

TUGAS METODE NUMERIK

Prabaswara Nasywa Maharani

21120122130076

Regresi Pangkat Sederhana

<https://github.com/pruubie/Tugas-MetNum-3---Prabaswara-Nasywa-Maharani>

SOURCE CODE

(maaf pak saya pakai data sendiri karena bingung dengan data yang di web,, terima kasih banyak)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# Data Jumlah Latihan Soal (NL) dan Nilai Ujian (NT)
NL = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) # Jumlah Latihan Soal
NT = np.array([52, 55, 60, 63, 65, 70, 72, 75, 80, 85]) # Nilai Ujian

# Fungsi model pangkat sederhana
def power_model(x, a, b):
    return a * np.power(x, b)

# Melakukan fitting pada model pangkat sederhana
popt, pcov = curve_fit(power_model, NL, NT)
a, b = popt

# Memprediksikan nilai NT menggunakan model pangkat sederhana
NT_pred_pangkat = power_model(NL, a, b)

# Plot hasil regresi pangkat sederhana
plt.figure(figsize=(10, 6))
plt.scatter(NL, NT, color='red', label='Data Aktual')
plt.plot(NL, NT_pred_pangkat, color='blue', label='Regresi Pangkat Sederhana')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.legend()
plt.title('Regresi Pangkat Sederhana Jumlah Latihan Soal vs Nilai Ujian')
plt.grid(True)
plt.show()

# Menampilkan koefisien regresi pangkat sederhana
print(f"Koefisien regresi pangkat sederhana: a = {a}, b = {b}")
```

LANGKAH-LANGKAH & PENJELASAN

1. Mengimpor modul yang diperlukan:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

- `numpy`: Digunakan untuk operasi numerik dan manipulasi data (array, matriks, fungsi matematika).
 - `matplotlib.pyplot`: Digunakan untuk visualisasi data (plot, grafik).
 - `curve_fit` dari `scipy.optimize` digunakan untuk melakukan fitting kurva.
2. Mendefinisikan dua array NumPy, `NL` dan `NT`, yang berisi jumlah latihan soal dan nilai ujian masing-masing.

```
NL = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) # Jumlah
Latihan Soal
NT = np.array([52, 55, 60, 63, 65, 70, 72, 75, 80, 85]) #
Nilai Ujian
```

- `NL` menyimpan jumlah latihan soal dari 1 hingga 10.
 - `NT` menyimpan nilai ujian yang sesuai dengan jumlah latihan soal.
3. Definisikan fungsi model pangkat sederhana

```
def power_model(x, a, b):
    return a * np.power(x, b)
```

- Fungsi ini menerima parameter `x` (jumlah latihan soal), `a`, dan `b`.
- Mengembalikan hasil dari $a * x^b$.

4. Fitting kurva pada data yang diberikan:

```
popt, pcov = curve_fit(power_model, NL, NT)
a, b = popt
```

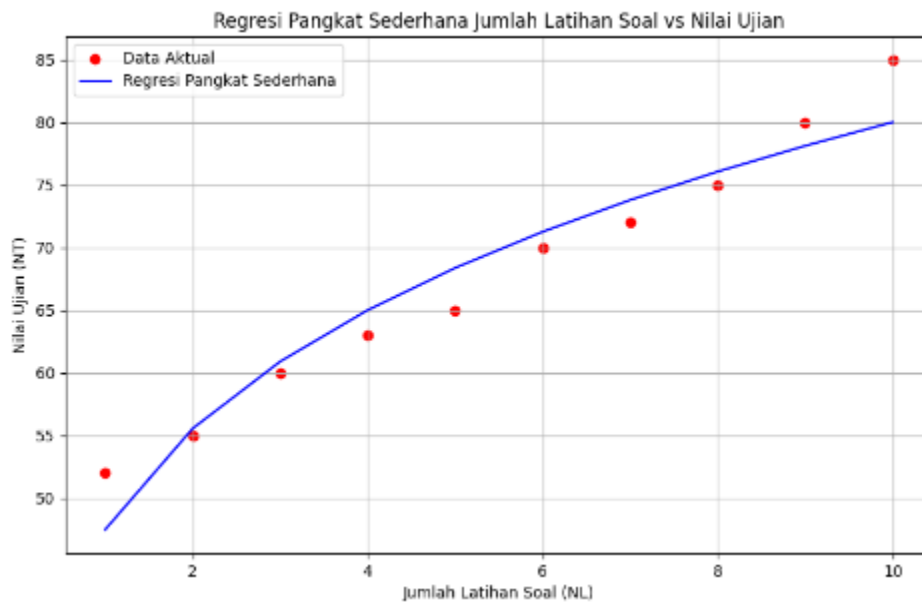
- `curve_fit` mengembalikan parameter yang paling sesuai (`popt`) untuk fungsi model yang diberikan dan kovariansinya (`pcov`).
- Anda menyimpan nilai parameter yang optimal dalam variabel `a` dan `b`.

5. Plot:

```
plt.figure(figsize=(10, 6))
plt.scatter(NL, NT, color='red', label='Data Aktual')
plt.plot(NL, NT_pred_pangkat, color='blue', label='Regresi
Pangkat Sederhana')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.legend()
plt.title('Regresi Pangkat Sederhana Jumlah Latihan Soal
vs Nilai Ujian')
plt.grid(True)
plt.show()
```

Membuat plot dengan jumlah latihan soal di sumbu x dan nilai ujian di sumbu y. Titik data aktual direpresentasikan dalam warna merah, sementara hasil regresi ditampilkan sebagai garis biru.

ANALISIS GRAFIK



- **Data Aktual:**

Titik merah menunjukkan nilai ujian yang sebenarnya untuk setiap jumlah latihan soal. Tampak pola kenaikan nilai ujian seiring dengan bertambahnya jumlah latihan soal.

- **Kurva Prediksi:**

Kurva biru menunjukkan nilai ujian yang diprediksi oleh model regresi pangkat sederhana. Kurva tersebut tampaknya cocok dengan pola data aktual, yang menunjukkan bahwa model dapat menggambarkan tren data dengan cukup baik.

- **Interpretasi Kurva**

Kurva tidak linier, mencerminkan sifat hubungan pangkat antara jumlah latihan soal dan nilai ujian. Peningkatan nilai ujian tidak sepenuhnya proporsional dengan jumlah latihan soal, tetapi ada peningkatan yang semakin menurun seiring dengan bertambahnya jumlah latihan soal.