

# 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
NT = df['Performance Index'].values

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

# Metode 2 dengan menggunakan Model Pangkat Sederhana
def power_law(x, a, b):
    return a * np.power(x, b)

params_power, _ = curve_fit(power_law, TB, NT)
NT_pred_power = power_law(TB, *params_power)
rms_power = np.sqrt(mean_squared_error(NT, NT_pred_power))

# Metode 3 dengan menggunakan Model Eksponensial
def exponential_model(x, a, b):
    return a * np.exp(b * x)

params_exp, _ = curve_fit(exponential_model, TB, NT, p0=(1, 0.1))
NT_pred_exp = exponential_model(TB, *params_exp)
rms_exp = np.sqrt(mean_squared_error(NT, NT_pred_exp))

# Plot grafik titik data dan hasil regresi
plt.figure(figsize=(18, 6))

plt.subplot(1, 3, 1)
plt.scatter(TB, NT, label='Data Asli')
```

```
plt.plot(TB, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Linear')
plt.legend()

plt.subplot(1, 3, 2)
plt.scatter(TB, NT, label='Data Asli')
plt.plot(TB, NT_pred_power, color='green', label='Regresi Pangkat Sederhana')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Pangkat Sederhana')
plt.legend()

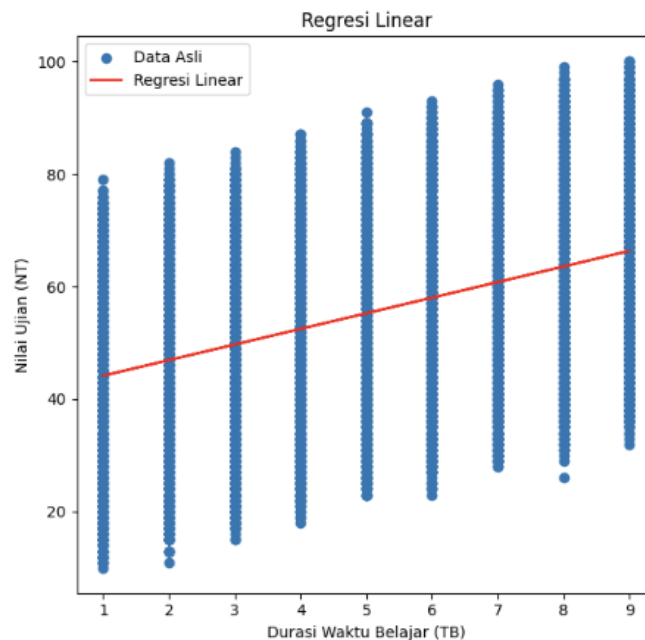
plt.subplot(1, 3, 3)
plt.scatter(TB, NT, label='Data Asli')
plt.plot(TB, NT_pred_exp, color='blue', label='Regresi Eksponensial')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Eksponensial')
plt.legend()

plt.tight_layout()
plt.show()

print(f"RMS galat - Regresi Linear: {rms_linear}")
print(f"RMS galat - Model Pangkat Sederhana: {rms_power}")
print(f"RMS galat - Model Eksponensial: {rms_exp}")
```

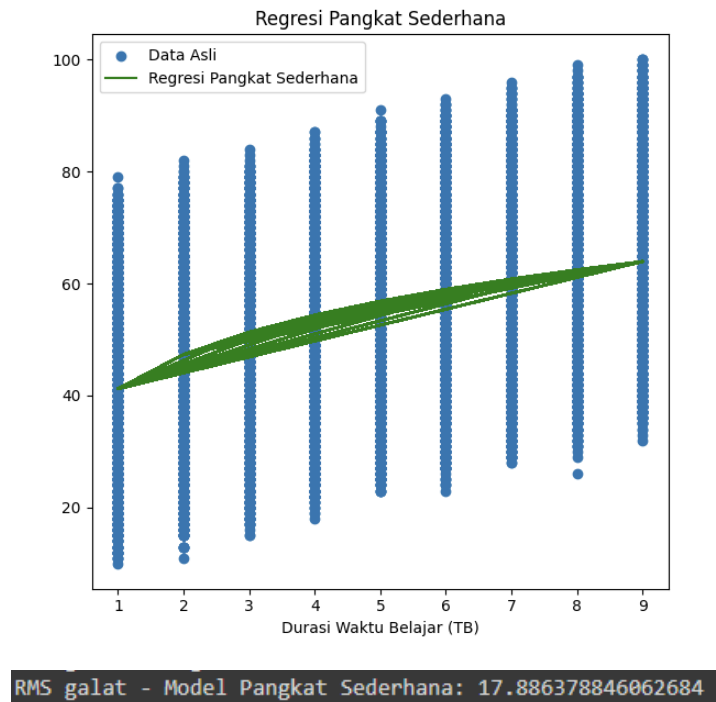
Hasil Grafik Output:

### 1. Metode Regresi Linear

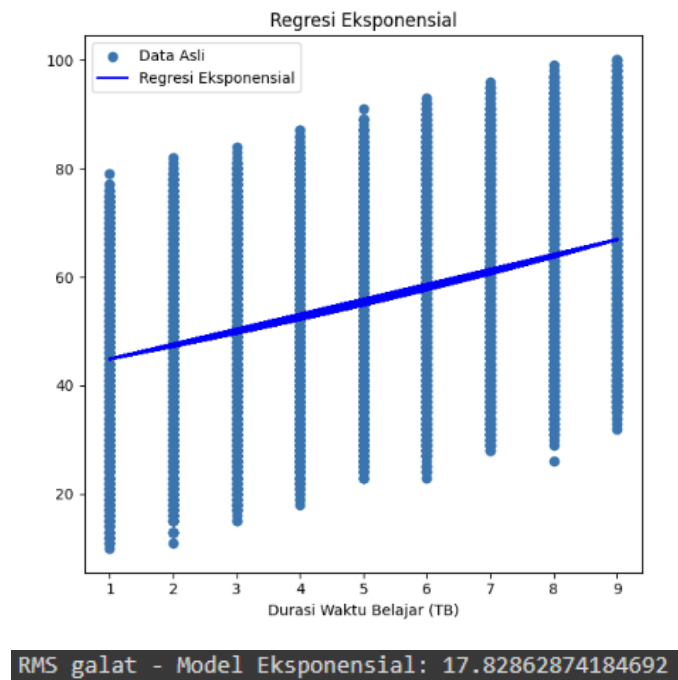


RMS galat - Regresi Linear: 17.819474832547773

## 2. Metode Regresi Pangkat Sederhana



## 3. Metode Regresi Eksponensial



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
NT = df['Performance Index'].values

```

Pada bagian tersebut, kita mengimpor pustaka seperti `numpy`, `pandas`, `matplotlib.pyplot`, dan modul dari `scikit-learn` 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()
TB_reshaped = TB.reshape(-1, 1)
linear_model.fit(TB_reshaped, NT)
NT_pred_linear = linear_model.predict(TB_reshaped)
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))

```

Pada metode pertama ini, model linear diinisialisasi. Kemudian, data durasi waktu belajar (TB) diubah menjadi matriks 2D agar sesuai dengan persyaratan format model. Model linear kemudian disesuaikan dengan data latih TB dan nilai ujian (NT). Hasil prediksi nilai ujian (NT\_pred\_linear) diperoleh dari model yang telah dilatih. Keakuratan model dalam memprediksi nilai ujian dievaluasi dengan menghitung Root Mean Squared Error (rms\_linear).

```

# Metode 2 dengan menggunakan Model Pangkat Sederhana
def power_law(x, a, b):
    return a * np.power(x, b)

params_power, _ = curve_fit(power_law, TB, NT)
NT_pred_power = power_law(TB, *params_power)
rms_power = np.sqrt(mean_squared_error(NT, NT_pred_power))

```

Metode kedua ini digunakan untuk memodelkan hubungan antara durasi waktu belajar (TB) dan nilai ujian (NT). Fungsi power\_law pada metode kedua ini untuk

merepresentasikan model tersebut yang mana a dan b adalah parameter model. Parameter ini disesuaikan dengan menggunakan fungsi `curve_fit` dari `scipy.optimize`. Setelah mendapatkan parameter yang dioptimalkan, nilai ujian diprediksi menggunakan model tersebut (`NT_pred_power`). RMS (Root Mean Squared Error) memberikan ukuran kesalahan prediksi yang dihasilkan oleh model, sehingga dapat digunakan untuk membandingkan akurasi model ini dengan model lainnya.

```
# Metode 3 dengan menggunakan Model Eksponensial
def exponential_model(x, a, b):
    return a * np.exp(b * x)

params_exp, _ = curve_fit(exponential_model, TB, NT, p0=(1, 0.1))
NT_pred_exp = exponential_model(TB, *params_exp)
rms_exp = np.sqrt(mean_squared_error(NT, NT_pred_exp))
```

Sama hal-nya dengan metode kedua, namun metode ketiga menggunakan Model Eksponensial. Fungsi `exponential_model` didefinisikan untuk memodelkan hubungan eksponensial antara TB dan NT. Parameter model (a dan b) diestimasi dengan fungsi `curve_fit`, dengan nilai awal (`p0`) yang ditentukan. Prediksi nilai ujian (`NT_pred_exp`) kemudian dihasilkan menggunakan model eksponensial yang telah disesuaikan. Akurasi model dievaluasi dengan menghitung RMS.

```
plt.subplot(1, 3, 1)
plt.scatter(TB, NT, label='Data Asli')
plt.plot(TB, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
```

Kode ini bertujuan untuk memvisualisasikan data asli dan hasil prediksi model regresi linear. Pertama, menyiapkan subplot. Kemudian, data asli (TB dan NT) ditampilkan sebagai scatter plot dengan titik-titik yang merepresentasikan hubungan antara durasi waktu belajar dan nilai ujian, dan prediksi regresi linear (`NT_pred_linear`) digambarkan sebagai garis merah. Label sumbu x dan y serta legenda digunakan untuk memperjelas grafik, dan menunjukkan hubungan antara durasi belajar dan nilai ujian.

```
print(f"RMS galat - Regresi Linear: {rms_linear}")
print(f"RMS galat - Model Pangkat Sederhana: {rms_power}")
print(f"RMS galat - Model Eksponensial: {rms_exp}")
```

Kode tersebut digunakan untuk mencetak Root Mean Squared Error (RMS) dari tiga pemodelan untuk mengevaluasi performa tiap model dalam memprediksi nilai ujian (NT) berdasarkan durasi waktu belajar (TB). Tujuannya adalah untuk membandingkan kesalahan prediksi dari ketiga model ini dan menentukan model mana yang paling akurat dalam memodelkan hubungan antara TB dan NT.

#### Analisis Hasil:

Kode yang digunakan bertujuan untuk mengevaluasi serta membandingkan tiga 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. Model eksponensial juga menggunakan parameter 'curve\_fit' dan digunakan untuk memprediksi NT. Root Mean Squared Error (RMS) pada kode digunakan untuk mengukur prediksi untuk setiap model.

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 ketiga model tersebut, kita bisa menentukan model mana yang paling efektif dalam memprediksi nilai ujian berdasarkan durasi belajar.