

PRAKTIKUM FISIKA KOMPUTASI
MODUL 11 "SUPPORT VECTOR MACHINE"

A. Kode Program Tugas No.1

```
# Fungsi untuk mencari integral menggunakan metode Trapezoid
def trapezoid(a, b, f):
    # Fungsi Trapezoid
    n = 100 # Jumlah partisi
    h = (b - a) / n
    sum = 0.0
    for i in range(1, n):
        x = a + i * h
        sum += f(x)
    integral = (h / 2) * (f(a) + 2 * sum + f(b)) # Rumus Trapezoid
    print(a, ",", b, ",", round(integral, 2)) # Menampilkan hasil integral
    return integral

# Daftar fungsi menggunakan lambda
functions = [
    lambda x: 2 * x,          #Fungsi pertama:f(x) = 2x
    lambda x: 2 * x + 2,      #Fungsi kedua:f(x) = 2x + 2
    lambda x: 4 * x + 4,      #Fungsi ketiga:f(x) = 4x + 4
    lambda x: 4 * x + 6,      #Fungsi keempat:f(x) = 4x + 6
    lambda x: 6 * x + 8,      #Fungsi kelima:f(x) = 6x + 8
    lambda x: 8 * x + 10,     #Fungsi keenam:f(x) = 8x + 10
    lambda x: 10 * x + 12,    #Fungsi ketujuh:f(x) = 10x + 12
    lambda x: 12 * x + 14,    #Fungsi kedelapan:f(x) = 12x + 14
    lambda x: 14 * x + 12,    #Fungsi kesembilan:f(x) = 14x + 12
    lambda x: 20 * x + 40     #Fungsi kesepuluh:f(x) = 20x + 40
]

# Looping untuk menghitung integral masing-masing fungsi
for i, func in enumerate(functions):
    a = i + 1 # Batas bawah
    b = i + 2 # Batas atas
    trapezoid(a, b, func) # Menghitung integral
```

dengan hasil prediksinya:

```
a,b,Target
1 , 2 , 3.0
2 , 3 , 7.0
3 , 4 , 18.0
4 , 5 , 24.0
5 , 6 , 41.0
6 , 7 , 62.0
7 , 8 , 87.0
8 , 9 , 116.0
9 , 10 , 145.0
10 , 11 , 250.0
```

```

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

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

# Path ke file Function.txt di Google Drive
file_path = '/content/drive/My Drive/Pertamax.txt' # Sesuaikan
dengan path file Anda

# 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.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 (Tugas 1)')
plt.legend()

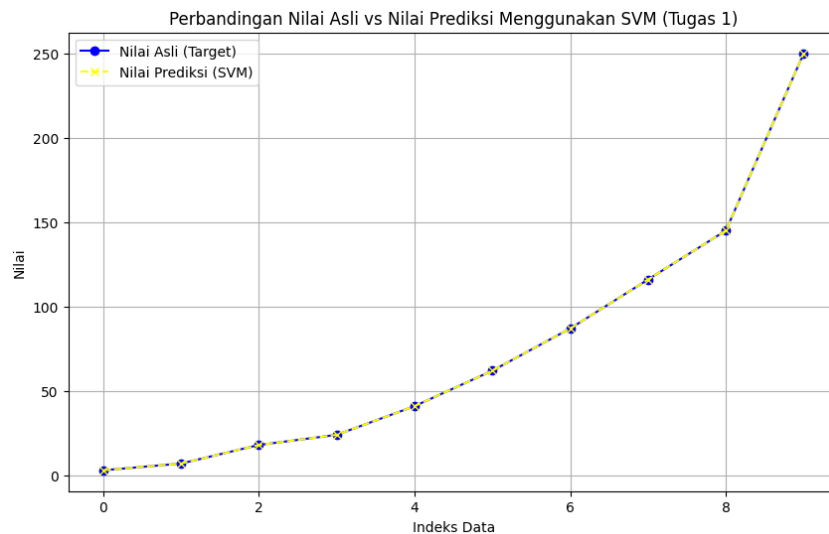
# Menambahkan grid

```

```
plt.grid()
```

```
plt.show()
```

dengan grafik yang dihasilkan:



Berdasarkan grafik yang dihasilkan tersebut, dapat terlihat bahwa untuk setiap fungsi, perbandingan antara nilai asli dan nilai prediksi dengan menggunakan metode SVM sangat akurat. Interpretasi yang terlihat adalah sumbu x sebagai indeks data, sumbu y sebagai variabel nilai, garis biru sebagai nilai asli, dan garis kuning putus-putus sebagai nilai prediksi menggunakan SVM. Dari grafik tersebut terlihat bahwa besar nilai akan meningkat sesuai dengan meningkatnya indeks data.

B. Kode Program Tugas No.3

```
# Fungsi untuk mencari integral menggunakan metode Trapezoid
def trapezoid(a, b, f):
    # Fungsi Trapezoid
    n = 100 # Jumlah partisi
    h = (b - a) / n
    sum = 0.0
    for i in range(1, n):
        x = a + i * h
        sum += f(x)
    integral = (h / 2) * (f(a) + 2 * sum + f(b)) # Rumus Trapezoid
    print(a, ",", b, ",", round(integral, 2)) # Menampilkan hasil integral
    return integral

# Daftar fungsi menggunakan lambda
functions = [
    lambda x: 2 * x, # Fungsi pertama: f(x) = 2x
```

```

lambda x: 2 * x + 2,      #Fungsi kedua:f(x) = 2x + 2
lambda x: 4 * x + 4,      #Fungsi ketiga:f(x) = 4x + 4
lambda x: 4 * x + 6,      #Fungsi keempat:f(x) = 4x + 6
lambda x: 6 * x + 8,      #Fungsi kelima:f(x) = 6x + 8
lambda x: 8 * x + 10,     #Fungsi keenam:f(x) = 8x + 10
lambda x: 10 * x + 12,    #Fungsi ketujuh:f(x) = 10x + 12
lambda x: 12 * x + 14,    #Fungsi kedelapan:f(x) = 12x + 14
lambda x: 14 * x + 12,    #Fungsi kesembilan:f(x) = 14x + 12
lambda x: 20 * x + 40     #Fungsi kesepuluh: f(x) = 20x + 40
]

# Looping untuk menghitung integral masing-masing fungsi
for i, func in enumerate(functions):
    a = i + 2 # Batas bawah
    b = i + 4 # Batas atas
    trapezoid(a, b, func) # Menghitung integral

```

dengan hasil prediksinya:

```

a,b,Target
2 , 4 , 12.0
3 , 5 , 20.0
4 , 6 , 48.0
5 , 7 , 60.0
6 , 8 , 100.0
7 , 9 , 148.0
8 , 10 , 204.0
9 , 11 , 268.0
10 , 12 , 332.0
11 , 13 , 560.0

```

```

# 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/Ketiga.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.iloc[i, 0]}, {X.iloc[i, 1]}), {pred}")

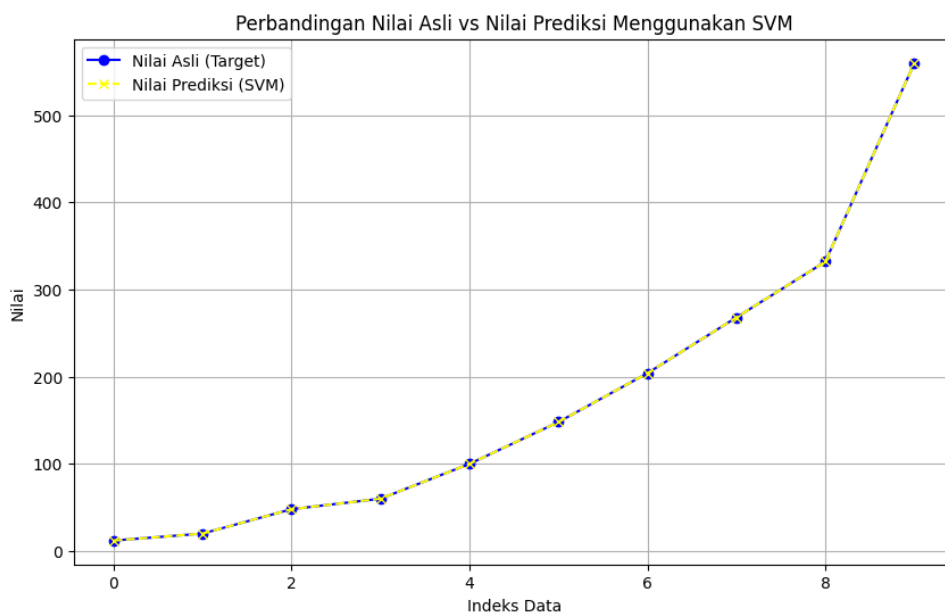
# Membuat plot perbandingan nilai asli dengan nilai prediksi
plt.figure(figsize=(10, 6))
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)',
color='blue')
plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai
Prediksi (SVM)', color='yellow')

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

plt.show()

```

dengan grafik yang dihasilkan:



Berdasarkan grafik yang dihasilkan tersebut, dapat terlihat bahwa untuk setiap fungsi, perbandingan antara nilai asli dan nilai prediksi dengan menggunakan metode SVM sangat akurat. Interpretasi yang terlihat adalah sumbu x sebagai indeks data, sumbu y sebagai variabel nilai, garis biru sebagai nilai asli, dan garis kuning putus-putus sebagai nilai prediksi menggunakan SVM. Dari grafik tersebut terlihat bahwa besar nilai akan meningkat sesuai dengan meningkatnya indeks data. Kurva yang didapatkan pada tugas ketiga ini sama dengan tugas pertama, karena fungsi yang digunakannya juga sama.

C. Algoritma Pemrograman:

I. Menghitung Prediksi Integral Trapezoid

1. Mendefinisikan fungsi trapezoid dengan nilai a (batas atas), b (batas bawah), dan f (prediksi).
2. Mendefinisikan rumus trapezoid dimana batas bawah dikurangi batas akhir kemudian dibagi dengan jumlah partisi (data).
3. Mendefinisikan beberapa fungsi untuk menentukan nilai prediksi integralnya.
4. Menyimpan hasil prediksi yang didapatkan dalam format txt.

II. Memplot nilai prediksi SVM

1. Mengimport beberapa library yang digunakan.
2. Memberikan perintah mount google drive agar bisa memanggil file txt yang telah disimpan dalam drive.
3. Mendefinisikan sumbu x sebagai nilai a dan b, sedangkan sumbu y sebagai target prediksi.
4. Membuat dan melatih model SVM dan kemudian melakukan prediksi.
5. Menampilkan hasil prediksi sesuai dengan perintah yang diberikan kemudian plot grafik untuk nilai asli dan nilai prediksi menggunakan SVM.