

**PENJELASAN SOURCE CODE MENGENAI APLIKASI
REGRESI UNTUK PEMECAHAN PROBLEM**



NOVENDRA ANUGRAH FITRIATMOKO

21120122130074

METODE NUMERIK

KELAS A

A. Ringkasan

Dokumen merupakan penjelasan dari source code yang dibuat untuk menyelesaikan permasalahan berdasarkan tugas yang diberikan. Dimana permasalahan tersebut yaitu hubungan jumlah latihan soal terhadap nilai ujian. Dalam penyelesaiannya digunakan kode pemrograman menggunakan bahasa Python dengan menggunakan Metode Linier dan Metode Eksponensial serta penghitungan Galat RMS berdasarkan tiap metode yang digunakan.

B. Konsep

- Regresi Linier

Menggunakan persamaan $y = a + bx$ untuk memprediksi nilai y (Nilai Ujian) berdasarkan x (Jumlah Latihan Soal).

- Regresi Eksponensial

Menggunakan persamaan $y = a \times e^{bx}$ untuk memprediksi nilai y (Nilai Ujian) berdasarkan x (Jumlah Latihan Soal).

- Galat RMS

Menggunakan rumus $\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$ untuk menghitung perbedaan nilai yang diprediksi berdasarkan model yang digunakan dengan nilai sebenarnya.

C. Implementasi Kode

Dibawah ini merupakan source code yang digunakan untuk menyelesaikan problem yang telah diberikan.

1. Metode Linier

```
import matplotlib.pyplot as plt

NL = [6, 5, 0, 2, 6, 0, 3, 1, 1, 4, 4, 6, 6, 1, 7, 7, 9, 5, 7, 2]
NT = [61.0, 84.0, 69.0, 27.0, 73.0, 85.0, 74.0, 36.0, 47.0, 74.0,
25.0, 34.0, 57.0, 76.0, 83.0, 31.0, 58.0, 49.0, 73.0, 54.0]

def regresi_linier(x, y):
    n = len(x)
    x_mean = sum(x) / n
    y_mean = sum(y) / n
    b = sum((x[i] - x_mean) * (y[i] - y_mean) for i in range(n)) /
sum((x[i] - x_mean) ** 2 for i in range(n))
```

```

    a = y_mean - b * x_mean
    return a, b

def predict_linier(x, a, b):
    return [a + b * xi for xi in x]

a_linier, b_linier = regresi_linier(NL, NT)
NT_pred_linier = predict_linier(NL, a_linier, b_linier)

def galat_rms(y_true, y_pred):
    n = len(y_true)
    return (sum((y_true[i] - y_pred[i]) ** 2 for i in range(n)) / n)
    ** 0.5

galat_rms_linier = galat_rms(NT, NT_pred_linier)
print(f'Galat RMS (Linier): {galat_rms_linier}')

plt.scatter(NL, NT, color='blue', label='Data')
plt.plot(NL, NT_pred_linier, color='red', label='Linier Fit')
plt.title('Regresi Linier')
plt.xlabel('Jumlah Latihan Soal')
plt.ylabel('Nilai Ujian')
plt.legend()
plt.show()

```

Penjelasan Source Code:

- Pertama yang dilakukan adalah meng-import library “matplotlib.pyplot as plt” yang nantinya digunakan untuk membuat grafik.
- Selanjutnya membuat mendeklarasikan data yang akan digunakan dimana “NL” adalah jumlah latihan soal dan “NT” adalah nilai ujian.
- Kemudian membuat function “def regresi_linier(x, y)” untuk menghitung regresi linier dimana ‘n’ merupakan jumlah data, ‘x_mean’ dan ‘y_mean’ merupakan rata-rata dari nilai ‘x’ dan ‘y’, ‘b’ merupakan nilai kemiringan dari regresi, serta ‘a’ merupakan intersep dari garis regresi.
- Function “def predict_linier(x, a, b)” berfungsi untuk menghasilkan nilai prediksi ‘y’ yang didasarkan dari input ‘x’ dan parameter regresinya yaitu ‘a’ dan ‘b’.

- Selanjutnya `"a_linier, b_linier = regresi_linier(NL, NT)"` dimana `'a_linier'` dan `'b_linier'` merupakan parameter regresi linier yang dapat dihitung menggunakan function `"def predict_linier(x, a, b)"`. Sementara `"NT_pred_linier = predict_linier(NL, a_linier, b_linier)"` dimana `'NT'` merupakan nilai prediksi yang didasarkan perhitungan jumlah latihan soal dan parameter regresi yang telah dihitung.
- Lalu membuat function `"def galat_rms(y_true, y_pred)"` untuk menghitung Galat RMS antara nilai `'y_true'` dan `'y_pred'`.
- Selanjutnya mebuat kode untuk mencetak atau menampilkan Galat RMS yang telah dihitung `"print(f'Galat RMS (Linier): {galat_rms_linier}')"`.
- Langkah terakhir adalah membuat kode untuk menampilkan grafik, dimana `"plt.scatter"` untuk menampilkan data asli sebagai titik-titik berwarna biru, `"plt.plot"` untuk menggambar garis regresi linier berdasarkan nilai prediksi yang dihitung, `"plt.title, plt.xlabel, dan plt.ylabel"` untuk menambahkan judul dan label pada sumbu grafik, `"plt.legend"` untuk menambahkan legenda untuk membedakan data asli dan garis regresi, `"plt.show()"` untuk menampilkan grafik.

2. Metode Eksponensial

```
import math
import matplotlib.pyplot as plt

NL = [6, 5, 0, 2, 6, 0, 3, 1, 1, 4, 4, 6, 6, 1, 7, 7, 9, 5, 7, 2]
NT = [61.0, 84.0, 69.0, 27.0, 73.0, 85.0, 74.0, 36.0, 47.0, 74.0,
25.0, 34.0, 57.0, 76.0, 83.0, 31.0, 58.0, 49.0, 73.0, 54.0]

def regresi_eksponensial(x, y):
    n = len(x)
    log_y = [math.log(yi) for yi in y]
    x_mean = sum(x) / n
    log_y_mean = sum(log_y) / n
    b = sum((x[i] - x_mean) * (log_y[i] - log_y_mean) for i in
range(n)) / sum((x[i] - x_mean) ** 2 for i in range(n))
    a = log_y_mean - b * x_mean
```

```

    return math.exp(a), b

def predict_eksponensial(x, a, b):
    return [a * math.exp(b * xi) for xi in x]

a_eksponensial, b_eksponensial = regresi_eksponensial(NL, NT)
NT_pred_eksponensial = predict_eksponensial(NL, a_eksponensial,
b_eksponensial)

def galat_rms(y_true, y_pred):
    n = len(y_true)
    return (sum((y_true[i] - y_pred[i]) ** 2 for i in range(n)) / n)
** 0.5

galat_rms_eksponensial = galat_rms (NT, NT_pred_eksponensial)
print(f'Galat RMS (Exponential): {galat_rms_eksponensial}')

plt.scatter(NL, NT, color='blue', label='Data')
plt.plot(NL, NT_pred_eksponensial, color='green', label='Exponential
Fit')
plt.title('Regresi Eksponensial')
plt.xlabel('Jumlah Latihan Soal')
plt.ylabel('Nilai Ujian')
plt.legend()
plt.show()

```

Penjelasan Source Code

- Pertama yaitu meng-import library “math” untuk melakukan perhitungan matematika dan “matplotlib.pyplot as plt” untuk membuat grafik.
- Selanjutnya mendeklarasikan data yang akan digunakan dimana ‘NL’ merupakan jumlah latihan soal dan ‘NT’ merupakan nilai ujian.
- Kemudian membuat function “def regresi_eksponensial(x, y)” untuk menghitung regresi eksponensial yang dicari. Dimana ‘n’ merupakan jumlah data, ‘log_y’ merupakan logaritma dari ‘y’ atau nilai ujian ‘x_mean’ dan ‘log_y_mean’ merupakan rata-rata dari nilai ‘x’ dan ‘log_y’, ‘b’ merupakan model regresi untuk kemiringan dari garis regresi terhadap logaritma dari ‘y’, serta ‘a’ merupakan model regresi yang dihitung dalam bentuk logaritma dan dikonversi menjadi nilai sebenarnya menggunakan “math.exp”.

- Selanjutnya function “predict_esponensial” menghasilkan nilai prediksi ‘y’ berdasarkan input ‘x’ dan parameter regresi ‘a’ dan ‘b’ dalam model eksponensial.
- Selanjutnya “a_eksponensial, b_eksponensial = regresi_eksponensial(NL, NT)” dimana ‘a_eksponensial’ dan ‘b_eksponensial’ merupakan parameter regresi eksponensial yang dapat dihitung menggunakan function “def regresi_eksponensial(x, y”. Sementara “NT_pred_eksponensial = predict_exponential(NL, a_eksponensial, b_eksponensial)” dimana ‘NT’ merupakan nilai prediksi yang didasarkan perhitungan jumlah latihan soal dan parameter regresi yang telah dihitung.
- Lalu membuat function “def galat_rms(y_true, y_pred)” untuk menghitung Galat RMS antara nilai ‘y_true’ dan ‘y_pred’.
- Selanjutnya mebuat kode untuk mencetak atau menampilkan Galat RMS yang telah dihitung “print(f'Galat RMS (Exponential): {galat_rms_eksponensial}'))”.
- Langkah terakhir adalah membuat kode untuk menampilkan grafik, dimana “plt.scatter” untuk menampilkan data asli sebagai titik-titik berwarna biru, “plt.plot” untuk menggambar garis regresi linier berdasarkan nilai prediksi yang dihitung, “plt.title, plt.xlabel, dan plt.ylabel” untuk menambahkan judul dan label pada sumbu grafik, “plt.legend” untuk menambahkan legenda untuk membedakan data asli dan garis regresi, “plt.show()” untuk menampilkan grafik.

D. Kode Testing

```
import math
import matplotlib.pyplot as plt
from MetodeLinier import regresi_linier, predict_linier, galat_rms
from MetodeEksponensial import regresi_eksponensial,
predict_eksponensial,galat_rms

NL = [6, 5, 0, 2, 6, 0, 3, 1, 1, 4, 4, 6, 6, 1, 7, 7, 9, 5, 7, 2]
NT = [61.0, 84.0, 69.0, 27.0, 73.0, 85.0, 74.0, 36.0, 47.0, 74.0, 25.0,
34.0, 57.0, 76.0, 83.0, 31.0, 58.0, 49.0, 73.0, 54.0]

#Testing Regresi Linier
```

```

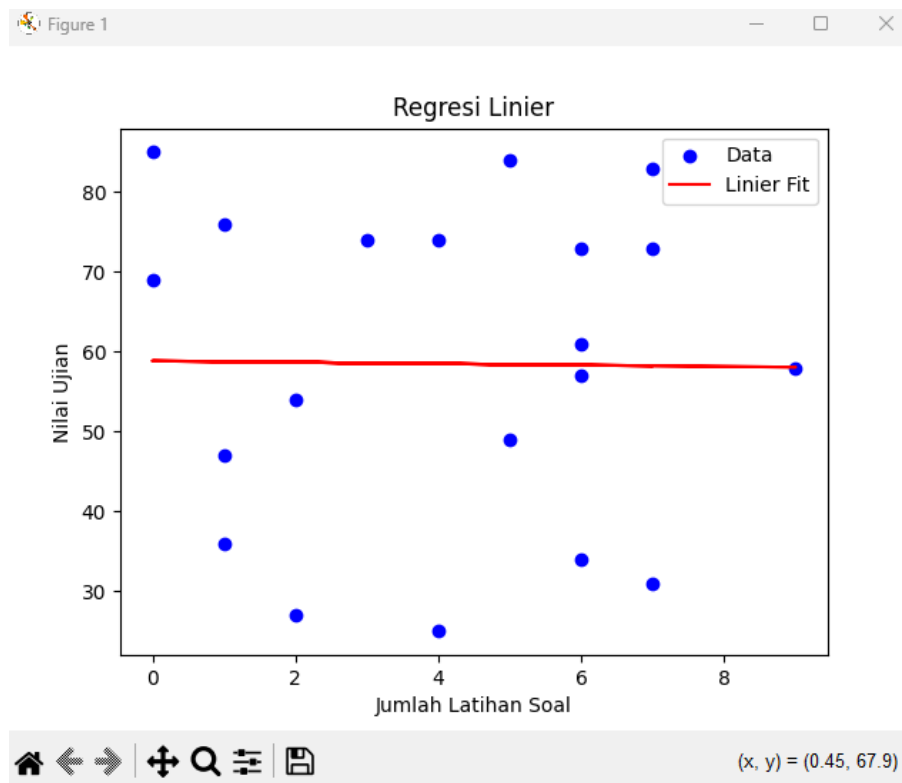
a_linier, b_linier = regresi_linier(NL, NT)
NT_pred_linier = predict_linier(NL, a_linier, b_linier)
galat_rms_linear = galat_rms(NT, NT_pred_linier)
print(f'Galat RMS (Linear): {galat_rms_linear}')

#Testing Regresi Eksponensial
a_eksponensial, b_eksponensial = regresi_eksponensial(NL, NT)
NT_pred_eksponensial = predict_eksponensial(NL, a_eksponensial,
b_eksponensial)
galat_rms_eksponensial = galat_rms(NT, NT_pred_eksponensial)
print(f'Galat RMS (Exponential): {galat_rms_eksponensial}')

```

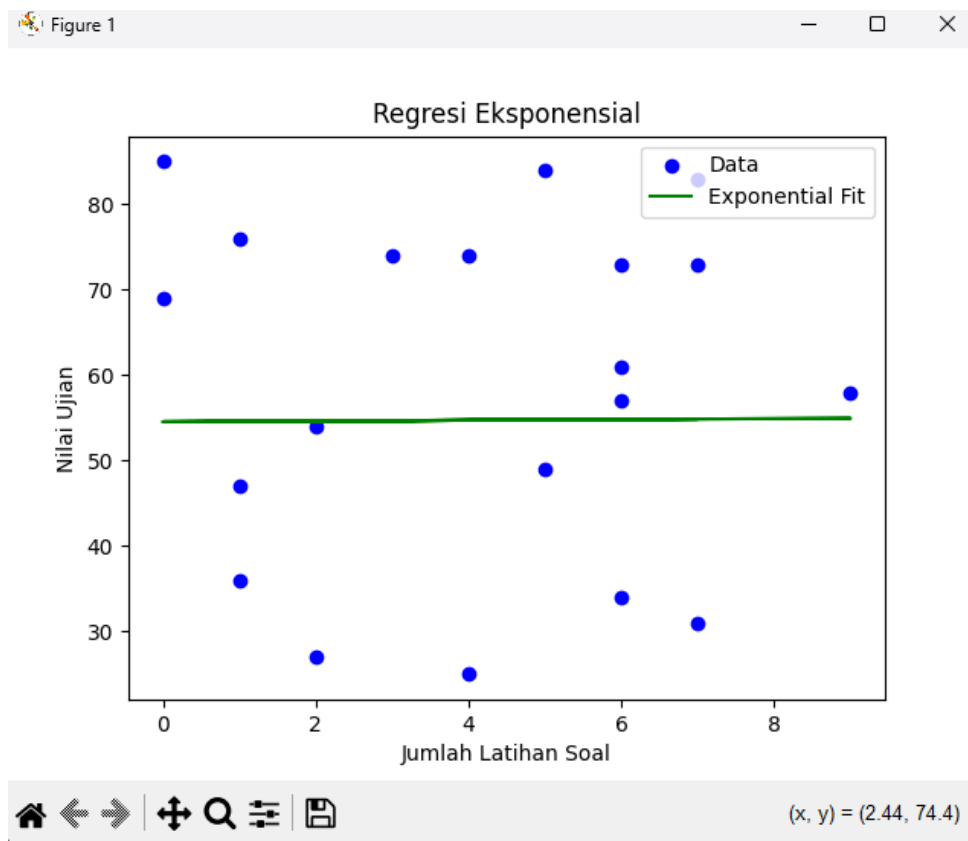
E. Hasil Pengujian

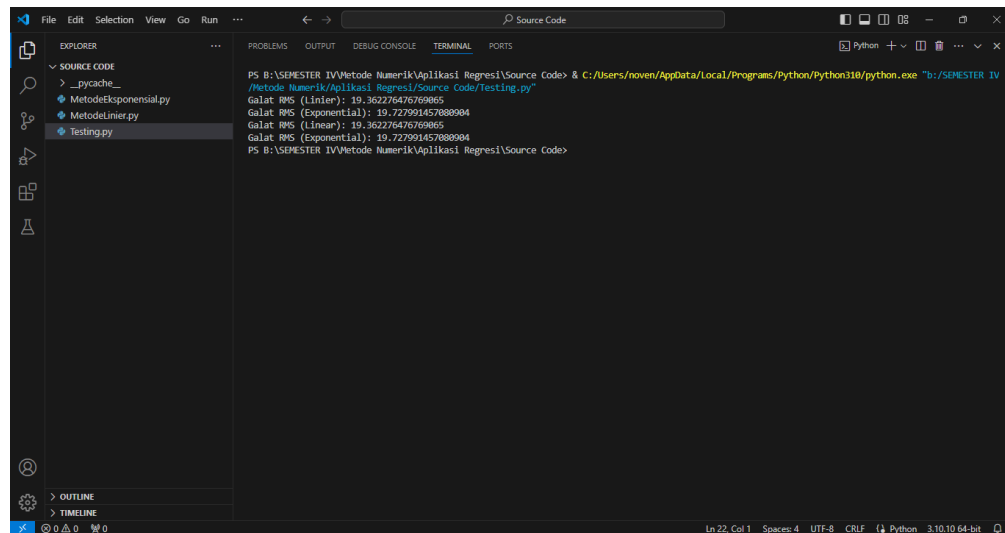
1. Metode Linier



```
PS B:\SEMESTER IV\Metode Numerik\Aplikasi Regresi\Source Code> & C:/Users/noven/AppData/Local/Programs/Python/Python310/python.exe "b:/SEMESTER IV /Metode Numerik/Aplikasi Regresi/Source Code/testing.py"
Galat RMS (Linear): 19.3622764769865
Galat RMS (Exponential): 19.727991457888984
Galat RMS (Linear): 19.3622764769865
Galat RMS (Exponential): 19.727991457888984
PS B:\SEMESTER IV\Metode Numerik\Aplikasi Regresi\Source Code>
```

2. Metode Eksponensial





```
PS B:\SEMESTER IV\Metode Numerik\Aplikasi Regresi\Source Code> & C:/Users/noven/AppData/Local/Programs/Python/Python310/python.exe "b:/SEMESTER IV
/Metode Numerik/Aplikasi Regresi/Source Code/Testing.py"
Galat RMS (Linier): 19.362276476769065
Galat RMS (Exponential): 19.727991457080904
Galat RMS (Linear): 19.362276476769065
Galat RMS (Exponential): 19.727991457080904
PS B:\SEMESTER IV\Metode Numerik\Aplikasi Regresi\Source Code>
```

F. Analisa Hasil

- Metode Linier menyatakan hubungan secara langsung atau linier antara jumlah latihan soal dengan nilai ujian.
- Metode Eksponensial menyatakan hubungan secara non-linier antara jumlah latihan soal dengan nilai ujian.
- Pemodelan yang memiliki Galat RMS lebih kecil dapat dianggap lebih baik dalam memprediksi nilai ujian berdasarkan jumlah latihan soal.