

Nama : Femas Arianda Rizki

NIM : 21120122130080

Kelas : Metode Numerik – B

A. Regresi Metode Linear

1. Ringkasan

Terdapat sebuah data di mana di dalamnya terdapat kolom durasi waktu belajar (TB) dan kolom nilai ujian siswa (NT). Dalam data tersebut terdapat 1000 baris. Dengan menggunakan regresi metode linear dapat digunakan untuk mencari hubungan tersebut. Dalam menerapkan regresi linear dapat menggunakan beberapa cara, salah satunya seperti di sini yaitu dengan bahasa pemrograman python.

Setelah dibuat kode untuk menjalankan program regresi linear, dan juga kode untuk menampilkan plot grafik dari data yang ada. Keluaran dari kode program itu didapat, di mana bisa digunakan untuk mencari berapa kira-kira nilai ujian siswa (NT) ketika dimasukkan masukan durasi waktu belajar (TB) tertentu. Keluaran juga berupa berupa plot grafik untuk lebih jelas mengenai hubungan antara TB dan NT. Dalam kode ini juga ditampilkan galat yang terjadi dalam kasus ini.

2. Konsep

- **Regresi Linear** adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu variabel independen (prediktor) dan satu variabel dependen (respons) dengan menyesuaikan garis lurus (linear) ke data. Tujuan utama regresi linear adalah untuk menemukan nilai koefisien a dan b dari persamaan garis $y=ax+ b$ yang paling sesuai dengan data.
- **Root Mean Square Error (RMSE)** adalah metrik yang digunakan untuk mengukur seberapa baik model regresi sesuai dengan data yang diberikan. RMSE mengukur rata-rata perbedaan kuadrat antara nilai yang diprediksi oleh model dan nilai sebenarnya. Nilai RMSE yang lebih kecil menunjukkan model yang lebih baik.

3. Implementasi kode

Kode Sumber, Kode Testing, Plot Grafik, dan Galat RMS:

```
# Nama      : Femas Arianda Rizki
# NIM       : 21120122130080
# Kelas     : Metode Numerik - B
```

```

# kode sumber
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv("Student_Performance.csv")
data

data.shape

data.info()

data["Hours Studied"]

data["Performance Index"]

def barchart(column):
    count = data[column].value_counts()

    index = count.index
    values = count.values

    fig = plt.figure(figsize= (10, 5))
    plt.bar(index, values, color = "blue", width = 0.2)

barchart("Hours Studied")
barchart("Performance Index")

# mengekstrak kolom "Hours Studied" dan "Performance Index"
TB = data['Hours Studied'].values
NT = data['Performance Index'].values

# jumlah data poin (n)
n = len(TB)

# membangun matriks A dan vektor b untuk sistem persamaan linier
 $Ax = b$ 
# A adalah matriks (n x 2), di mana kolom pertama berisi data
hours_studied dan kolom kedua berisi angka satu (untuk
menambahkan intercept ke dalam model)
A = np.vstack([TB, np.ones(n)]).T
b = NT

# menyelesaikan persamaan  $Ax = b$  menggunakan metode kuadrat
terkecil (least squares)
x = np.linalg.lstsq(A, b, rcond=None)[0]

# memisahkan nilai a dan b dari solusi x

```

```

a, b = x

# kode testing
print(f'Persamaan garis lurusnya yaitu:  $y = {a:.4f}x + {b:.4f}$ ')
x = 4.5
print(f'TB = {x}')
print(f'Untuk TB = {x}, akan menghasilkan NT kira-kira sebesar {(a*x + b):.4f}')

# plot grafik data dan hasil regresi
plt.figure(figsize=(10, 6))
plt.plot(TB, NT, 'o', label='Original data', markersize=4)
plt.plot(TB, a*TB + b, 'r', label=f'Regresi:  $y = {a:.4f}x + {b:.4f}$ ')
plt.xlabel('Waktu belajar (TB)')
plt.ylabel('Nilai ujian siswa (NT)')
plt.title('Hubungan durasi waktu belajar (TB) terhadap nilai ujian siswa (NT)')
plt.legend()
plt.grid(True)
plt.show()

# Galat RMS
predicted_NT = a * TB + b
residuals = NT - predicted_NT
squared_residuals = residuals ** 2
mean_squared_error = np.mean(squared_residuals)
rms = np.sqrt(mean_squared_error)

print(f'Galat pada regresi metode linear ini yaitu: {rms:.6f}')

```

Penjelasan Alur Kode:

a. Mengimpor pustaka

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

```

Bagian ini mengimpor pustaka yang diperlukan:

- pandas untuk manipulasi data.
- numpy untuk operasi numerik.
- matplotlib.pyplot untuk visualisasi data.

b. Membaca file CSV, menampilkan data, dimensi, dan info

```

data = pd.read_csv("Student_Performance.csv")
data
data.shape
data.info()

```

- Bagian ini membaca file CSV berjudul "Student_Performance.csv" menggunakan `pandas` dan menyimpan data dalam `DataFrame` `data`. Data ini kemudian ditampilkan.
- Bagian ini menampilkan dimensi `DataFrame` `data` dalam bentuk tuple (jumlah baris, jumlah kolom).
- Bagian ini memberikan informasi ringkas tentang `DataFrame` `data`, termasuk jumlah entri, tipe data untuk setiap kolom, dan jumlah nilai non-null.

c. Mengekstrak dan menampilkan kolom "Hours Studied" dan "Performance Index"

```
data["Hours Studied"]
data["Performance Index"]
```

Bagian ini mengekstrak dan menampilkan kolom "Hours Studied" dan "Performance Index" dari `DataFrame` `data`.

d. Mendefinisikan fungsi `barchart`

```
def barchart(column):
    count = data[column].value_counts()

    index = count.index
    values = count.values

    fig = plt.figure(figsize=(10, 5))
    plt.bar(index, values, color="blue", width=0.2)

    barchart("Hours Studied")
    barchart("Performance Index")
```

Bagian ini mendefinisikan fungsi `barchart` yang membuat diagram batang untuk kolom yang diberikan. Fungsi ini menghitung frekuensi nilai unik dalam kolom, kemudian memplotnya sebagai diagram batang. Fungsi ini kemudian dipanggil untuk kolom "Hours Studied" dan "Performance Index".

e. Mengekstrak kolom "Hours Studied" dan "Performance Index" serta menghitung jumlah data poin (n)

```
TB = data['Hours Studied'].values
NT = data['Performance Index'].values
# jumlah data poin (n)
n = len(TB)
```

- Bagian ini mengekstrak kolom "Hours Studied" dan "Performance Index" dari `DataFrame` `data` dan mengubahnya menjadi array `NumPy` yang disimpan dalam variabel `TB` dan `NT`.

- Bagian ini menghitung jumlah data poin (n) dengan menghitung panjang dari array TB.

f. Membangun matriks A dan vektor b untuk sistem persamaan linier $Ax=b$

```
# membangun matriks A dan vektor b untuk sistem persamaan
linier Ax = b
# A adalah matriks (n x 2), di mana kolom pertama berisi data
hours_studied dan kolom kedua berisi angka satu (untuk
menambahkan intercept ke dalam model)
A = np.vstack([TB, np.ones(n)]).T
b = NT
```

Bagian ini membangun matriks A dan vektor b untuk sistem persamaan linier $Ax=b$:

- A adalah matriks (n x 2), di mana kolom pertama berisi data TB dan kolom kedua berisi angka satu (untuk menambahkan intercept ke dalam model).
- np.vstack digunakan untuk menumpuk array TB dan array satu di atas satu sama lain secara vertikal, dan kemudian .T (transpose) digunakan untuk menukar baris dan kolom sehingga bentuknya menjadi (n x 2).

g. Menyelesaikan persamaan $Ax=b$, dan memisahkan nilai a dan b dari solusi x

```
# menyelesaikan persamaan Ax = b menggunakan metode kuadrat
terkecil (least squares) x = np.linalg.lstsq(A, b,
rcond=None)[0]
# memisahkan nilai a dan b dari solusi x
a, b = x
```

- Bagian ini menyelesaikan persamaan $Ax=b$ menggunakan metode kuadrat terkecil (least squares). Fungsi np.linalg.lstsq mengembalikan beberapa nilai, tetapi kita hanya membutuhkan solusi x, yang merupakan array yang berisi koefisien a dan b. Dengan [0], kita mengambil elemen pertama dari hasil yang berisi nilai a dan b.
- Bagian ini memisahkan nilai a dan b dari solusi x.

h. Mencetak persamaan garis lurus

```
# kode testing
print(f'Persamaan garis lurusnya yaitu: y = {a:.4f}x +
{b:.4f}')
x = 4.5
print(f'TB = {x}')
print(f'Untuk TB = {x}, akan menghasilkan NT kira-kira sebesar
{(a*x + b):.4f}')
```

Bagian ini mencetak persamaan garis lurus dalam bentuk $y=ax+b$ dengan koefisien a dan b yang telah dihitung, diformat hingga empat angka desimal. Kemudian, bagian ini menghitung dan mencetak nilai NT yang diprediksi untuk TB sebesar 4.5.

i. Membuat plot grafik

```
# plot grafik data dan hasil regresi
plt.figure(figsize=(10, 6))
plt.plot(TB, NT, 'o', label='Original data', markersize=4)
plt.plot(TB, a*TB + b, 'r', label=f'Regresi: y = {a:.4f}x + {b:.4f}')
plt.xlabel('Waktu belajar (TB)')
plt.ylabel('Nilai ujian siswa (NT)')
plt.title('Hubungan durasi waktu belajar (TB) terhadap nilai ujian siswa (NT)')
plt.legend()
plt.grid(True)
plt.show()
```

- `plt.figure(figsize=(10, 6))` membuat figure baru dengan ukuran 10x6 inci.
- `plt.plot(TB, NT, 'o', label='Original data', markersize=4)` memplot data asli sebagai titik-titik (marker 'o') dengan label "Original data" dan ukuran marker 4.
- `plt.plot(TB, a*TB + b, 'r', label=f'Regresi: y = {a:.4f}x + {b:.4f}')` memplot garis lurus yang difitkan (dengan warna merah 'r') berdasarkan persamaan $y=ax+b$, dengan label `f'Regresi: y = {a:.4f}x + {b:.4f}'`
- `plt.xlabel('Waktu belajar (TB)')` dan `plt.ylabel('Nilai ujian siswa (NT)')` memberikan label pada sumbu x dan y.
- `plt.title('Hubungan durasi waktu belajar (TB) terhadap nilai ujian siswa (NT)')` memberikan judul pada plot.
- `plt.legend()` menampilkan legenda pada plot.
- `plt.grid(True)` menambahkan grid pada plot.
- `plt.show()` menampilkan plot.

j. Menghitung galat RMS

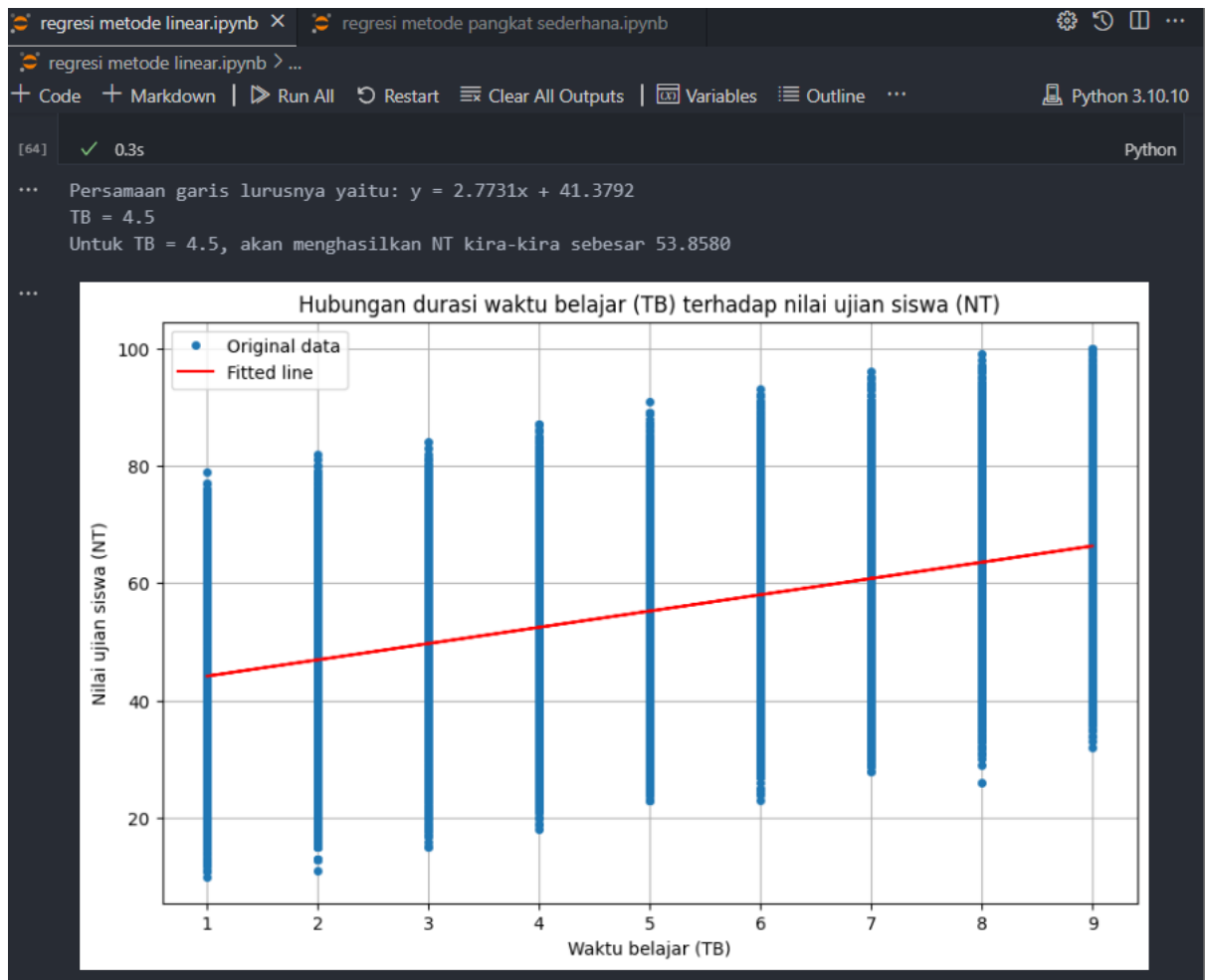
```
# Galat RMS
predicted_NT = a * TB + b
residuals = NT - predicted_NT
squared_residuals = residuals ** 2
mean_squared_error = np.mean(squared_residuals)
rms = np.sqrt(mean_squared_error)

print(f'Galat pada regresi metode linear ini yaitu: {rms:.6f}')
```

- Menghitung nilai prediksi NT menggunakan persamaan garis regresi $y=ax+b$
- Menghitung selisih antara nilai sebenarnya dan nilai prediksi (residuals) untuk setiap titik data.
- Menghitung kuadrat dari setiap residual.
- Menghitung rata-rata dari kuadrat residual (Mean Squared Error).

- Menghitung akar kuadrat dari rata-rata kuadrat residual (Root Mean Square Error, RMSE).
- Mencetak nilai RMSE dari model regresi linear, diformat hingga enam angka desimal.

4. Hasil pengujian



5. Analisis hasil

Analisis hasil keluaran dari kode regresi metode linear:

1. Hasil regresi pada TB = 4.5

- Hasil interpolasi pada titik TB = 4.5 adalah 53.8580
- Artinya, nilai yang regresi pada titik TB = 4.5 kira-kira sekitar 53.8580

2. Grafik hasil regresi

- Grafik menunjukkan pola yang tidak melalui semua titik, namun cukup mewakili kecenderungan titik data.

- Grafik menunjukkan bahwa regresi metode linear secara keseluruhan cocok dengan data atau permasalahan yang diberikan.
- Garis regresi lurus sempurna, hal ini karena dihasilkan dari persamaan linear.

Penjelasan hasil keluaran:

- Titik-titik data yang diketahui jumlahnya 1000 digunakan untuk melakukan regresi.
- Hasil regresi pada titik TB = 4.5 kira-kira sekitar 53.8580, yang didapatkan dari perhitungan regresi linear.
- Grafik hasil regresi menunjukkan bahwa metode linear mampu meregresi data dengan baik, yaitu cukup mewakili kecenderungan titik data.

Dengan demikian, metode regresi metode linear dapat digunakan untuk memperkirakan nilai di antara titik-titik data yang diketahui dengan kurang baik.

B. Regresi Metode Pangkat Sederhana

1. Ringkasan

Terdapat sebuah data di mana di dalamnya terdapat kolom durasi waktu belajar (TB) dan kolom nilai ujian siswa (NT). Dalam data tersebut terdapat 1000 baris. Dengan menggunakan regresi metode pangkat sederhana dapat digunakan untuk mencari hubungan tersebut. Dalam menerapkan regresi pangkat sederhana dapat menggunakan beberapa cara, salah satunya seperti di sini yaitu dengan bahasa pemrograman python.

Setelah dibuat kode untuk menjalankan program regresi pangkat sederhana, dan juga kode untuk menampilkan plot grafik dari data yang ada. Keluaran dari kode program itu didapat, di mana bisa digunakan untuk mencari berapa kira-kira nilai ujian siswa (NT) ketika dimasukkan masukan durasi waktu belajar (TB) tertentu. Keluaran juga berupa berupa plot grafik untuk lebih jelas mengenai hubungan antara TB dan NT. Dalam kode ini juga ditampilkan galat yang terjadi dalam kasus ini.

2. Konsep

- **Regresi metode pangkat sederhana** adalah metode yang digunakan untuk memodelkan hubungan antara satu variabel independen (prediktor) dan satu variabel dependen (respons) dengan menyesuaikan kurva yang mengikuti bentuk pangkat ke data. Tujuan utama regresi pangkat sederhana adalah untuk menemukan nilai koefisien C dan pangkat b dari persamaan kurva $y = Cx^b$ yang paling sesuai dengan data.

- **Root Mean Square Error (RMSE)** adalah metrik yang digunakan untuk mengukur seberapa baik model regresi sesuai dengan data yang diberikan. RMSE mengukur rata-rata perbedaan kuadrat antara nilai yang diprediksi oleh model dan nilai sebenarnya. Nilai RMSE yang lebih kecil menunjukkan model yang lebih baik.

3. Implementasi kode

Kode Sumber, Kode Testing, Plot Grafik, dan Galat RMS:

```
# Nama      : Femas Arianda Rizki
# NIM       : 21120122130080
# Kelas     : Metode Numerik - B

# Kode Sumber
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Memuat data dari file CSV
data = pd.read_csv("Student_Performance.csv")

# Mengekstrak kolom "Hours Studied" dan "Performance Index"
TB = data['Hours Studied'].values
NT = data['Performance Index'].values

# Log-transformasi data
log_TB = np.log(TB)
log_NT = np.log(NT)

# Menyusun matriks untuk eliminasi Gauss
A = np.vstack([log_TB, np.ones(len(log_TB))]).T
b = log_NT

# Eliminasi Gauss untuk menyelesaikan Ax = b
coeffs = np.linalg.lstsq(A, b, rcond=None)[0]
b_est = coeffs[0]
log_C_est = coeffs[1]
C_est = np.exp(log_C_est)

# kode testing
print(f"Persamaan regresi pangkat sederhana: y = {C_est:.4f} * x^{b_est:.4f}")
x = 2.5
print(f'TB = {x}')
print(f'Untuk TB = {x}, akan menghasilkan NT kira-kira sebesar {(C_est * x**b_est):.4f}')
```

```

# Buat garis regresi dengan x yang lebih halus
x_fit = np.linspace(min(TB), max(TB), 100)
y_fit = C_est * x_fit**b_est

# Plot grafik data dan hasil regresi
plt.figure(figsize=(10, 6))
plt.plot(TB, NT, 'o', label='Original data', markersize=4)
plt.plot(x_fit, y_fit, color='red', label=f'Regresi: y = {C_est:.4f} * x^{b_est:.4f}')
plt.xlabel('Waktu belajar (TB)')
plt.ylabel('Nilai ujian siswa (NT)')
plt.title('Hubungan durasi waktu belajar (TB) terhadap nilai ujian siswa (NT)')
plt.legend()
plt.grid(True)
plt.show()

# Galat RMS
predicted_NT = C_est * TB**b_est
residuals = NT - predicted_NT
squared_residuals = residuals ** 2
mean_squared_error = np.mean(squared_residuals)
rms = np.sqrt(mean_squared_error)

print(f'Galat pada regresi metode pangkat sederhana ini yaitu: {rms:.6f}')

```

Penjelasan Alur Kode:

a. Mengimpor pustaka

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

```

Bagian ini mengimpor pustaka yang diperlukan:

- pandas untuk manipulasi data.
- numpy untuk operasi numerik.
- matplotlib.pyplot untuk visualisasi data.

b. Membaca file CSV

```

data = pd.read_csv("Student_Performance.csv")

```

- Membaca data dari file Student_Performance.csv dan menyimpannya dalam dataframe data.

c. Mengekstrak dan menampilkan kolom "Hours Studied" dan "Performance Index"

```

TB = data['Hours Studied'].values
NT = data['Performance Index'].values

```

- Mengambil kolom Hours Studied sebagai variabel independen TB.
- Mengambil kolom Performance Index sebagai variabel dependen NT.

d. Log-transformasi data

```
log_TB = np.log(TB)
log_NT = np.log(NT)
```

- Mentransformasi TB dan NT dengan logaritma natural untuk mengubah model pangkat menjadi model linear.

e. Menyusun matriks untuk eliminasi Gauss

```
A = np.vstack([log_TB, np.ones(len(log_TB))]).T
b = log_NT
```

- Membentuk matriks A dari logaritma TB dan sebuah vektor konstanta untuk persamaan linear.
- Membentuk vektor b dari logaritma NT.

f. Eliminasi Gauss untuk menyelesaikan $Ax=b$

```
coeffs = np.linalg.lstsq(A, b, rcond=None)[0]
b_est = coeffs[0]
log_C_est = coeffs[1]
C_est = np.exp(log_C_est)
```

- Menggunakan metode least squares untuk menyelesaikan persamaan linear dan mendapatkan koefisien coeffs.
- b_est adalah eksponen dari TB.
- log_C_est adalah logaritma natural dari konstanta C, yang kemudian diubah kembali menjadi C_est dengan fungsi eksponensial.

g. Kode testing

```
print(f"Persamaan regresi pangkat sederhana: y = {C_est:.4f} * x^{b_est:.4f}")
x = 2.5
print(f'TB = {x}')
print(f'Untuk TB = {x}, akan menghasilkan NT kira-kira sebesar {(C_est * x**b_est):.4f}')
```

- Mencetak persamaan regresi pangkat sederhana yang dihasilkan.
- Menguji persamaan dengan nilai x tertentu untuk melihat hasil prediksi NT.

h. Buat garis regresi dengan x yang lebih halus

```
x_fit = np.linspace(min(TB), max(TB), 100)
y_fit = C_est * x_fit**b_est
```

- Membuat rentang nilai x yang lebih halus (x_fit) untuk menggambar garis regresi yang lebih mulus.
- Menghitung nilai y_fit menggunakan persamaan regresi pangkat sederhana.

i. Plot grafik data dan hasil regresi

```
plt.figure(figsize=(10, 6))
plt.plot(TB, NT, 'o', label='Original data', markersize=4)
plt.plot(x_fit, y_fit, color='red', label=f'Regresi: y = {C_est:.4f} * x^{b_est:.4f}')
plt.xlabel('Waktu belajar (TB)')
plt.ylabel('Nilai ujian siswa (NT)')
plt.title('Hubungan durasi waktu belajar (TB) terhadap nilai ujian siswa (NT)')
plt.legend()
plt.grid(True)
plt.show()
```

- Membuat plot grafik dengan data asli sebagai titik-titik biru.
- Menggambar garis regresi merah berdasarkan persamaan yang dihasilkan.
- Menambahkan label, judul, legenda, dan grid pada grafik untuk meningkatkan keterbacaan.

j. Menghitung galat RMS

```
predicted_NT = C_est * TB**b_est
residuals = NT - predicted_NT
squared_residuals = residuals ** 2
mean_squared_error = np.mean(squared_residuals)
rms = np.sqrt(mean_squared_error)

print(f'Galat pada regresi metode pangkat sederhana ini yaitu: {rms:.6f}')
```

- Menghitung nilai NT yang diprediksi (predicted_NT) menggunakan model regresi pangkat sederhana.
- Menghitung residuals (selisih antara nilai asli dan nilai prediksi).
- Menghitung mean squared error (MSE) dengan mengkuadratkan residuals dan mengambil rata-ratanya.
- Menghitung root mean squared error (RMS) sebagai akar dari MSE untuk menilai galat model.
- Mencetak nilai galat RMS untuk menilai seberapa baik model regresi pangkat sederhana dalam memprediksi data.

- Grafik menunjukkan pola yang tidak melalui semua titik, namun cukup mewakili kecenderungan titik data.
- Grafik menunjukkan bahwa regresi metode pangkat sederhana secara keseluruhan cocok dengan data atau permasalahan yang diberikan.
- Garis regresi tidak lurus sempurna, namun sedikit melengkung karena dihasilkan dari persamaan yang memiliki pangkat.

Penjelasan hasil keluaran:

- Titik-titik data yang diketahui jumlahnya 1000 digunakan untuk melakukan regresi.
- Hasil regresi pada titik $TB = 4.5$ kira-kira sekitar 52.4003, yang didapatkan dari perhitungan regresi pangkat sederhana.
- Grafik hasil regresi menunjukkan bahwa metode pangkat sederhana mampu meregresi data dengan baik, yaitu cukup mewakili kecenderungan titik data.

Dengan demikian, metode regresi metode pangkat sederhana dapat digunakan untuk memperkirakan nilai di antara titik-titik data yang diketahui dengan kurang baik.