Praktikum Fisika Komputasi

Machine Learning Regresi Linear dan Polinomial

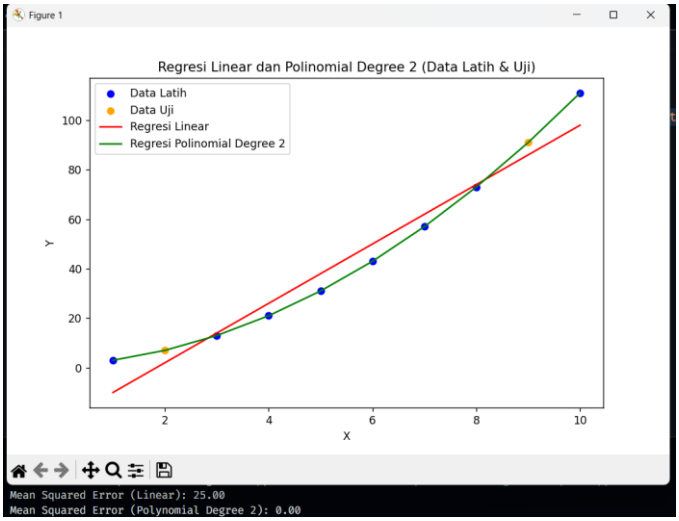
Ramli Zhafran Amarillo (1227030027)

1.) Modifikasi code ML Ke-3 nilai X dan Y

Code dengan dataset data yang telah diubah sesuai dengan modul

```
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

# Membuat dataset (hanya nilai positif untuk 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]

# Membagi dataset menjadi data latih dan uji
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)

# Membuat model regresi linear
linear_model = LinearRegression()
linear_model.fit(X_train, Y_train)

# Membuat model regresi polinomial derajat 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)

X_sorted = np.sort(X, axis=0)  # Urutkan X untuk membuat plot
mulus
Y_pred_linear_all = linear_model.predict(X_sorted)
Y_pred_poly_2_all =
poly_model_2.predict(poly_features_2.transform(X_sorted))
```

```
# Evaluasi model
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}")

# Plot hasil regresi untuk seluruh dataset
plt.figure(figsize=(10, 6))
plt.scatter(X_train, Y_train, color='blue', label='Data
Latih')  # Data latih
plt.scatter(X_test, Y_test, color='orange', label='Data
Uji')  # Data uji
plt.plot(X_sorted, Y_pred_linear_all, color='red',
label='Regresi Linear')  # Garis regresi linear
plt.plot(X_sorted, Y_pred_poly_2_all, color='green',
label='Regresi Polinomial Degree 2')  # Garis regresi
polinomial degree 2
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Regresi Linear dan Polinomial Degree 2 (Data Latih
& Uji)')
plt.legend()
plt.show()
```



2.) 80% data latih dan 20% data uji

```
# Membuat dataset (hanya nilai positif untuk 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]

# Membagi dataset menjadi data latih dan uji
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

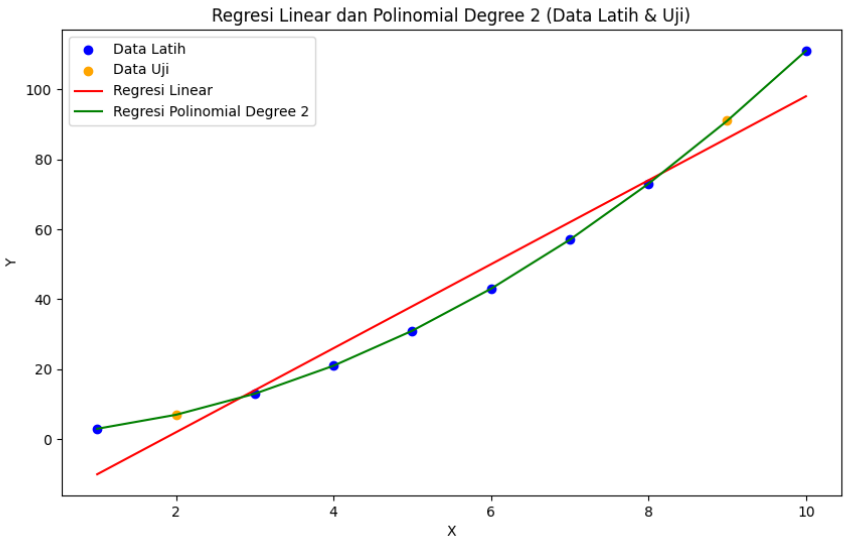
Output yang keluar akan sama seperti no.1, disitu range untuk data uji sebesar 0.2 atau 20% karena besar full range nya 1 sehingga 80% sisanya adalah data latih

3.) Perbandingan Mean Squad Error Linear dan Polynomial

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

Karena dataset tersebut sepertinya kuadratik maka MSE akan bernilai 0 tetapi data regresi linear tidak menangkap hasil sama sehingga terdapat error sebesar 25.00

4.) Visualisasi hasil



5.) Penjelasan Kode Program dan Hasil gambar

Kode ini menggunakan 2 pendekeatan prediksi yaitu regresi linear dan regresi polinomial derajat 2, menggunakan dataset sederhana yang dikasih di modul untuk dicoba. Pertama, di import library yang dibutuhkan, seperti **numpy** untuk operasi matematika numerik, **matplotlib.pyplot** untuk visualisasi data, dan beberapa fungsi dari library Scikit-Learn yang harus di instal dulu lewat CMD kalo running di compiler code editor seperti vs code atau IDLE Python untuk membuat model. Lalu program akan membagi dataset, serta mengevaluasi performa model. Selanjutnya, dataset didefinisikan menggunakan variabel **X** (input) dan **Y** (output), di mana **Y** menunjukkan pola kenaikan yang tidak linier, misalnya 3,7,13,…3, 7, 13, dst. Dataset ini dibagi menjadi 80% data pelatihan dan 20% data pengujian dengan fungsi **train_test_split**. Pembagian ini penting supaya bisa melatih model di sebagian data dan menguji performanya di data lain yang belum pernah dilihat.

Setelah dataset siap, model pertama yang dibuat adalah regresi linear menggunakan **LinearRegression**, yang mencoba menyatukan hubungan antara **X** dan **Y** dengan garis lurus. Model kedua menggunakan regresi polinomial derajat 2, yang dibuat dengan **PolynomialFeatures(degree=2)** untuk menambahkan fitur baru berupa X2X^2, sehingga model bisa menangkap pola nonlinear. Kedua model ini dilatih menggunakan data pelatihan, dan hasil prediksi diuji pada data pengujian menggunakan metrik Mean Squared Error (MSE). Fungsi ini menghitung rata-rata error kuadrat antara prediksi model dan nilai asli. Semakin kecil nilai MSE, semakin baik performa model sepertinya ini kalau anggapan saya.

Hasil akhirnya divisualisasikan dalam grafik gambar pada nomor 4. Titik biru adalah data pelatihan, sedangkan titik orange kekuningan adalah data pengujian. Garis merah adalah prediksi dari regresi linear, dan garis hijau menunjukkan hasil regresi polinomial derajat 2. Dari grafik ini terlihat bahwa regresi linear hanya mampu membuat garis lurus, sehingga kurang akurat untuk data nonlinear terlihat titik biru tidak pas dengan tengah garisnya. Sebaliknya, regresi polinomial lebih fleksibel dan lebih tepat menangkap pola data yang nonlinear, seperti terlihat dari garis hijau yang lebih sesuai dengan data.