsi kelima: $f(x) = 6x + 8$
Fungsi keenam: $f(x) = 8x + 10$
Fungsi ketujuh: $f(x) = 10x + 12$
Fungsi kedelapan: $f(x) = 12x + 14$
Fungsi kesembilan: $f(x) = 14x + 16$
Fungsi kesepuluh: $f(x) = 20x + 40$



```
1 def Trapezoid(a,b,f):
2     n = 100
3     def trapezoid(f,a,b,n=100):
4         h = (b - a)/n
5         sum = 0.0
6         for i in range(1,n):
7             x = a + i * h
8             sum = sum + f(x)
9         integral = (h/2)*(f(a)+2*sum+f(b))
10        return integral
11    integral = trapezoid(f,a,b,n)
12    print(a," ",b," ",round(integral,2))
13
14 for i in range(0,5):
15     Trapezoid(i+1,i+2,lambda x: 2*x)
16
17 for i in range(0,5):
18     Trapezoid(i+1,i+2,lambda x: 2*x + 2)
19
20 for i in range(0,5):
21     Trapezoid(i+1,i+2,lambda x: 2*x + 4)
22
23 for i in range(0,5):
24     Trapezoid(i+1,i+2,lambda x: 4*x + 6)
25
26 for i in range(0,5):
27     Trapezoid(i+1,i+2,lambda x: 6*x + 8)
28
29 for i in range(0,5):
30     Trapezoid(i+1,i+2,lambda x: 8*x + 10)
31
32 for i in range(0,5):
33     Trapezoid(i+1,i+2,lambda x: 10*x + 12)
34
35 for i in range(0,5):
36     Trapezoid(i+1,i+2,lambda x: 12*x + 14)
37
38 for i in range(0,5):
39     Trapezoid(i+1,i+2,lambda x: 14*x + 16)
40
41 for i in range(0,5):
42     Trapezoid(i+1,i+2,lambda x: 20*x + 40)
43
```

1, 2, 3.0
2, 3, 5.0
3, 4, 7.0
4, 5, 9.0
5, 6, 11.0
1, 2, 5.0
2, 3, 7.0
3, 4, 9.0
4, 5, 11.0
5, 6, 13.0
1, 2, 7.0
2, 3, 9.0
3, 4, 11.0
4, 5, 13.0
5, 6, 15.0
1, 2, 12.0
2, 3, 16.0
3, 4, 20.0
4, 5, 24.0
5, 6, 28.0
1, 2, 17.0
2, 3, 23.0
3, 4, 29.0
4, 5, 35.0
5, 6, 41.0
1, 2, 22.0
2, 3, 30.0
3, 4, 38.0
4, 5, 46.0
5, 6, 54.0
1, 2, 27.0
2, 3, 37.0
3, 4, 47.0
4, 5, 57.0
5, 6, 67.0
1, 2, 32.0
2, 3, 44.0
3, 4, 56.0
4, 5, 68.0
5, 6, 80.0
1, 2, 51.0
2, 3, 75.0
3, 4, 89.0
4, 5, 103.0
5, 6, 117.0
1, 2, 70.0
2, 3, 90.0
3, 4, 110.0
4, 5, 130.0
5, 6, 150.0

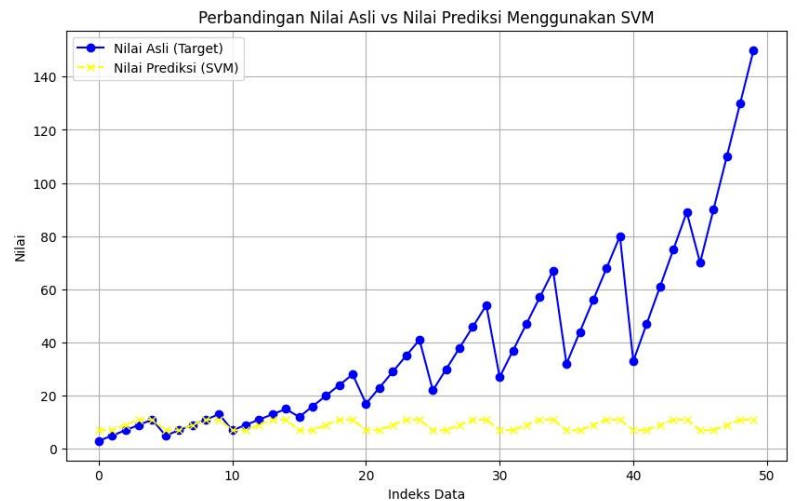
2. Analisis grafik perbandingan nilai asli dan nilai prediksi pada soal nomor 1!

```
Untitled12.ipynb ☆
File Edit Lihat Simpan Runtime Fitur Bantuan Semua perubahan telah disimpan
+ Kode + Teks

1 # Import library yang diperlukan
2 import numpy as np
3 import pandas as pd
4 from sklearn import svm
5 from google.colab import drive
6 import matplotlib.pyplot as plt
7
8 # Mount Google Drive
9 drive.mount('/content/drive')
10
11 # Path ke file Database.txt di Google Drive
12 file_path = '/content/drive/MyDrive/trapezoid.txt' # Ganti dengan path sesuai lokasi file Anda di Google Drive
13
14 # Membaca data dari file (periksa delimiter, misalnya koma atau tab)
15 Database = pd.read_csv(file_path, sep=',', header=0) # Pastikan delimiter sesuai dengan file
16
17 # Cek isi data untuk memastikan format benar
18 print(Database.head())
19
20 # x = data, y = target
21 X = Database[['a', 'b']] # Pastikan kolom sesuai dengan nama yang ada di file
22 y = Database['target'] # Pastikan kolom 'target' ada di file
23
24 # Membuat dan melatih model SVM
25 clf = svm.SVC()
26 clf.fit(X.values, y)
27
28 # Melakukan prediksi
29 y_pred = clf.predict(X.values)
30
31 # Menampilkan hasil prediksi
32 print("Hasil prediksi:")
33 for i, pred in enumerate(y_pred):
34     print(f"({X.iloc[i, 0]}, {X.iloc[i, 1]}) : {pred}")
35
36 # Membuat plot perbandingan nilai asli dengan nilai prediksi
37 plt.figure(figsize=(10, 6))
38
39 # Periksa panjang data sebelum plotting untuk menghindari error
40 plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)', color='blue')
41 plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi (SVM)', color='yellow')
42
43 # Menambahkan label dan judul
44 plt.xlabel('Indeks Data')
45 plt.ylabel('Nilai')
46 plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
47 plt.legend()
48 plt.grid()
49 plt.show()
50
```

```

a b Target
0 1 2 3.0
1 2 3 5.0
2 3 4 7.0
3 4 5 9.0
4 5 6 11.0
Hasil prediksi:
1, 2, 7.0
2, 3, 7.0
3, 4, 9.0
4, 5, 11.0
5, 6, 11.0
1, 2, 7.0
2, 3, 7.0
3, 4, 9.0
4, 5, 11.0
5, 6, 11.0
1, 2, 7.0
2, 3, 7.0
3, 4, 9.0
4, 5, 11.0
5, 6, 11.0
1, 2, 7.0
2, 3, 7.0
3, 4, 9.0
4, 5, 11.0
5, 6, 11.0
1, 2, 7.0
```



Pada grafik perbandingan antara nilai asli dan nilai prediksi yang dihasilkan oleh metode Support Vector Machine (SVM), terdapat dua garis dengan warna yang berbeda. Garis biru menggambarkan nilai asli atau nilai target yang sebenarnya, sedangkan garis kuning dengan tanda silang menunjukkan nilai prediksi yang dihasilkan oleh model Support Vector Machine (SVM). Di awal grafik, kedua garis ini cukup berdekatan, yang berarti prediksi model SVM cukup akurat dan mampu mengikuti pola nilai asli dengan baik. Namun, ketika grafik berlanjut ke bagian tengah hingga akhir data, ada perbedaan yang mulai terlihat. Nilai asli terus mengalami kenaikan secara bertahap, sedangkan nilai prediksi kadang tertinggal atau tidak sepenuhnya sejajar dengan pola nilai asli. Ini menunjukkan bahwa model SVM belum sepenuhnya menangkap pola data yang lebih kompleks atau variasi yang terjadi diengah dan akhir grafik.

3. Kerjakan soal nomor 1 dengan nilai $a = i+2$ dan $b = i+4$!

```

1 def trapezoid(a,b,f):
2     n = 100
3     def trapezoid(f,a,b,n=100):
4         h = (b - a)/n
5         sum = 0.0
6         for i in range(1,n):
7             x = a + i * h
8             sum = sum + f(x)
9         integral = (n/2)*(f(a)+2*sum+f(b))
10        return integral
11    integral = trapezoid(f,a,b,n)
12    print(a, " , ", b, " , ",round(integral,2))
13
14 for i in range(0,5):
15     trapezoid(i+2,i+4,lambda x: 2*x)
16
17 for i in range(0,5):
18     trapezoid(i+2,i+4,lambda x: 2*x + 2)
19
20 for i in range(0,5):
21     trapezoid(i+2,i+4,lambda x: 2*x + 4)
22
23 for i in range(0,5):
24     trapezoid(i+2,i+4,lambda x: 4*x + 6)
25
26 for i in range(0,5):
27     trapezoid(i+2,i+4,lambda x: 6*x + 8)
28
29 for i in range(0,5):
30     trapezoid(i+2,i+4,lambda x: 8*x + 10)
31
32 for i in range(0,5):
33     trapezoid(i+2,i+4,lambda x: 10*x + 12)
34
35 for i in range(0,5):
36     trapezoid(i+2,i+4,lambda x: 12*x + 14)
37
38 for i in range(0,5):
39     trapezoid(i+2,i+4,lambda x: 14*x + 16)
40
41 for i in range(0,5):
42     trapezoid(i+2,i+4,lambda x: 20*x + 40)
43

```

Output of the code:

```

2 , 4 , 12.0
3 , 5 , 16.0
4 , 6 , 20.0
5 , 7 , 24.0
6 , 8 , 28.0
2 , 4 , 16.0
3 , 5 , 20.0
4 , 6 , 24.0
5 , 7 , 28.0
2 , 4 , 20.0
3 , 5 , 24.0
4 , 6 , 28.0
5 , 7 , 32.0
2 , 4 , 24.0
3 , 5 , 28.0
4 , 6 , 32.0
5 , 7 , 36.0
2 , 4 , 28.0
3 , 5 , 32.0
4 , 6 , 36.0
5 , 7 , 40.0
2 , 4 , 32.0
3 , 5 , 36.0
4 , 6 , 40.0
5 , 7 , 44.0
2 , 4 , 36.0
3 , 5 , 40.0
4 , 6 , 44.0
5 , 7 , 48.0
2 , 4 , 40.0
3 , 5 , 44.0
4 , 6 , 48.0
5 , 7 , 52.0
2 , 4 , 44.0
3 , 5 , 48.0
4 , 6 , 52.0
5 , 7 , 56.0
2 , 4 , 48.0
3 , 5 , 52.0
4 , 6 , 56.0
5 , 7 , 60.0
2 , 4 , 52.0
3 , 5 , 56.0
4 , 6 , 60.0
5 , 7 , 64.0
2 , 4 , 56.0
3 , 5 , 60.0
4 , 6 , 64.0
5 , 7 , 68.0
2 , 4 , 60.0
3 , 5 , 64.0
4 , 6 , 68.0
5 , 7 , 72.0
2 , 4 , 64.0
3 , 5 , 68.0
4 , 6 , 72.0
5 , 7 , 76.0
2 , 4 , 68.0
3 , 5 , 72.0
4 , 6 , 76.0
5 , 7 , 80.0
2 , 4 , 72.0
3 , 5 , 76.0
4 , 6 , 80.0
5 , 7 , 84.0
2 , 4 , 76.0
3 , 5 , 80.0
4 , 6 , 84.0
5 , 7 , 88.0
2 , 4 , 80.0
3 , 5 , 84.0
4 , 6 , 88.0
5 , 7 , 92.0
2 , 4 , 84.0
3 , 5 , 88.0
4 , 6 , 92.0
5 , 7 , 96.0
2 , 4 , 88.0
3 , 5 , 92.0
4 , 6 , 96.0
5 , 7 , 100.0
2 , 4 , 92.0
3 , 5 , 96.0
4 , 6 , 100.0
5 , 7 , 104.0
2 , 4 , 96.0
3 , 5 , 100.0
4 , 6 , 104.0
5 , 7 , 108.0
2 , 4 , 100.0
3 , 5 , 104.0
4 , 6 , 108.0
5 , 7 , 112.0
2 , 4 , 104.0
3 , 5 , 108.0
4 , 6 , 112.0
5 , 7 , 116.0
2 , 4 , 108.0
3 , 5 , 112.0
4 , 6 , 116.0
5 , 7 , 120.0
2 , 4 , 112.0
3 , 5 , 116.0
4 , 6 , 120.0
5 , 7 , 124.0
2 , 4 , 116.0
3 , 5 , 120.0
4 , 6 , 124.0
5 , 7 , 128.0
2 , 4 , 120.0
3 , 5 , 124.0
4 , 6 , 128.0
5 , 7 , 132.0
2 , 4 , 124.0
3 , 5 , 128.0
4 , 6 , 132.0
5 , 7 , 136.0
2 , 4 , 128.0
3 , 5 , 132.0
4 , 6 , 136.0
5 , 7 , 140.0
2 , 4 , 132.0
3 , 5 , 136.0
4 , 6 , 140.0
5 , 7 , 144.0
2 , 4 , 136.0
3 , 5 , 140.0
4 , 6 , 144.0
5 , 7 , 148.0
2 , 4 , 140.0
3 , 5 , 144.0
4 , 6 , 148.0
5 , 7 , 152.0
2 , 4 , 144.0
3 , 5 , 148.0
4 , 6 , 152.0
5 , 7 , 156.0
2 , 4 , 148.0
3 , 5 , 152.0
4 , 6 , 156.0
5 , 7 , 160.0
2 , 4 , 152.0
3 , 5 , 156.0
4 , 6 , 160.0
5 , 7 , 164.0
2 , 4 , 156.0
3 , 5 , 160.0
4 , 6 , 164.0
5 , 7 , 168.0
2 , 4 , 160.0
3 , 5 , 164.0
4 , 6 , 168.0
5 , 7 , 172.0
2 , 4 , 164.0
3 , 5 , 168.0
4 , 6 , 172.0
5 , 7 , 176.0
2 , 4 , 168.0
3 , 5 , 172.0
4 , 6 , 176.0
5 , 7 , 180.0
2 , 4 , 172.0
3 , 5 , 176.0
4 , 6 , 180.0
5 , 7 , 184.0
2 , 4 , 176.0
3 , 5 , 180.0
4 , 6 , 184.0
5 , 7 , 188.0
2 , 4 , 180.0
3 , 5 , 184.0
4 , 6 , 188.0
5 , 7 , 192.0
2 , 4 , 184.0
3 , 5 , 188.0
4 , 6 , 192.0
5 , 7 , 196.0
2 , 4 , 188.0
3 , 5 , 192.0
4 , 6 , 196.0
5 , 7 , 200.0
2 , 4 , 192.0
3 , 5 , 196.0
4 , 6 , 200.0
5 , 7 , 204.0
2 , 4 , 196.0
3 , 5 , 200.0
4 , 6 , 204.0
5 , 7 , 208.0
2 , 4 , 200.0
3 , 5 , 204.0
4 , 6 , 208.0
5 , 7 , 212.0
2 , 4 , 204.0
3 , 5 , 208.0
4 , 6 , 212.0
5 , 7 , 216.0
2 , 4 , 208.0
3 , 5 , 212.0
4 , 6 , 216.0
5 , 7 , 220.0
2 , 4 , 212.0
3 , 5 , 216.0
4 , 6 , 220.0
5 , 7 , 224.0
2 , 4 , 216.0
3 , 5 , 220.0
4 , 6 , 224.0
5 , 7 , 228.0
2 , 4 , 220.0
3 , 5 , 224.0
4 , 6 , 228.0
5 , 7 , 232.0
2 , 4 , 224.0
3 , 5 , 228.0
4 , 6 , 232.0
5 , 7 , 236.0
2 , 4 , 228.0
3 , 5 , 232.0
4 , 6 , 236.0
5 , 7 , 240.0
2 , 4 , 232.0
3 , 5 , 236.0
4 , 6 , 240.0
5 , 7 , 244.0
2 , 4 , 236.0
3 , 5 , 240.0
4 , 6 , 244.0
5 , 7 , 248.0
2 , 4 , 240.0
3 , 5 , 244.0
4 , 6 , 248.0
5 , 7 , 252.0
2 , 4 , 244.0
3 , 5 , 248.0
4 , 6 , 252.0
5 , 7 , 256.0
2 , 4 , 248.0
3 , 5 , 252.0
4 , 6 , 256.0
5 , 7 , 260.0
2 , 4 , 252.0
3 , 5 , 256.0
4 , 6 , 260.0
5 , 7 , 264.0
2 , 4 , 256.0
3 , 5 , 260.0
4 , 6 , 264.0
5 , 7 , 268.0
2 , 4 , 260.0
3 , 5 , 264.0
4 , 6 , 268.0
5 , 7 , 272.0
2 , 4 , 264.0
3 , 5 , 268.0
4 , 6 , 272.0
5 , 7 , 276.0
2 , 4 , 268.0
3 , 5 , 272.0
4 , 6 , 276.0
5 , 7 , 280.0
2 , 4 , 272.0
3 , 5 , 276.0
4 , 6 , 280.0
5 , 7 , 284.0
2 , 4 , 276.0
3 , 5 , 280.0
4 , 6 , 284.0
5 , 7 , 288.0
2 , 4 , 280.0
3 , 5 , 284.0
4 , 6 , 288.0
5 , 7 , 292.0
2 , 4 , 284.0
3 , 5 , 288.0
4 , 6 , 292.0
5 , 7 , 296.0
2 , 4 , 288.0
3 , 5 , 292.0
4 , 6 , 296.0
5 , 7 , 300.0
2 , 4 , 292.0
3 , 5 , 296.0
4 , 6 , 300.0
5 , 7 , 304.0
2 , 4 , 296.0
3 , 5 , 300.0
4 , 6 , 304.0
5 , 7 , 308.0
2 , 4 , 300.0
3 , 5 , 304.0
4 , 6 , 308.0
5 , 7 , 312.0
2 , 4 , 304.0
3 , 5 , 308.0
4 , 6 , 312.0
5 , 7 , 316.0
2 , 4 , 308.0
3 , 5 , 312.0
4 , 6 , 316.0
5 , 7 , 320.0
2 , 4 , 312.0
3 , 5 , 316.0
4 , 6 , 320.0
5 , 7 , 324.0
2 , 4 , 316.0
3 , 5 , 320.0
4 , 6 , 324.0
5 , 7 , 328.0
2 , 4 , 320.0
3 , 5 , 324.0
4 , 6 , 328.0
5 , 7 , 332.0
2 , 4 , 324.0
3 , 5 , 328.0
4 , 6 , 332.0
5 , 7 , 336.0
2 , 4 , 328.0
3 , 5 , 332.0
4 , 6 , 336.0
5 , 7 , 340.0
2 , 4 , 332.0
3 , 5 , 336.0
4 , 6 , 340.0
5 , 7 , 344.0
2 , 4 , 336.0
3 , 5 , 340.0
4 , 6 , 344.0
5 , 7 , 348.0
2 , 4 , 340.0
3 , 5 , 344.0
4 , 6 , 348.0
5 , 7 , 352.0
2 , 4 , 344.0
3 , 5 , 348.0
4 , 6 , 352.0
5 , 7 , 356.0
2 , 4 , 348.0
3 , 5 , 352.0
4 , 6 , 356.0
5 , 7 , 360.0

```

4. Analisis grafik perbandingan nilai asli dan nilai prediksi pada soal nomor 3!

```
untitled12.ipynb ☆
Edit Lihat Sisipkan Runtime Fitur Bantuan

+ Teks

1 # Import library yang diperlukan
2 import numpy as np
3 import pandas as pd
4 from sklearn import svm
5 from google.colab import drive
6 import matplotlib.pyplot as plt
7
8 # Mount Google Drive
9 drive.mount('/content/drive')
10
11 # Path ke file Database.txt di Google Drive
12 file_path = '/content/drive/MyDrive/Trapezoid3.txt' # Sesuaikan dengan lokasi
   file Anda di Google Drive
13
14 # Membaca data dari file (periksa delimiter, misalnya koma atau tab)
15 Database = pd.read_csv(file_path, sep=",", header=0) # Pastikan delimiter
   sesuai dengan file
16
17 # Cek isi data untuk memastikan format benar
18 print(Database.head())
19
20 # x = Data, y = Target
21 X = Database[['a', 'b']] # Pastikan kolom sesuai dengan nama yang ada di file
22 y = Database['Target'] # Pastikan kolom 'Target' ada di file
23
24 # Membuat dan melatih model SVM
25 clf = svm.SVC()
26 clf.fit(X.values, y)
27
28 # Melakukan prediksi
29 y_pred = clf.predict(X.values)
30
31 # Menampilkan hasil prediksi
32 print("Hasil prediksi:")
33 for i, pred in enumerate(y_pred):
34     print(f"{X.iloc[i, 0]}, {X.iloc[i, 1]}, {pred}")
35
36 # Membuat plot perbandingan nilai asli dengan nilai prediksi
37 plt.figure(figsize=(10, 6))
38
39 # Periksa panjang data sebelum plotting untuk menghindari error
40 plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)', color='blue')
41 plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi (SVM)',
   color='yellow')
42
43 # Menambahkan label dan judul
44 plt.xlabel('Indeks Data')
45 plt.ylabel('Nilai')
46 plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
47 plt.legend()
48 plt.grid()
49 plt.show()
50
```



Pada grafik dapat dilihat dua garis yang digunakan untuk membandingkan nilai asli (ditampilkan dalam warna biru dengan titik-titik) dan nilai prediksi (ditampilkan dalam warna kuning dengan tanda silang). Sama seperti soal nomor 2, grafik ini menunjukkan perbedaan yang cukup menarik untuk diamati. Di awal data, garis biru dan kuning terlihat cukup dekat satu sama lain. Hal ini berarti model SVM mampu memberikan prediksi yang mendekati nilai asli, terutama untuk data yang cenderung stabil atau berubah secara bertahap. Namun, ketika data mendekati akhir, nilai asli mulai naik secara signifikan garis biru menanjak nilai prediksi tetap cenderung datar atau hanya naik sedikit.

5. Jelaskan algoritma program integral trapezoid dan prediksi nilai integral trapezoid menggunakan metode Support Vector Machine!

Metode trapezoid adalah pendekatan numerik sederhana untuk menghitung integral suatu fungsi. Cara kerjanya bisa diibaratkan seperti membagi area di bawah kurva menjadi beberapa trapezium kecil. Masing-masing trapezium dihitung luasnya dengan rumus dasar yang melibatkan tinggi dan lebar trapezium. Prosesnya dimulai dengan menentukan interval awal dan akhir yang ingin dihitung. Interval tersebut kemudian dibagi menjadi beberapa bagian kecil yang panjangnya sama. Pada setiap titik pembagian, nilai fungsi dihitung, termasuk pada batas awal dan akhir. Hasilnya, nilai-nilai ini dijumlahkan dengan bobot tertentu untuk menghasilkan perkiraan nilai integral. Semakin kecil intervalnya, semakin banyak trapezium yang digunakan, dan semakin akurat hasil yang diperoleh. Sementara itu, metode Support Vector Machine (SVM) digunakan untuk memprediksi nilai integral berdasarkan pola data sebelumnya. Cara kerjanya berbeda dengan metode numerik. SVM adalah model pembelajaran mesin yang memanfaatkan data pelatihan untuk mengenali pola hubungan antara variabel input (seperti titik-titik interval) dan output (nilai integral). Model ini dilatih menggunakan data yang sudah diketahui hasil integralnya, sehingga mampu "belajar" dari pola tersebut. Setelah model terlatih, ia dapat memprediksi nilai integral untuk data baru tanpa perlu melakukan perhitungan trapezoid manual. Keuntungan dari metode SVM adalah kecepatan prediksi, terutama jika datanya sangat banyak. Namun, akurasi prediksi sangat bergantung pada kualitas data pelatihan dan parameter model. Jika pola data pelatihan tidak mencakup variasi yang cukup atau parameter model tidak diatur dengan baik, hasil prediksinya bisa jauh dari kenyataan.