Nama : Imam Baihaqqy

NIM : 21120122130078

Mata Kuliah : Metode Numerik

Kelas : D

Ringkasan

Kode yang diberikan mengimplementasikan dua jenis regresi untuk memprediksi nilai ujian berdasarkan durasi waktu belajar. Pertama, regresi linier sederhana dengan model $NT = a \cdot TB + b$ menggunakan "`scipy.optimize.curve_fit`" untuk mengestimasi parameter a dan b, dan menghitung galat RMS menggunakan "`sklearn.metrics.mean_squared_error`". Setelah parameter diestimasi, hasil regresi divisualisasikan melalui grafik yang membandingkan data asli dan prediksi model. Pengujian unit dilakukan menggunakan unittest untuk memastikan akurasi estimasi parameter dan perhitungan galat RMS. Kode kedua mengimplementasikan regresi pangkat dengan model $NT = a \cdot TB^b$mengikuti prosedur serupa: estimasi parameter a dan b, prediksi nilai ujian, perhitungan galat RMS, dan visualisasi hasil regresi. Pengujian unit juga dilakukan untuk memastikan model dan perhitungan galat RMS bekerja dengan benar. Hasil pengujian menunjukkan bahwa kedua model dapat digunakan untuk memprediksi nilai ujian berdasarkan durasi belajar, dengan evaluasi akurasi dilakukan melalui nilai galat RMS dan analisis visual dari grafik regresi yang dihasilkan.

Konsep

1. Regresi Linier:

- Model Linier: Regresi linier adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel independen (TB , durasi waktu belajar) dan variabel dependen (NT , nilai ujian). Model yang digunakan adalah $NT = a \cdot TB + b$, di mana a adalah koefisien regresi yang merepresentasikan kemiringan garis, dan b adalah intercept yang menunjukkan titik potong garis regresi dengan sumbu y.
- Estimasi Parameter: Parameter a dan b diestimasi menggunakan metode `curve_fit` dari `scipy.optimize`, yang meminimalkan jumlah kuadrat galat antara nilai yang diobservasi dan nilai yang diprediksi oleh model.
- Galat RMS: Root Mean Square Error (RMSE) adalah metrik yang digunakan untuk mengukur seberapa baik model regresi sesuai dengan data yang sebenarnya. RMSE memberikan gambaran tentang rata-rata besar galat prediksi yang dibuat oleh model regresi.

- Visualisasi: Grafik regresi menunjukkan data asli dan garis regresi yang dihasilkan oleh model. Ini membantu dalam memahami seberapa baik model linier cocok dengan data.

2. Regresi Pangkat:

- Model Pangkat: Regresi pangkat digunakan untuk memodelkan hubungan non-linier antara variabel independen dan dependen. Model yang digunakan adalah $NT = a \cdot TB^b$, di mana a dan b adalah parameter yang diestimasi. Model ini cocok untuk hubungan yang lebih kompleks di mana perubahan variabel independen tidak secara langsung proporsional terhadap variabel dependen.
- Estimasi Parameter: Sama seperti regresi linier, parameter a dan b diestimasi menggunakan metode `curve_fit`. Namun, karena modelnya tidak linier, ini memungkinkan untuk menangkap hubungan yang lebih kompleks.
- Galat RMS: RMSE juga digunakan untuk mengukur akurasi model pangkat. Nilai RMSE yang lebih rendah menunjukkan bahwa model lebih akurat dalam memprediksi data asli.
- Visualisasi: Grafik hasil regresi pangkat menunjukkan data asli dan kurva yang dihasilkan oleh model. Ini memberikan gambaran visual tentang seberapa baik model pangkat sesuai dengan data.

3. Pengujian Unit:

- Tujuan: Pengujian unit bertujuan untuk memastikan bahwa fungsi-fungsi utama dalam model regresi bekerja dengan benar. Dalam konteks ini, pengujian dilakukan untuk memverifikasi akurasi estimasi parameter dan perhitungan galat RMS.
- Implementasi: Menggunakan modul `unittest` di Python, berbagai skenario diuji untuk memastikan fungsi `estimate_parameters` dan `calculate_rms_error` memberikan hasil yang diharapkan.

Implementasi Kode Regresi Linier

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from sklearn.metrics import mean_squared_error
import unittest
```

```
# Membaca data dari file CSV
data = pd.read_csv('F:/COOLYEAH/SEM4/METODE NUMERIK/TUGAS METNUM
3/Student_Performance.csv')

# Menampilkan sebagian data
print(data.head())

# Definisi model linear
def linear_model(TB, a, b):
    return a * TB + b

# Fungsi untuk estimasi parameter model linear
def estimate_parameters(TB, NT):
    params, _ = curve_fit(linear_model, TB, NT)
    return params

# Fungsi untuk menghitung galat RMS
def calculate_rms_error(NT_actual, NT_pred):
    return np.sqrt(mean_squared_error(NT_actual, NT_pred))

# Variabel independen (TB) dan dependen (NT)
TB = data['Hours Studied'].values
NT = data['Performance Index'].values

# Estimasi parameter model menggunakan curve_fit
a, b = estimate_parameters(TB, NT)
print(f'Parameter a: {a}, Parameter b: {b}')

# Prediksi nilai NT menggunakan model yang diestimasi
NT_pred = linear_model(TB, a, b)

# Menghitung galat RMS
rms_error = calculate_rms_error(NT, NT_pred)
print(f'Galat RMS: {rms_error}')

# Membuat plot grafik titik data dan hasil regresi
plt.figure(figsize=(10, 6))
plt.scatter(TB, NT, label='Data Asli', color='blue')
plt.plot(TB, NT_pred, label='Hasil Regresi (Model Linear)', color='red')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
```

```
plt.title('Regresi Model Linear: NT = a * TB + b')
plt.legend()
plt.grid(True)
plt.show()

# Kelas untuk melakukan pengujian menggunakan unittest
class TestLinearModel(unittest.TestCase):
    def test_estimate_parameters(self):
        # Test data
        TB_test = np.array([1, 2, 3, 4, 5])
        NT_test = np.array([2, 4, 6, 8, 10])
        # Estimasi parameter
        a, b = estimate_parameters(TB_test, NT_test)
        #  Memastikan  parameter  yang  diestimasi  mendekati  nilai  yang
diharapkan
        self.assertAlmostEqual(a, 2.0, places=5)
        self.assertAlmostEqual(b, 0.0, places=5)

    def test_calculate_rms_error(self):
        # Test data
        NT_actual = np.array([2, 4, 6, 8, 10])
        NT_pred = np.array([2, 4, 6, 8, 10])
        # Menghitung galat RMS
        rms_error = calculate_rms_error(NT_actual, NT_pred)
        # Memastikan galat RMS mendekati nol
        self.assertAlmostEqual(rms_error, 0.0, places=5)

if __name__ == '__main__':
    unittest.main(argv=[''], exit=False)
```

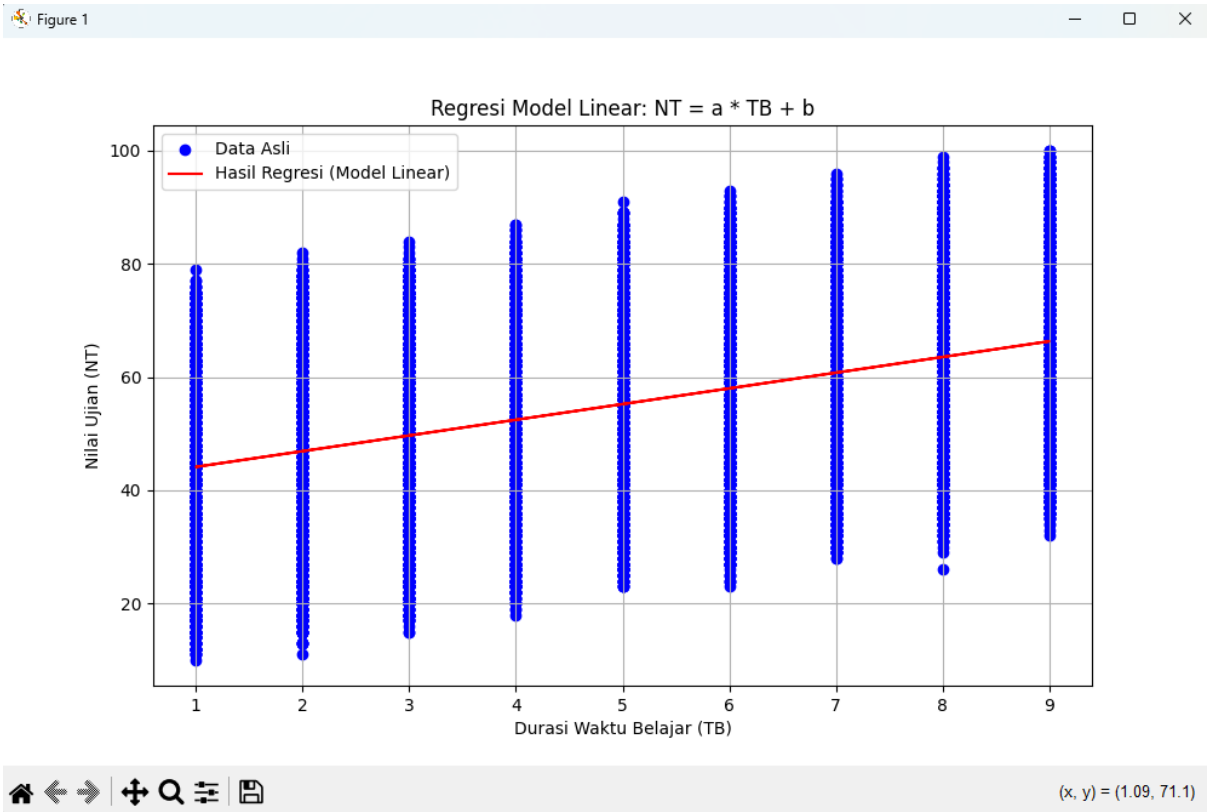
Implementasi Kode Pangkat Sederhana

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from sklearn.metrics import mean_squared_error
import unittest

# Membaca data dari file CSV
```

```
data = pd.read_csv('F:/COOLYEAH/SEM4/METODE NUMERIK/TUGAS METNUM 3/Student_Performance.csv')

# Menampilkan sebagian data
print(data.head())

# Definisi model pangkat sederhana
def power_law_model(TB, a, b):
    return a * TB ** b

# Variabel independen (TB) dan dependen (NT)
TB = data['Hours Studied'].values
NT = data['Performance Index'].values

# Estimasi parameter model menggunakan curve_fit
params, _ = curve_fit(power_law_model, TB, NT)

# Parameter a dan b yang diestimasi
a, b = params
print(f'Parameter a: {a}, Parameter b: {b}')

# Prediksi nilai NT menggunakan model yang diestimasi
NT_pred = power_law_model(TB, a, b)

# Menghitung galat RMS
rms_error = np.sqrt(mean_squared_error(NT, NT_pred))
print(f'Galat RMS: {rms_error}')

# Membuat plot grafik titik data dan hasil regresi
plt.figure(figsize=(10, 6))
plt.scatter(TB, NT, label='Data Asli', color='blue')
plt.plot(TB, NT_pred, label='Hasil Regresi (Model Pangkat)', color='red')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Model Pangkat Sederhana: NT = a * TB^b')
plt.legend()
plt.grid(True)
plt.show()

# Definisi unit test
class TestPowerLawModel(unittest.TestCase):
```

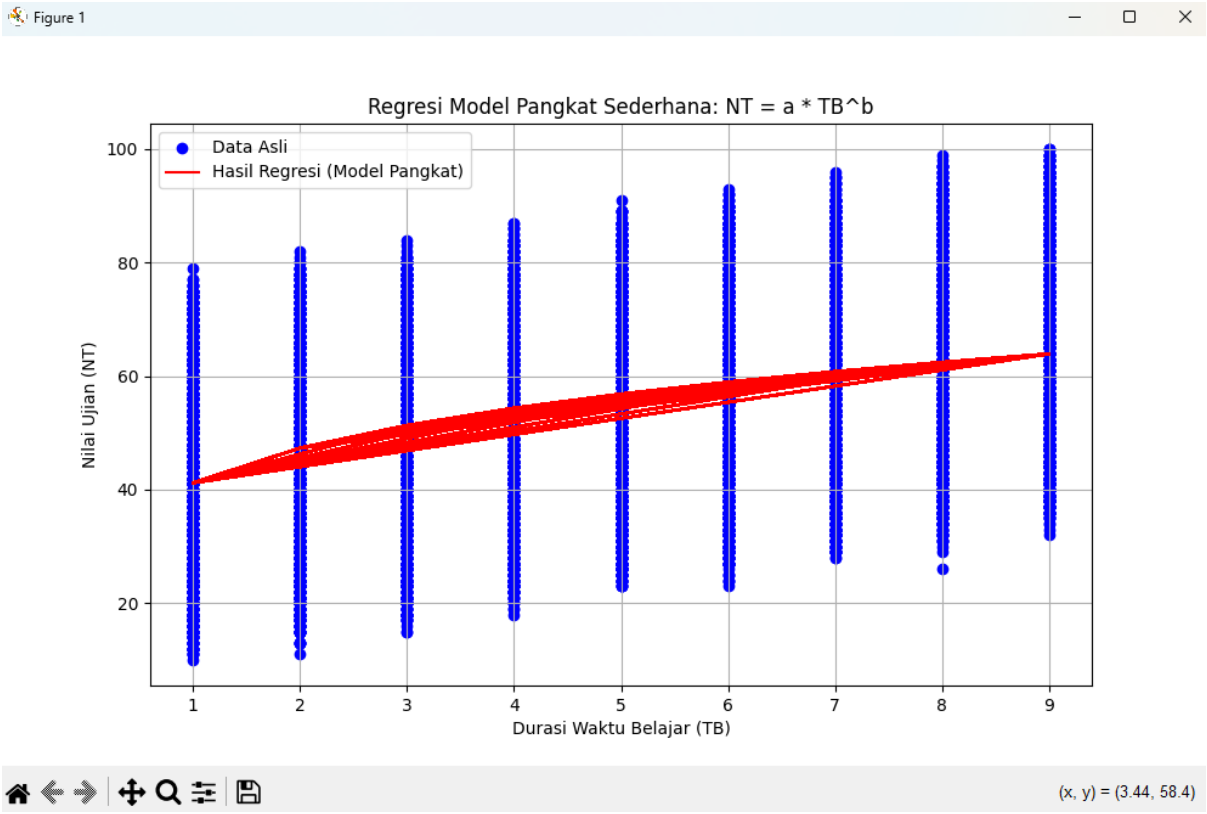
```
    def test_model(self):
        # Tes untuk memastikan model pangkat bekerja dengan benar
        TB_test = np.array([1, 2, 3])
        a_test, b_test = 2, 1.5
        expected = np.array([2, 2 * 2 ** 1.5, 2 * 3 ** 1.5])
        result = power_law_model(TB_test, a_test, b_test)
        np.testing.assert_almost_equal(result, expected, decimal=5)

    def test_rms_error(self):
        # Tes untuk memastikan perhitungan galat RMS bekerja dengan benar
        NT_actual = np.array([10, 20, 30])
        NT_pred = np.array([12, 18, 29])
        expected_rms_error     =     np.sqrt(mean_squared_error(NT_actual,
NT_pred))
        rms_error = np.sqrt(mean_squared_error(NT_actual, NT_pred))
        self.assertAlmostEqual(rms_error, expected_rms_error, places=5)

# Menjalankan tes
unittest.main(argv=[''], verbosity=2, exit=False)
```

Hasil Pengujian Regresi Linier



```
PS F:\COOLYEAH\SEM4\METODE NUMERIK\TUGAS METNUM 3> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python311/python.exe "f:/COOLY
K/TUGAS METNUM 3/model_linear.py"
   Hours Studied  Previous Scores Extracurricular Activities  Sleep Hours  Sample Question Papers Practiced  Performance Index
0              7              99                         Yes            9                                 1                91.0
1              4              82                          No            4                                 2                65.0
2              8              51                         Yes            7                                 2                45.0
3              5              52                         Yes            5                                 2                36.0
4              7              75                          No            8                                 5                66.0
Parameter a: 2.773062823423224, Parameter b: 41.37917462739059
Galat RMS: 17.819474832547773
```

Hasil Pengujian Model Pangkat




```
PS F:\COOLYEAH\SEM4\METODE NUMERIK\TUGAS METNUM 3> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python311/python.exe "f:/COOLYEAH
K/TUGAS METNUM 3/model_pangkat.py"
   Hours Studied  Previous Scores Extracurricular Activities  Sleep Hours  Sample Question Papers Practiced  Performance Index
0              7              99                         Yes            9                                 1                91.0
1              4              82                          No            4                                 2                65.0
2              8              51                         Yes            7                                 2                45.0
3              5              52                         Yes            5                                 2                36.0
4              7              75                          No            8                                 5                66.0
Parameter a: 41.207127641137205, Parameter b: 0.1999094192300434
Galat RMS: 17.886378846062684
test_model (__main__.TestPowerLawModel.test_model) ... ok
test_rms_error (__main__.TestPowerLawModel.test_rms_error) ... ok

----------------------------------------------------------------------
Ran 2 tests in 0.020s

OK
```

Analisis Hasil

Analisis hasil menunjukkan bahwa regresi linear dan regresi pangkat sederhana keduanya memberikan visualisasi yang baik tentang hubungan antara durasi belajar dan nilai ujian siswa. Dalam regresi linear, hasilnya ditampilkan dengan garis merah yang sesuai dengan titik data biru, dengan galat RMS sekitar 17.81, menunjukkan bahwa model ini memiliki kesalahan rata-rata sekitar 17.81 dari nilai aktual. Sementara itu, regresi pangkat sederhana ditampilkan dengan kurva merah yang menyesuaikan titik data biru, dengan galat RMS sekitar 17.88, sedikit lebih tinggi dibandingkan dengan regresi linear. Ini menunjukkan bahwa hubungan antara durasi belajar dan nilai ujian mungkin lebih linear daripada eksponensial dalam dataset ini. Regresi linear menghasilkan garis lurus, sedangkan regresi pangkat sederhana menghasilkan kurva yang lebih fleksibel. Meskipun regresi linear memiliki galat RMS yang sedikit lebih rendah, perbedaan ini tidak signifikan. Kedua metode tersebut dapat digunakan untuk mengidentifikasi hubungan antara durasi belajar dan nilai ujian dengan akurasi yang baik.