

Nama : Ulfah Hasanah

NIM : 1227030036

Jawaban Tugas Modul 10 : Machine Learning Regresi Linear dan Polinomial

1. Import Library:

```
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
```

- NumPy: Digunakan untuk manipulasi data numerik.
- Matplotlib.pyplot: Digunakan untuk membuat visualisasi grafik.
- LinearRegression dan PolynomialFeatures: Digunakan untuk membuat model regresi linear dan polinomial.
- mean_squared_error: Digunakan untuk menghitung kesalahan rata-rata kuadrat (MSE) antara nilai prediksi dan nilai sebenarnya.

2. Data Set

```
# 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]
```

- X: Data input (variabel independen).
- Y: Data output (variabel dependen).

3. Membagi Dataset

```
# 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)
```

- Data Latih: `X_train` adalah data untuk variabel independen (misalnya `X`), dan `Y_train` adalah data untuk variabel dependen (misalnya `Y`) yang mengikuti pola kuadrat ($y=x^2$).
- Data Uji: `X_test` adalah data yang digunakan untuk menguji model, dan `Y_test` adalah nilai sebenarnya untuk evaluasi.
- `train_test_split`: Membagi dataset menjadi:
 - 80% untuk data latih.
 - 20% untuk data uji.
- `random_state`: Untuk memastikan pembagian data tetap sama jika dijalankan ulang.

4. Membuat Model Regresi Linear

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

- `LinearRegression`: Membuat model regresi linear.
- `fit`: Melatih model dengan data latih.

5. Membuat Model Regresi Polinomial Derajat 2

```
# 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)
```

- `poly_features = PolynomialFeatures(degree=2)`: Membuat objek untuk mengubah data `X` menjadi fitur polinomial derajat 2 (menambahkan X^2 sebagai fitur baru).
- `X_train_poly = poly_features.fit_transform(X_train)`: Mengubah data latih `X` menjadi fitur polinomial (menambahkan kolom X^2).
- `poly_model = LinearRegression()`: Membuat objek model regresi linear untuk regresi polinomial.

- `poly_model.fit(X_train_poly, y_train)`: Melatih model regresi polinomial dengan data latih yang sudah diubah menjadi fitur polinomial.

6. Prediksi

```
# Membuat prediksi untuk keseluruhan dataset
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))
```

- `predict`: Membuat prediksi menggunakan model linear dan polinomial.
- Menggunakan model regresi linear yang telah dilatih untuk memprediksi nilai `yyy` berdasarkan data uji.
- Menggunakan model regresi polinomial untuk memprediksi nilai `yyy` berdasarkan data uji yang telah diubah menjadi fitur polinomial.

7. Evaluasi Model

```
# 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}")
```

- `mse_linear = mean_squared_error(y_test, y_pred_linear)`: Menghitung MSE untuk model regresi linear, yang membandingkan nilai `yyy` yang sebenarnya (data uji `ytesty_testytest`) dengan hasil prediksi model regresi linear.
- `mse_poly = mean_squared_error(y_test, y_pred_poly)`: Menghitung MSE untuk model regresi polinomial derajat 2, yang membandingkan nilai `y` yang sebenarnya dengan hasil prediksi model regresi polinomial.
- Kemudian program ini akan menampilkan nilai MSE untuk kedua model, yang memberi gambaran tentang akurasi prediksi.

8. Visualisasi

Untuk memvisualisasikan hasil dari kedua model, kita menambahkan kode untuk menampilkan grafik yang menggambarkan data latih, data uji, serta hasil prediksi dari kedua model. Berikut adalah kode untuk grafik:

```
# 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()
```

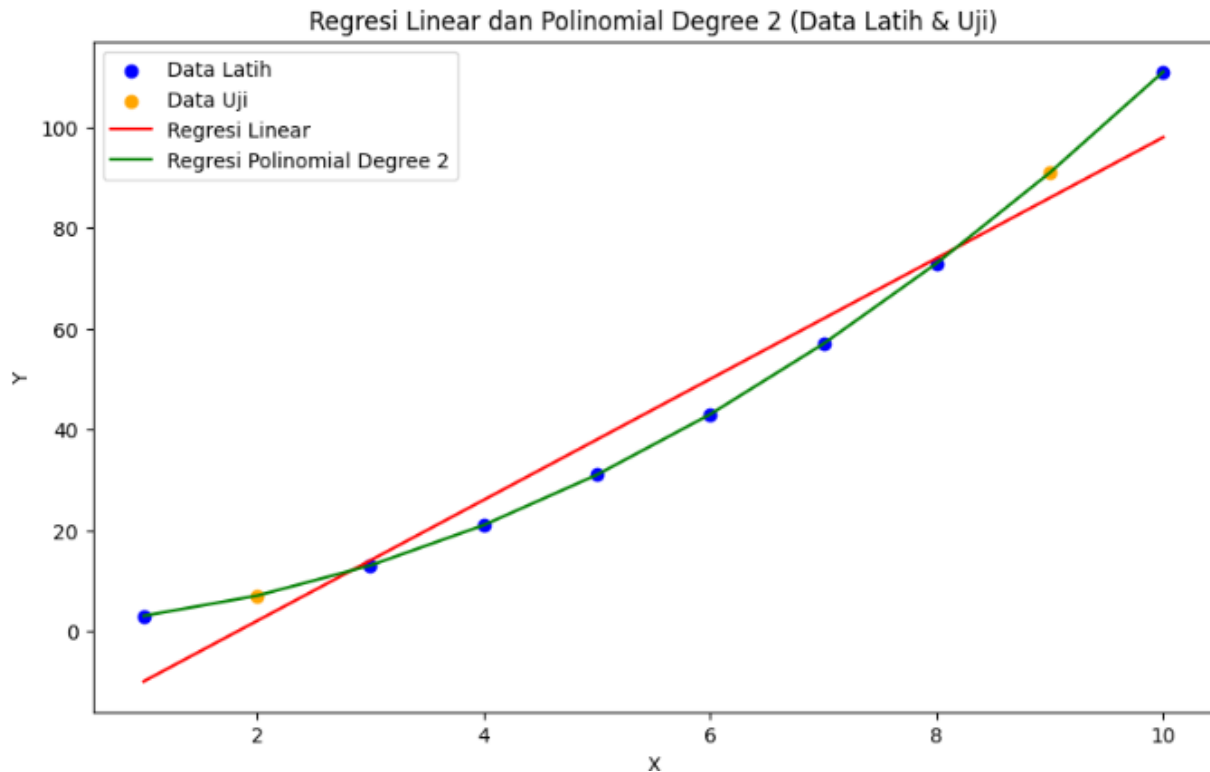
- Scatter plot: Menampilkan data latih (biru) dan uji (oranye).
- Garis regresi:
 - Merah: Garis regresi linear.
 - Hijau: Garis regresi polinomial derajat 2.
- Label & Judul: Menjelaskan isi plot.

Berikut tampilan hasil MSE dan grafik dari kode program ini:

❖ Nilai Mean Scured Error

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

❖ Grafik



Grafik ini menggambarkan hubungan antara data asli, yang terdiri dari data latih dan data uji, serta prediksi dari dua model: regresi linear dan regresi polinomial derajat 2. Data latih direpresentasikan oleh titik-titik biru, sementara data uji ditampilkan dengan titik-titik oranye. Kedua jenis data ini menunjukkan pola yang melengkung, menandakan hubungan non-linear antara variabel X dan Y. Model regresi linear, yang diwakili oleh garis merah, mencoba memprediksi hubungan ini dengan asumsi linearitas. Namun, dari grafik terlihat bahwa garis ini tidak mampu mengikuti kelengkungan pola data dengan baik, khususnya di ujung rentang X, di mana prediksinya jauh dari data sebenarnya. Akibatnya, model ini memiliki kesalahan prediksi yang cukup besar, dengan nilai Mean Squared Error (MSE) sebesar 25.00. Hal ini menunjukkan bahwa model regresi linear tidak sesuai untuk dataset ini. Sebaliknya, regresi polinomial derajat 2, yang diwakili oleh garis hijau, memberikan hasil yang jauh lebih akurat. Model ini lebih fleksibel dan mampu menangkap kelengkungan pola data dengan sangat baik, baik pada data latih maupun data uji. Garis hijau hampir melewati setiap titik data, menunjukkan bahwa model ini dapat memprediksi data dengan sempurna. Hal ini tercermin dalam nilai MSE untuk regresi polinomial derajat 2, yaitu 0.00, yang menunjukkan tidak adanya kesalahan dalam prediksi data uji.

Perbedaan mencolok dalam performa kedua model ini, sebagaimana terlihat pada grafik, mendukung hasil evaluasi berdasarkan nilai MSE. Regresi linear menunjukkan kesalahan yang signifikan karena kurang fleksibel dalam merepresentasikan hubungan non-linear, sedangkan regresi polinomial derajat 2 menunjukkan prediksi sempurna karena kemampuannya mencocokkan pola data yang kompleks. Grafik ini dengan jelas menunjukkan pentingnya memilih model yang tepat sesuai dengan karakteristik data. Dalam hal ini, regresi polinomial derajat 2 terbukti lebih sesuai karena mampu menangkap pola kuadrat yang mendasari hubungan antara variabel X dan Y, menghasilkan prediksi yang akurat tanpa kesalahan.