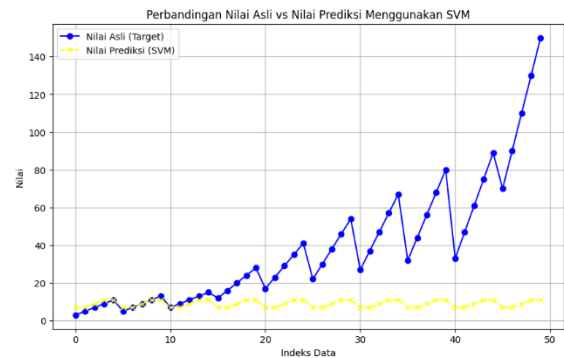
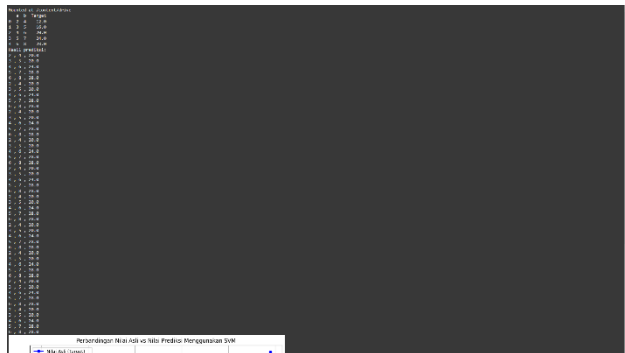


Praktikum Fisika Komputasi

Support Vector Machine (SVM)

Ramli Zhafran Amarillo (1227030027)

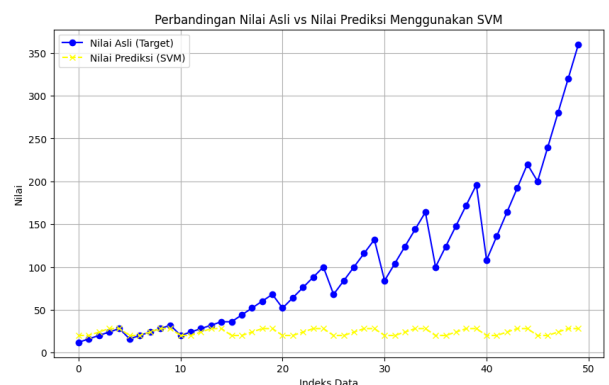
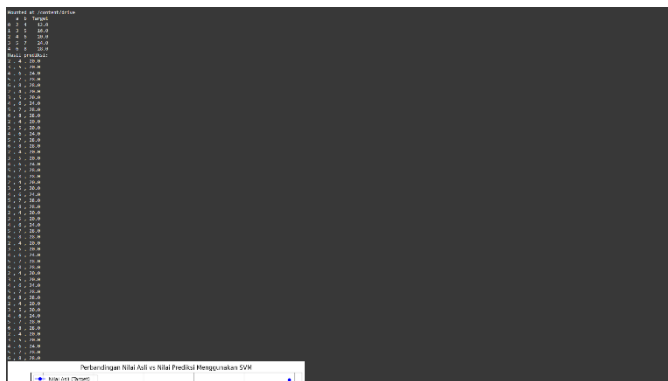
1.) Prediksi integral trapezoid Ke-1 menggunakan Support Vector Machine



2.) Perbandingan Nilai asli dan nilai prediksi

Berdasarkan grafik perbandingan antara nilai asli (target) dan nilai prediksi yang dihasilkan oleh Support Vector Machine (SVM), terlihat bahwa prediksi SVM cukup konsisten tetapi tidak sepenuhnya mengikuti pola nilai asli. Nilai asli (garis biru) menunjukkan perubahan yang signifikan seiring bertambahnya indeks data, sedangkan prediksi SVM (garis kuning) cenderung rata dengan pola naik turun yang kecil di sekitar nilai tertentu. Hal ini menunjukkan bahwa model SVM kesulitan menangkap pola pertumbuhan eksponensial atau kompleks pada data target.

3.) Prediksi integral trapezoid Ke-2 menggunakan Support Vector Machine



4.) Prediksi integral trapezoid Ke-2 menggunakan Support Vector Machine

Dari grafik hasil nomor 3, terlihat bahwa nilai prediksi SVM (garis kuning) sama seperti pola nomor 1 dimana konsisten pada kisaran nilai rendah, sementara nilai asli (garis biru) menunjukkan pola pertumbuhan eksponensial yang semakin tajam dengan meningkatnya indeks data. Hal tersebut menunjukkan SVM kembali tidak mampu menangkap pola non-linier yang kompleks.

5.) Penjelasan Algoritma SVM

```
# 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/MyDrive/Tugastrapezoid.txt' # Ganti
dengan path sesuai lokasi file Anda di Google Drive

# Membaca data dari file (periksa delimiter, misalnya koma atau tab)
Database = pd.read_csv(file_path, sep=",", header=0) # Pastikan
delimiter sesuai dengan file

# Cek isi data untuk memastikan format benar
print(Database.head())

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

# 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))

# 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()

```

Pada bagian pertama mengimpor library seperti **numpy** untuk manipulasi array, **pandas** untuk pengolahan data, **scikit-learn (svm)** untuk algoritma SVM, **google.colab** untuk mengakses file di Google Drive, dan **matplotlib** untuk visualisasi grafik. File data (Tugastrapezoid.txt) yang disimpan di Google Drive diakses menggunakan `drive.mount` dan dibaca ke dalam DataFrame menggunakan `pd.read_csv`. Data ini diverifikasi formatnya dengan menampilkan beberapa baris pertama menggunakan `print(Database.head())`. Dataset memiliki kolom a dan b sebagai data input (X) serta Target sebagai label target (y).

Model SVM dibuat menggunakan `svm.SVC()` dari Scikit-learn dan dilatih menggunakan `clf.fit()` dengan data input dan target. Setelah model dilatih, prediksi dilakukan pada data input menggunakan `clf.predict(X.values)`, dan hasilnya disimpan dalam `y_pred`. Prediksi ini dibandingkan dengan nilai asli (target) untuk memverifikasi performa model. Grafik dibuat menggunakan **Matplotlib** untuk memvisualisasikan perbandingan antara nilai asli dan nilai prediksi dari SVM.