

Attala Muflih Gumilang
1227030007
Praktikum Fisika Komputasi

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

```
def Trapezoid(a, b, f):  
    """  
    Fungsi untuk mencari Integral Trapezoid dengan mengganti nilai  
    a = batas atas  
    b = batas bawah,  
    serta f = yang akan diintegalkan  
    """  
    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 += 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)
```

```
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, 33.0
2, 3, 47.0
3, 4, 61.0
4, 5, 75.0
5, 6, 89.0
1, 2, 70.0
2, 3, 90.0
3, 4, 110.0
4, 5, 130.0
5, 6, 150.0
```

```
# 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', force_remount=True)

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

# Membaca data dari file
try:
    Database = pd.read_csv(file_path, sep=",", header=0)
    print("Data berhasil dimuat.")
    print("Kolom yang ditemukan dalam file:", Database.columns)
except Exception as e:
    print(f"Terjadi error saat membaca file: {e}")
    exit()

# Membersihkan nama kolom dari spasi tambahan
Database.columns = Database.columns.str.strip()
print("Nama kolom setelah dibersihkan:", Database.columns)

# Validasi dan sesuaikan nama kolom
required_columns = ['1', '2', '3.0'] # Sesuaikan nama kolom berdasarkan file
if all(col in Database.columns for col in required_columns):
    X = Database[['1', '2']] # Menggunakan kolom fitur
    y = Database['3.0']      # Menggunakan kolom target
    print("Kolom sesuai, data siap diproses.")
else:
```

```

print(f'Kolom yang diperlukan tidak ditemukan. Pastikan file memiliki kolom:
{required_columns}')
exit()

# Membuat dan melatih model SVM
try:
    clf = svm.SVC()
    clf.fit(X.values, y)
    print("Model SVM berhasil dilatih.")
except Exception as e:
    print(f'Terjadi error saat melatih model: {e}')
    exit()

# Melakukan prediksi
try:
    y_pred = clf.predict(X.values)
    print("Hasil prediksi:")
    for i, pred in enumerate(y_pred):
        print(f'X: {X.iloc[i, 0]}, {X.iloc[i, 1]} -> Prediksi: {pred}')
except Exception as e:
    print(f'Terjadi error saat melakukan prediksi: {e}')
    exit()

# Membuat plot perbandingan nilai asli dengan nilai prediksi
try:
    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')
    plt.xlabel('Indeks Data')
    plt.ylabel('Nilai')
    plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
    plt.legend()
    plt.grid()
    plt.show()
except Exception as e:
    print(f'Terjadi error saat membuat plot: {e}')

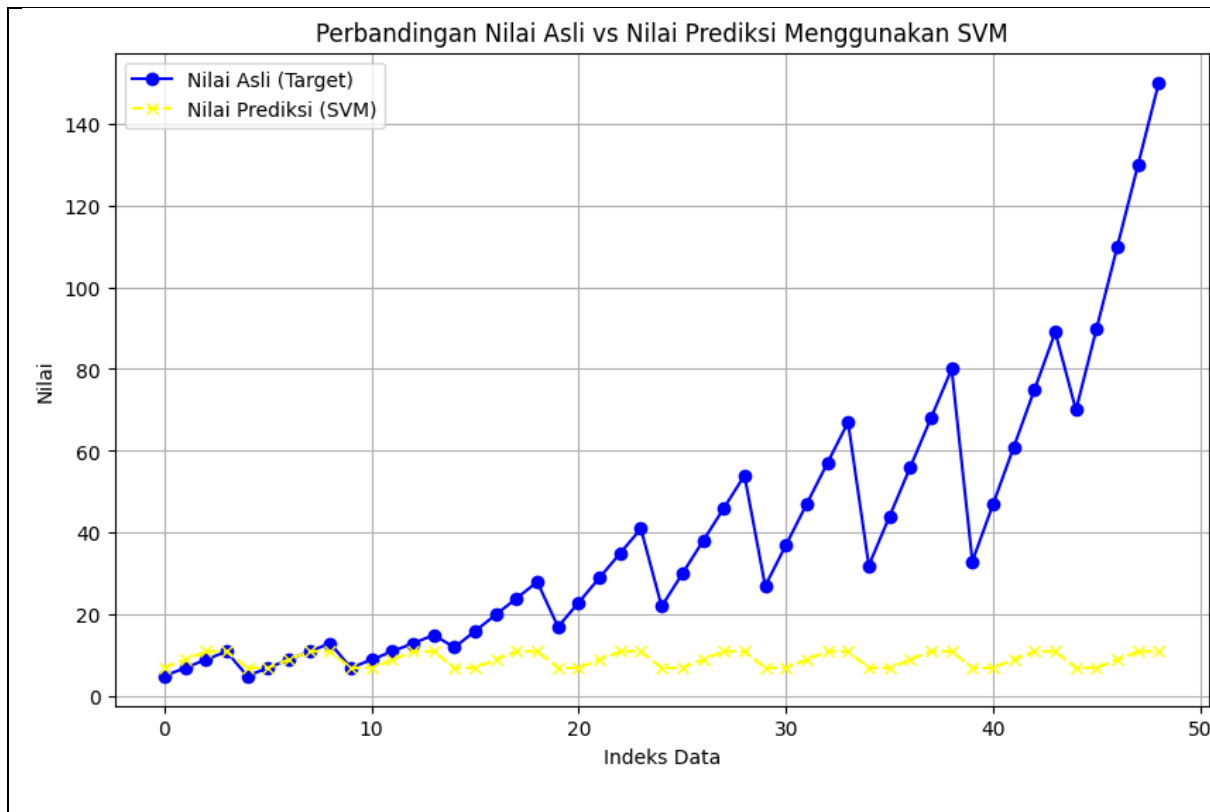
```

```

Mounted at /content/drive
Data berhasil dimuat.
Kolom yang ditemukan dalam file: Index(['1 ', '2 ', '3.0'], dtype='object')
Nama kolom setelah dibersihkan: Index(['1', '2', '3.0'], dtype='object')
Kolom sesuai, data siap diproses.
Model SVM berhasil dilatih.
Hasil prediksi:
X: 2, 3 -> Prediksi: 7.0
X: 3, 4 -> Prediksi: 9.0
X: 4, 5 -> Prediksi: 11.0

```

[illegible]



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

Grafik yang ditampilkan memperlihatkan perbandingan antara nilai asli (target) dan nilai prediksi yang dihasilkan oleh model Support Vector Machine (SVM). Berdasarkan analisis, terlihat adanya perbedaan yang cukup mencolok antara kedua nilai tersebut. Nilai asli, yang direpresentasikan dengan garis biru, menunjukkan pola yang meningkat secara signifikan seiring bertambahnya indeks data, dengan fluktuasi yang cukup tajam di beberapa titik. Sebaliknya, nilai prediksi yang ditunjukkan oleh garis kuning relatif stabil dan tidak menunjukkan tren peningkatan yang serupa.

Perbedaan ini mengindikasikan bahwa model SVM yang digunakan tidak berhasil menangkap pola atau tren yang ada dalam data asli. Model tampaknya hanya menghasilkan prediksi yang berada dalam rentang nilai yang sangat terbatas dan tidak responsif terhadap variasi data asli. Hal ini dapat disebabkan oleh beberapa faktor, seperti kurang representatifnya fitur yang digunakan dalam model, parameter default SVM yang mungkin tidak sesuai dengan karakteristik dataset, atau adanya distribusi data yang kompleks dan non-linear yang tidak dapat ditangani oleh model SVM secara optimal.

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

```
def Trapezoid(a, b, f):
    """
    Fungsi untuk mencari Integral Trapezoid dengan mengganti nilai
    a = batas atas
    b = batas bawah,
```

serta f = yang akan diintegalkan

'''

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 += 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+2, i+4, lambda x: 2*x)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 2*x+2)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 2*x+4)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 4*x+6)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 6*x+8)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 8*x+10)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 10*x+12)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 12*x+14)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 14*x+12)

for i in range(0, 5):

Trapezoid(i+2, i+4, lambda x: 20*x+40)

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

6 , 8 , 32.0

```
2, 4, 20.0
3, 5, 24.0
4, 6, 28.0
5, 7, 32.0
6, 8, 36.0
2, 4, 36.0
3, 5, 44.0
4, 6, 52.0
5, 7, 60.0
6, 8, 68.0
2, 4, 52.0
3, 5, 64.0
4, 6, 76.0
5, 7, 88.0
6, 8, 100.0
2, 4, 68.0
3, 5, 84.0
4, 6, 100.0
5, 7, 116.0
6, 8, 132.0
2, 4, 84.0
3, 5, 104.0
4, 6, 124.0
5, 7, 144.0
6, 8, 164.0
2, 4, 100.0
3, 5, 124.0
4, 6, 148.0
5, 7, 172.0
6, 8, 196.0
2, 4, 108.0
3, 5, 136.0
4, 6, 164.0
5, 7, 192.0
6, 8, 220.0
2, 4, 200.0
3, 5, 240.0
4, 6, 280.0
5, 7, 320.0
6, 8, 360.0
```

```
# 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', force_remount=True)

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

# Membaca data dari file
try:
    Database = pd.read_csv(file_path, sep=",", header=0)
    print("Data berhasil dimuat.")
    print("Kolom yang ditemukan dalam file:", Database.columns)
except Exception as e:
    print(f"Terjadi error saat membaca file: {e}")
    exit()

# Membersihkan nama kolom dari spasi tambahan
Database.columns = Database.columns.str.strip()
print("Nama kolom setelah dibersihkan:", Database.columns)

# Validasi dan sesuaikan nama kolom
required_columns = ['2', '4', '12.0'] # Sesuaikan nama kolom berdasarkan file
if all(col in Database.columns for col in required_columns):
    X = Database[['2', '4']] # Menggunakan kolom fitur
    y = Database['12.0'] # Menggunakan kolom target
else:
    print(f"Kolom yang diperlukan tidak ditemukan. Pastikan file memiliki kolom:
{required_columns}")
    exit()

# Membuat dan melatih model SVM
try:
    clf = svm.SVC()
    clf.fit(X.values, y)
    print("Model SVM berhasil dilatih.")
except Exception as e:
    print(f"Terjadi error saat melatih model: {e}")
    exit()

# Melakukan prediksi
try:
    y_pred = clf.predict(X.values)
    print("Hasil prediksi:")
    for i, pred in enumerate(y_pred):
        print(f"X: {X.iloc[i, 0]}, {X.iloc[i, 1]} -> Prediksi: {pred}")
except Exception as e:
    print(f"Terjadi error saat melakukan prediksi: {e}")
    exit()

# Membuat plot perbandingan nilai asli dengan nilai prediksi
try:

```

```
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')
plt.xlabel('Indeks Data')
plt.ylabel('Nilai')
plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
plt.legend()
plt.grid()
plt.show()
except Exception as e:
    print(f'Terjadi error saat membuat plot: {e}')
```

Mounted at /content/drive

Data berhasil dimuat.

Kolom yang ditemukan dalam file: Index(['2 ', '4 ', '12.0'], dtype='object')

Nama kolom setelah dibersihkan: Index(['2', '4', '12.0'], dtype='object')

Model SVM berhasil dilatih.

Hasil prediksi:

X: 3, 5 -> Prediksi: 20.0

X: 4, 6 -> Prediksi: 24.0

X: 5, 7 -> Prediksi: 28.0

X: 6, 8 -> Prediksi: 28.0

X: 2, 4 -> Prediksi: 20.0

X: 3, 5 -> Prediksi: 20.0

X: 4, 6 -> Prediksi: 24.0

X: 5, 7 -> Prediksi: 28.0

X: 6, 8 -> Prediksi: 28.0

X: 2, 4 -> Prediksi: 20.0

X: 3, 5 -> Prediksi: 20.0

X: 4, 6 -> Prediksi: 24.0

X: 5, 7 -> Prediksi: 28.0

X: 6, 8 -> Prediksi: 28.0

X: 2, 4 -> Prediksi: 20.0

X: 3, 5 -> Prediksi: 20.0

X: 4, 6 -> Prediksi: 24.0

X: 5, 7 -> Prediksi: 28.0

X: 6, 8 -> Prediksi: 28.0

X: 2, 4 -> Prediksi: 20.0

X: 3, 5 -> Prediksi: 20.0

X: 4, 6 -> Prediksi: 24.0

X: 5, 7 -> Prediksi: 28.0

X: 6, 8 -> Prediksi: 28.0

X: 2, 4 -> Prediksi: 20.0

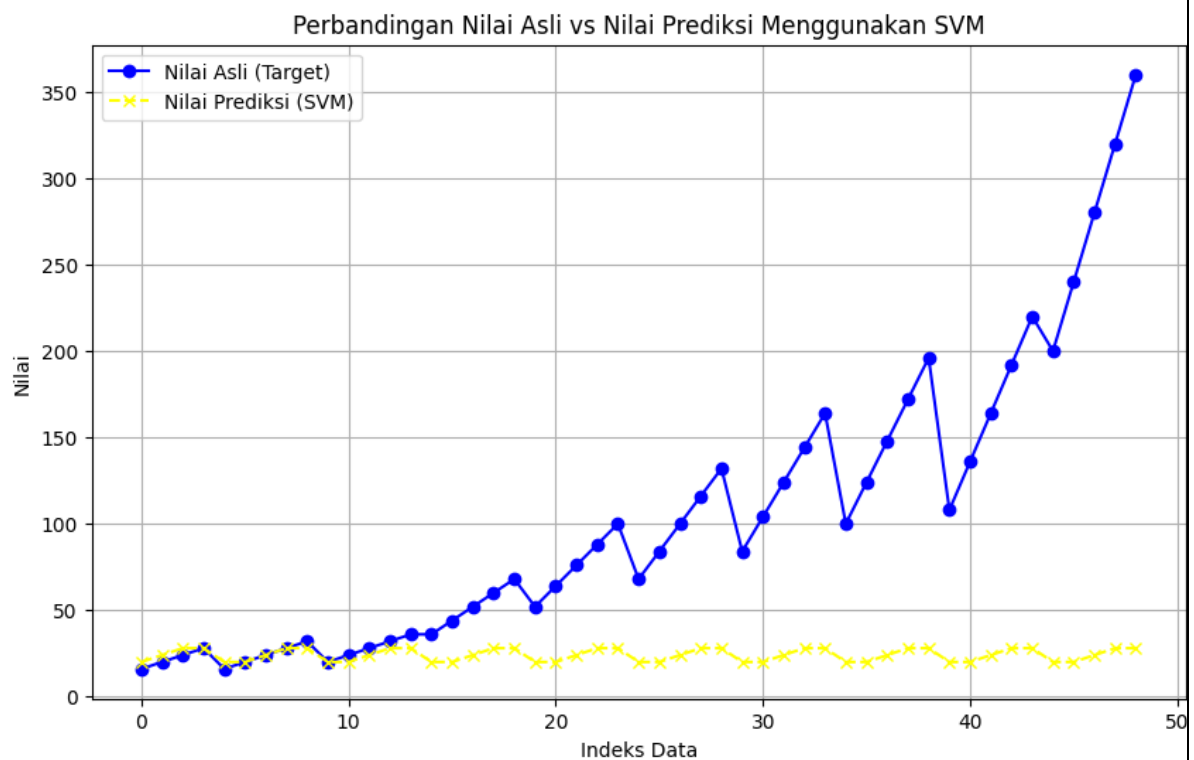
X: 3, 5 -> Prediksi: 20.0

X: 4, 6 -> Prediksi: 24.0

X: 5, 7 -> Prediksi: 28.0

X: 6, 8 -> Prediksi: 28.0

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



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

Grafik tersebut menggambarkan perbandingan antara nilai asli (target) dan nilai prediksi yang dihasilkan oleh model Support Vector Machine (SVM). Nilai asli direpresentasikan dengan garis biru yang menunjukkan kenaikan tajam seiring bertambahnya indeks data, sementara nilai prediksi digambarkan oleh garis kuning dengan pola yang cenderung konstan di seluruh

rentang data. Dari visualisasi ini, tampak jelas bahwa model SVM gagal mengikuti pola kenaikan nilai asli yang signifikan.

Ketidaksesuaian ini mengindikasikan adanya masalah pada proses pemodelan. Model tidak mampu menangkap hubungan yang ada dalam data, yang mungkin disebabkan oleh beberapa faktor. Salah satunya adalah ketidakcukupan informasi dalam fitur yang digunakan untuk melatih model. Jika variabel-variabel input tidak merepresentasikan pola yang ada dalam data target, hasil prediksi cenderung tidak akurat. Selain itu, parameter default pada SVM, seperti jenis kernel atau tingkat regularisasi, mungkin tidak sesuai dengan karakteristik data yang kompleks.

Pola nilai asli yang menunjukkan kenaikan tajam dan fluktuasi yang signifikan juga menandakan bahwa data target memiliki karakteristik non-linear yang kompleks. Model SVM mungkin tidak cukup fleksibel untuk menangkap pola ini tanpa penyesuaian yang lebih spesifik. Di sisi lain, distribusi data yang tidak merata atau adanya outlier juga dapat memengaruhi performa model secara keseluruhan.

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

Metode trapezoid adalah pendekatan numerik yang digunakan untuk menghitung integral fungsi pada interval tertentu dengan cara mendekati area di bawah kurva menggunakan bentuk trapesium. Dalam algoritma ini, interval $[a,b]$ dibagi menjadi beberapa bagian kecil yang disebut partisi. Lebar masing-masing partisi, yang disebut h , dihitung berdasarkan panjang interval dibagi dengan jumlah partisi yang diinginkan. Kemudian, nilai fungsi $f(x)$ pada setiap titik interval digunakan untuk membentuk trapesium, dan area masing-masing trapesium dihitung dan dijumlahkan untuk mendapatkan nilai integral total.

Program yang dibuat sebelumnya memanfaatkan metode ini untuk menghitung integral dari data yang disediakan dalam file teks. Langkah pertama yang dilakukan adalah membaca file tersebut, membersihkan nama kolom dari spasi tambahan, dan memastikan bahwa data memiliki format yang sesuai. Kemudian, nilai integral dihitung dengan menggunakan rumus trapezoid. Hasil dari perhitungan ini mencerminkan area yang mendekati nilai integral aktual dari fungsi yang diberikan. Program ini sangat berguna dalam kasus di mana fungsi $f(x)$ tidak dapat diintegrasikan secara analitik.

Untuk meningkatkan fungsionalitas program, metode Support Vector Machine (SVM) digunakan untuk memprediksi nilai integral berdasarkan data hasil perhitungan trapezoid. Data hasil perhitungan metode trapezoid (terdiri dari parameter batas interval a dan b , serta nilai integral yang diperoleh) digunakan sebagai dataset untuk melatih model SVM. Model SVM mempelajari hubungan antara parameter input (a dan b) dengan hasil integral, sehingga dapat digunakan untuk memprediksi nilai integral pada interval yang baru tanpa perlu melakukan perhitungan numerik lagi.