

# Implementasi Regresi

## Metode Numerik

Nama : Rachel Savitri

NIM : 21120122140111

Kelas : C

Link GitHub: <https://github.com/aaceelll/Implementasi-Regresi-Rachel-Savitri-21120122140111>

Source Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Membaca data dari file CSV
file_path = 'Student_Performance.csv'
df = pd.read_csv(file_path)

# Memisahkan kolom durasi waktu belajar dan nilai ujian
TB = df['Hours Studied'].values.reshape(-1, 1)
NT = df['Performance Index'].values.reshape(-1, 1)

# Metode 1 dengan menggunakan Metode Linear
linear_model = LinearRegression()
linear_model.fit(TB, NT)
NT_pred_linear = linear_model.predict(TB)
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))

# Metode 2 dengan menggunakan Model Pangkat Sederhana
log_TB = np.log(TB)
log_NT = np.log(NT)
pangkat_model = LinearRegression()
pangkat_model.fit(log_TB, log_NT)
log_NT_pred_pangkat = pangkat_model.predict(log_TB)
NT_pred_pangkat = np.exp(log_NT_pred_pangkat)
rms_pangkat = np.sqrt(mean_squared_error(NT, NT_pred_pangkat))

# Visualisasi hasil
plt.figure(figsize=(14, 6))

# Plot Model Linear
plt.subplot(1, 2, 1)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Durasi Waktu Belajar (Hours)')
plt.ylabel('Nilai Ujian (Performance Index)')
plt.title('Model Linear')
plt.legend()

# Plot Model Pangkat Sederhana
```

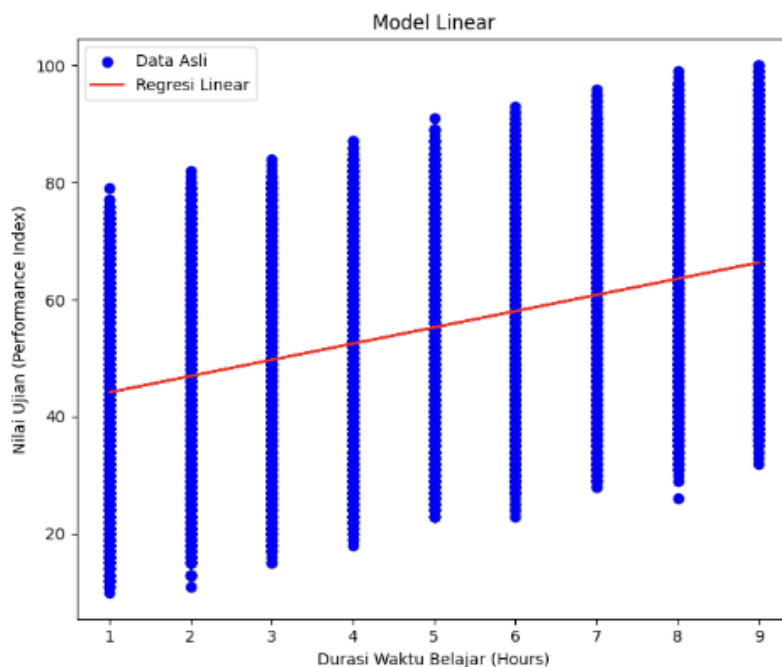
```
plt.subplot(1, 2, 2)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_pangkat, color='green', label='Regresi Pangkat
Sederhana')
plt.xlabel('Durasi Waktu Belajar (Hours)')
plt.ylabel('Nilai Ujian (Performance Index)')
plt.title('Model Pangkat Sederhana')
plt.legend()

plt.tight_layout()
plt.show()

print(f'RMS Model Linear: {rms_linear}')
print(f'RMS Model Pangkat Sederhana: {rms_pangkat}')
```

Hasil Pengujian:

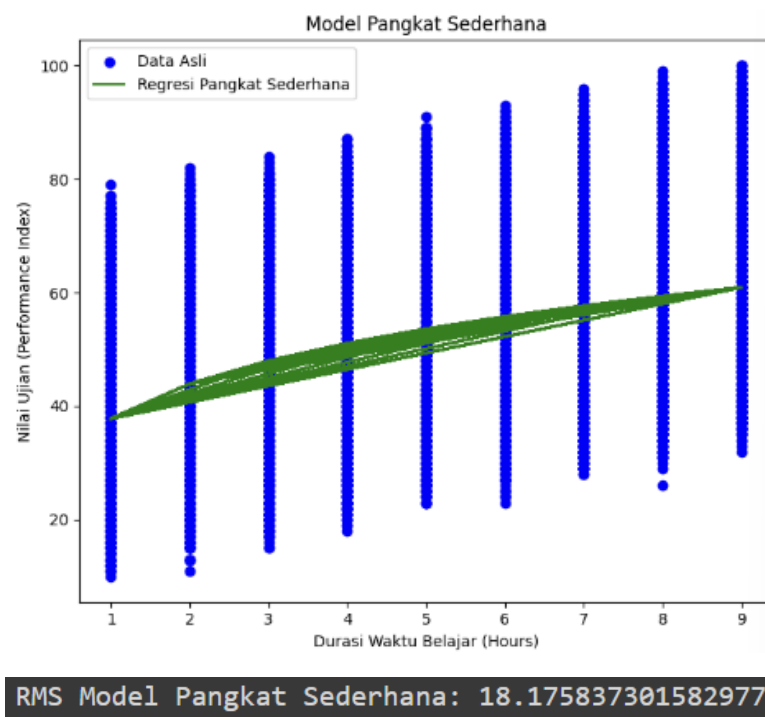
### 1. Metode Regresi Linear



**RMS galat - Regresi Linear: 17.819474832547773**

Grafik di atas menunjukkan hubungan linear positif antara durasi belajar dan nilai ujian, garis ini meminimalkan jumlah kuadrat kesalahan antara data asli dan prediksi model. Model ini juga menunjukkan bahwa ia memiliki tingkat kesalahan yang signifikan sebesar 17.82.

## 2. Metode Regresi Pangkat Sederhana



Berbeda dengan metode regresi linear, metode pangkat sederhana menunjukkan hubungan yang non-linear antara durasi belajar dengan nilai ujian, dan tidak menunjukkan peningkatan signifikan dalam akurasi prediksi. Kemudian, mengenai RMS, model ini memiliki tingkat akurasi yang lebih tinggi dalam memprediksi nilai ujian yaitu sebesar 18.18.

### Analisis Source Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Membaca data dari file CSV
file_path = 'Student_Performance.csv'
df = pd.read_csv(file_path)

# Memisahkan kolom durasi waktu belajar dan nilai ujian
TB = df['Hours Studied'].values.reshape(-1, 1)
NT = df['Performance Index'].values.reshape(-1, 1)
```

Pada bagian tersebut, kita mengimpor pustaka seperti 'numpy', 'pandas', 'matplotlib.pyplot', dan modul dari 'sklearn' untuk analisis data, pemodelan regresi, dan visualisasi. Kemudian, dengan menggunakan 'pandas', data performa siswa dibaca dari file CSV yang akan menghasilkan sebuah DataFrame. Durasi waktu belajar dan nilai ujian

dipisahkan dari DataFrame dan dikonversi menjadi array 'numpy' untuk mempermudah analisis. Dengan masing-masing array menyimpan data yang sesuai. TB array menyimpan data durasi waktu belajar, sedangkan NT array menyimpan data nilai ujian.

```
# Metode 1 dengan menggunakan Metode Linear
linear_model = LinearRegression()
linear_model.fit(TB, NT)
NT_pred_linear = linear_model.predict(TB)
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))
```

Dalam bagian ini, diterapkan metode regresi linear untuk mengungkap hubungan antara durasi waktu belajar (TB) dan nilai ujian (NT). Pertama-tama, membuat objek `LinearRegression` dari pustaka `sklearn`. Selanjutnya, model ini dilatih dengan menggunakan data TB sebagai variabel independen dan NT sebagai variabel dependen melalui metode `fit()`. Setelah itu, memanfaatkannya untuk memprediksi nilai ujian (`NT_pred_linear`) berdasarkan durasi waktu belajar. Terakhir, menghitung galat RMS (Root Mean Square Error) menggunakan fungsi `mean_squared_error()` untuk mengevaluasi akurasi model dengan membandingkan nilai prediksi (`NT_pred_linear`) dengan nilai aktual (NT).

```
# Metode 2 dengan menggunakan Model Pangkat Sederhana
log_TB = np.log(TB)
log_NT = np.log(NT)
pangkat_model = LinearRegression()
pangkat_model.fit(log_TB, log_NT)
log_NT_pred_pangkat = pangkat_model.predict(log_TB)
NT_pred_pangkat = np.exp(log_NT_pred_pangkat)
rms_pangkat = np.sqrt(mean_squared_error(NT, NT_pred_pangkat))
```

Metode kedua, yang disebut Model Pangkat Sederhana yang menggunakan transformasi logaritma untuk menganalisis hubungan non-linier antara durasi waktu belajar (TB) dan nilai ujian (NT). Pertama, data TB dan NT diubah menjadi `log_TB` dan `log_NT`. Selanjutnya, algoritma regresi linear `LinearRegression` dari `sklearn` diterapkan pada data yang telah ditransformasi. Model ini menghasilkan prediksi nilai ujian dalam bentuk logaritma (`log_NT_pred_pangkat`). Nilai ujian aktual (`NT_pred_pangkat`) diperoleh dengan mengembalikan prediksi logaritma ke skala aslinya menggunakan fungsi eksponensial. Terakhir, akurasi model dievaluasi dengan menghitung galat RMS (Root Mean Square Error).

```

# Visualisasi hasil
plt.figure(figsize=(14, 6))

# Plot Model Linear
plt.subplot(1, 2, 1)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Durasi Waktu Belajar (Hours)')
plt.ylabel('Nilai Ujian (Performance Index)')
plt.title('Model Linear')
plt.legend()

# Plot Model Pangkat Sederhana
plt.subplot(1, 2, 2)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_pangkat, color='green', label='Regresi Pangkat Sederhana')
plt.xlabel('Durasi Waktu Belajar (Hours)')
plt.ylabel('Nilai Ujian (Performance Index)')
plt.title('Model Pangkat Sederhana')
plt.legend()

plt.tight_layout()
plt.show()

```

Dua model regresi, yaitu model linear dan model pangkat sederhana, diaplikasikan pada data durasi waktu belajar (TB) dan nilai ujian (NT) siswa. Hasilnya divisualisasikan dalam dua subplot pada figure berukuran 14x6 inci. Subplot pertama, menampilkan data asli (TB vs NT) sebagai titik-titik biru, kemudian menggambarkan prediksi model linear (NT\_pred\_linear) sebagai garis merah, serta dilengkapi label sumbu X dan Y, judul "Model Linear". Sedangkan subplot kedua, menampilkan kembali data asli (TB vs NT) sebagai titik-titik biru, menggambarkan prediksi model pangkat sederhana (NT\_pred\_pangkat) sebagai garis hijau, dan dilengkapi label sumbu X dan Y, judul "Model Pangkat Sederhana". Tata letak subplot diatur dengan rapi menggunakan `plt.tight_layout()` dan visualisasinya ditampilkan dengan `plt.show()`.

```

print(f'RMS Model Linear: {rms_linear}')
print(f'RMS Model Pangkat Sederhana: {rms_pangkat}')

```

Kode tersebut digunakan untuk mencetak Root Mean Squared Error (RMS) dari dua pemodelan untuk mengevaluasi performa tiap model dalam memprediksi nilai ujian (NT) berdasarkan durasi waktu belajar (TB). Tujuannya adalah untuk membandingkan

kesalahan prediksi dari kedua model ini dan menentukan model mana yang paling akurat dalam memodelkan hubungan antara TB dan NT.

#### Analisis Hasil:

Visualisasi yang akan dihasilkan oleh tiga subplot dalam satu gambar pada setiap scatter plot akan menunjukkan data asli dari NT dan TB dengan garis dari masing-masing model regresi. Scatter plot akan menunjukkan hasil dari regresi setiap model. Nilai RMS kemudian dicetak untuk setiap model, yang membantu mengevaluasi akurasi prediksi masing-masing. Analisis ini menunjukkan bahwa dengan membandingkan nilai RMS dari kedua model tersebut, kita bisa menentukan model mana yang paling efektif dalam memprediksi nilai ujian berdasarkan durasi belajar.

#### Kesimpulan:

Kode yang digunakan bertujuan untuk mengevaluasi serta membandingkan dua model regresi yang berbeda dalam memprediksi nilai ujian (NT) berdasarkan durasi waktu belajar (TB). Model regresi linear mengubah menjadi matriks 2D untuk memenuhi format input model dan digunakan untuk memprediksi NT. Model pangkat sederhana menggunakan parameter model yang nantinya akan digunakan untuk memprediksi NT. Root Mean Squared Error (RMS) pada kode digunakan untuk mengukur prediksi untuk setiap model.