

Nama : Aszka Sazkia J.S

NIM : 1227030006

PRAKTIKUM FISIKA KOMPUTASI

MACHINE LEARNING

Kode Program Machine Learning

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

# Creating the dataset (only positive values for X)
np.random.seed(0)
X = [[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]]
Y = [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]

# Splitting the dataset into training and test sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)

# Creating a linear regression model
linear_model = LinearRegression()
linear_model.fit(X_train, Y_train)

# Creating a polynomial regression model of degree 2
poly_features_2 = PolynomialFeatures(degree=2)
X_train_poly_2 = poly_features_2.fit_transform(X_train)
poly_model_2 = LinearRegression()
poly_model_2.fit(X_train_poly_2, Y_train)

# Making predictions for the entire dataset
X_sorted = np.sort(X, axis=0) # Sorting X for smooth plotting
Y_pred_linear_all = linear_model.predict(X_sorted)
Y_pred_poly_2_all =
poly_model_2.predict(poly_features_2.transform(X_sorted))

# Evaluating the models
mse_linear = mean_squared_error(Y_test,
linear_model.predict(X_test))
mse_poly_2 = mean_squared_error(Y_test,
poly_model_2.predict(poly_features_2.transform(X_test)))
```

```

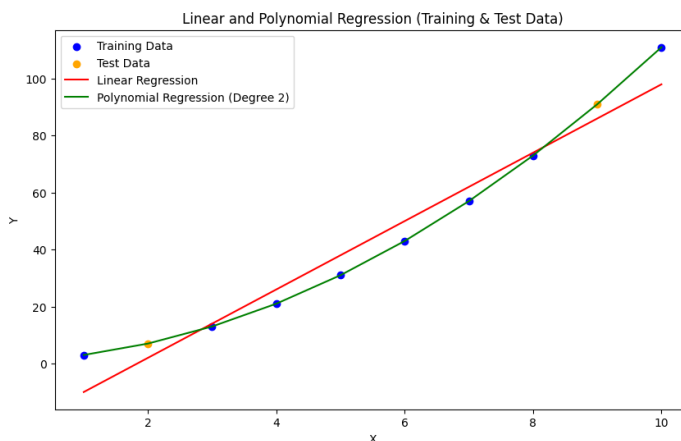
print(f"Mean Squared Error (Linear): {mse_linear:.2f}")
print(f"Mean Squared Error (Polynomial Degree 2):
{mse_poly_2:.2f}")

# Plotting regression results for the entire dataset
plt.figure(figsize=(10, 6))
plt.scatter(X_train, Y_train, color='blue', label='Training
Data') # Training data
plt.scatter(X_test, Y_test, color='orange', label='Test
Data') # Test data
plt.plot(X_sorted, Y_pred_linear_all, color='red', label='Linear
Regression') # Linear regression line
plt.plot(X_sorted, Y_pred_poly_2_all, color='green',
label='Polynomial Regression (Degree 2)') # Polynomial
regression line

plt.xlabel('X')
plt.ylabel('Y')
plt.title('Linear and Polynomial Regression (Training & Test
Data)')
plt.legend()
plt.show()

```

Mean Squared Error (Linear): 25.00
Mean Squared Error (Polynomial Degree 2): 0.00



Algoritma Pemrograman

1. Import Library yang akan digunakan.
 - Menggunakan `numpy` untuk pengolahan angka, penjumlahan dsb.
 - Menggunakan `matplotlib.pyplot` untuk membuat grafik.
 - Memanggil `sklearn.linear_model` untuk melakukan regresi linear.
 - Memanggil `sklearn.preprocessing` untuk regresi polinomial.
 - Memanfaatkan `sklearn.model_selection` untuk membagi data menjadi set pelatihan dan pengujian.

2. Dataset dibuat dengan nilai positif untuk variabel `X` dan `Y`
3. Dataset dibagi menjadi bagian pelatihan (`X_train`, `Y_train`) dan bagian pengujian (`X_test`, `Y_test`) menggunakan `train_test_split`.
4. Membuat regresi polinomial derajat 2 menggunakan `PolynomialFeatures`, kemudian model regresi dilatih.
5. Model melakukan prediksi untuk seluruh dataset agar dapat divisualisasikan. Kemudian Mean Squared Error (MSE) dihitung untuk kedua model untuk menilai kinerja mereka.

Penjelasan Grafik

Berdasarkan grafik, terlihat bahwa sumbu x merepresentasikan data X, sedangkan sumbu y merepresentasikan data Y. Data ini dibagi menjadi dua kelompok, yaitu 80% sebagai data pelatihan dan 20% sebagai data pengujian, sesuai dengan ketentuan soal. Jumlah data pada X dan Y sama. Grafik tersebut memvisualisasikan pasangan data mulai dari x1y1 hingga x10y10. Dua jenis kurva ditampilkan, yaitu regresi linear yang digambarkan dengan garis merah dan regresi polinomial derajat 2. Berdasarkan analisis dataset, kurva polinomial derajat 2 lebih mencerminkan pola yang sesuai dengan nilai data.