

Ahmad Nayef Al Fito  
082111733037  
Komputasi Biomedis T1

Link Google Colab :

[Tugas 2\\_082111733037\\_Ahmad Nayef Al Fito\\_Komputasi Biomedis](#)

Contoh Soal :

Tentukan S, a, dan b sehingga  $f(x) = a+bx$  sesuai dengan data berikut,

X	0.0	1.0	2.0	2.5	3.0
Y	2.9	3.7	4.1	4.4	5.0

Kode Program :

```
# Nama : Ahmad Nayef Al Fito
# NIM : 082111733037
# Kelas : Komputasi Biomedis T1

# contoh soal

# Deklarasi awal
import numpy as np
import pandas as pd
from numpy import *
import pylab

# Memasukkan data
x = np.array([0.0, 1.0, 2.0, 2.5, 3.0])
y = np.array([2.9, 3.7, 4.1, 4.4, 5.0])
n = len(x)

# Menghitung nilai xy dan x^2
xy = x * y
x_squared = x ** 2

# Menghitung nilai a dan b menggunakan regresi linier
mean_x = np.mean(x)
mean_y = np.mean(y)
b = np.sum((x - mean_x) * (y - mean_y)) / np.sum((x - mean_x) ** 2)
a = mean_y - b * mean_x
nilai_s = (y - a - b * x) ** 2

# Mengatur tampilan dataframe
```

```

data = {'No.': range(1, len(x) + 1), 'x': x, 'y': y, 'xy': xy, 'x^2':
x_squared, 's ke i' : nilai_s}
df = pd.DataFrame(data)

# Menghitung jumlah nilai s
S = np.sum(nilai_s)

# Menghitung jumlah x, jumlah y, jumlah xy, dan jumlah x^2
jumx=np.sum(x)
jumy=np.sum(y)
jumxy=np.sum(xy)
jumx_squared=np.sum(x_squared)

#C^-1.D
M = np.array([[n,jumx],[jumx,jumx_squared]])
N = np.array([[jumy],[jumxy]])
ab = np.dot(np.linalg.inv(M),N)
ybaru = zeros(n,float)
for i in range (0,n):
    ybaru[i] = ab[1]*x[i]+ab[0]

# Menampilkan nilai a dan b
print ('y = bX + a')
print("Bentuk Persamaan Regresi Liniernya adalah: f(x) = ",a,"+",b,"x"
)
print("Nilai a:", a)
print("Nilai b:", b)
print('\n')

# Mendampilkan DataFrame
print('Tabel regresi linear dari data:')
print(df)
print("Jumlah X:", jumx)
print("Jumlah Y:", jumy)
print("Jumlah XY:", jumxy)
print("Jumlah X^2:", jumx_squared)
print('\n')

# Menampilkan nilai a, b dan S
print('Maka, ')
print("Nilai a:", a)
print("Nilai b:", b)

```

```

print("Nilai S:", S)
print('\n')

# Menampilkan grafik hasil regresi
print('Grafik plot data:')
pylab.plot(x,y,'og')
pylab.plot(x,ybaru,'-y')
print('Keterangan : Garis kuning = hasil regresi linear')

```

Hasil Run :

```

y = bX + a
Bentuk Persamaan Regresi Liniernya adalah:  $f(x) = 2.926724137931035 + 0.643103448275862 x$ 
Nilai a: 2.926724137931035
Nilai b: 0.643103448275862

Tabel regresi linear dari data:

```

No.	x	y	xy	x^2	s ke i	
0	1	0.0	2.9	0.0	0.00	0.000714
1	2	1.0	3.7	3.7	1.00	0.016945
2	3	2.0	4.1	8.2	4.00	0.012753
3	4	2.5	4.4	11.0	6.25	0.018086
4	5	3.0	5.0	15.0	9.00	0.020726

```

Jumlah X: 8.5
Jumlah Y: 20.1
Jumlah XY: 37.9
Jumlah X^2: 20.25

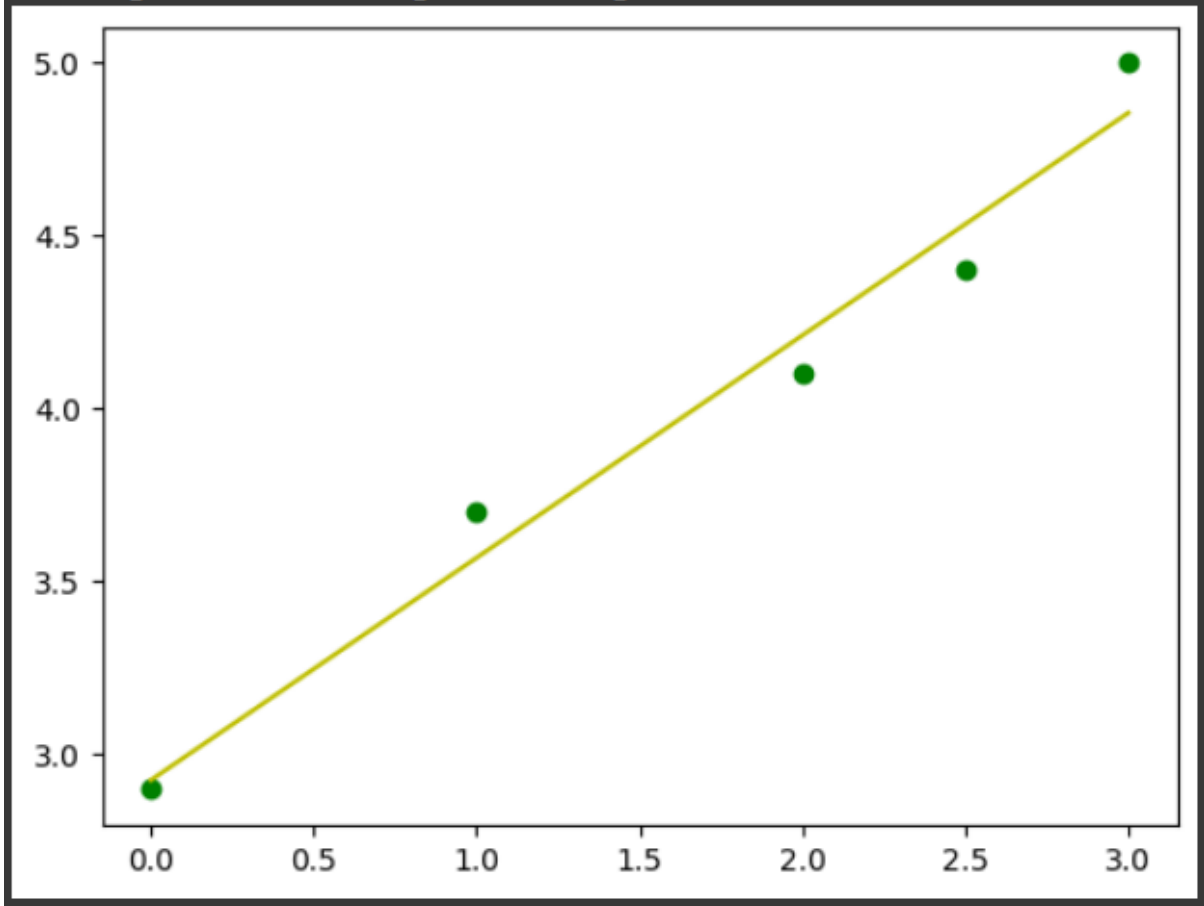
Maka,
Nilai a: 2.926724137931035
Nilai b: 0.643103448275862
Nilai S: 0.06922413793103449

```

Hasil Plot :

Grafik plot data:

Keterangan : Garis kuning = hasil regresi linear



Soal Latihan

1. Tentukan S, a, dan b sehingga  $f(x) = ax^b$  sesuai dengan data berikut,

X	0.5	1.0	1.5	2.0	2.5
Y	0.49	1.60	3.36	6.44	10.16

Gunakan persamaan linear  $\ln(f(x)) = \ln(a) + b \ln(x)$  untuk menggantikan persamaan  $f(x) = ax^b$

Kode Program :

```
# Nama : Ahmad Nayef Al Fito
# NIM : 082111733037
# Kelas : Komputasi Biomedis T1

# Soal 1

# Deklarasi awal
import numpy as np
import pandas as pd
from numpy import *
```

```

import pylab
import matplotlib.pyplot as plt
from scipy.stats import linregress

# Memasukkan data
x = np.array([0.5, 1.0, 1.5, 2.0, 2.5])
y = np.array([0.49, 1.60, 3.36, 6.44, 10.16])

# Menyesuaikan data dengan persamaan regresi linier
X = np.log(x)
Y = np.log(y)

# Menghitung nilai xy, y^2 dan x^2
XY = X * Y
X_squared = X ** 2
Y_squared = Y ** 2

# Menghitung nilai a dan b menggunakan regresi linier
mean_X = np.mean(X)
mean_Y = np.mean(Y)
b = np.sum((X - mean_X) * (Y - mean_Y)) / np.sum((X - mean_X) ** 2)
A = mean_Y - b * mean_X
a = np.exp(A)
nilai_s = (Y - A - b * X) ** 2

# Mengatur tampilan dataframe
data = {'No.': range(1, len(x) + 1), 'x': x, 'y': y, 'ln(x)': X,
        'ln(y)': Y, 'ln(x)*ln(y)': XY, '(ln(x))^2': X_squared, '(ln(y))^2':
        Y_squared, 's ke i' : nilai_s}
df = pd.DataFrame(data)

# Menghitung jumlah nilai s
S = np.sum(nilai_s)

# Menghitung jumlah x, jumlah y, jumlah ln(x), jumlah ln(y), jumlah
ln(x)ln(y), jumlah (lny)^2 dan jumlah (lnx)^2
jumx=np.sum(x)
jumy=np.sum(y)
jumX=np.sum(X)
jumY=np.sum(Y)
jumXY=np.sum(XY)
jumX_squared=np.sum(X_squared)
jumY_squared=np.sum(Y_squared)

```

```

# Menampilkan nilai a dan b
print ('Y = bX+A')
print ('ln(f(x))= b(ln(x))+ln(a)')
print("Bentuk Persamaan Regresi Liniernya adalah: ln(f(x)) =
",A,"+",b,"ln(x)" )
print("Nilai a:", a)
print("Nilai ln(a):", A)
print("Nilai b:", b)
print('\n')

# Mendampilkan DataFrame
print('Tabel regresi linear dari data:')
print(df)
print("Jumlah x:", jumx)
print("Jumlah y:", jumy)
print("Jumlah lnx:", jumX)
print("Jumlah lny:", jumY)
print("Jumlah ln(x)ln(y):", jumXY)
print("Jumlah (lnx)^2:", jumX_squared)
print("Jumlah (lny)^2:", jumY_squared)
print('\n')

# Menampilkan nilai a, b dan S
print('Maka, ')
print("Nilai a:", a)
print("Nilai b:", b)
print("Nilai ln(a):", A)
print("Nilai S:", S)
print('\n')

print('Grafik plot data:')

# Melakukan Plot Sebelum Linearisasi
plt.figure(figsize=(12, 4))
plt.subplot(131)
plt.scatter(x, y, label="Data Sebelum Linearisasi")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()

# Melakukan Plot Sesudah Linearisasi

```

```
plt.subplot(132)
plt.scatter(X, Y, label="Data Sesudah Linearisasi")
plt.xlabel("ln(X)")
plt.ylabel("ln(Y)")
plt.legend()

# Melakukan Plot Hasil Regresi Linear
plt.subplot(133)
plt.scatter(X, Y, label="Data Linearisasi")
x_fit = np.linspace(min(X), max(X), 100)
y_fit = A + b * x_fit
plt.plot(x_fit, y_fit, color="red", label="Regresi Linier")
plt.xlabel("ln(x)")
plt.ylabel("ln(y)")
plt.legend()

plt.tight_layout()
plt.show()
```

Hasil Run :

```

Y = bX+A
ln(f(x))= b(ln(x))+ln(a)
Bentuk Persamaan Regresi Liniernya adalah: ln(f(x)) = 0.532473481198709 + 1.8817501813721564 ln(x)
Nilai a: 1.7031397871561753
Nilai ln(a): 0.532473481198709
Nilai b: 1.8817501813721564

Tabel regresi linear dari data:

```

No.	x	y	ln(x)	ln(y)	ln(x)*ln(y)	(ln(x))^2	(ln(y))^2	\
0	1	0.5	0.49	-0.693147	-0.713350	0.494456	0.480453	0.508868
1	2	1.0	1.60	0.000000	0.470004	0.000000	0.000000	0.220903
2	3	1.5	3.36	0.405465	1.211941	0.491400	0.164402	1.468801
3	4	2.0	6.44	0.693147	1.862529	1.291006	0.480453	3.469013
4	5	2.5	10.16	0.916291	2.318458	2.124382	0.839589	5.375250

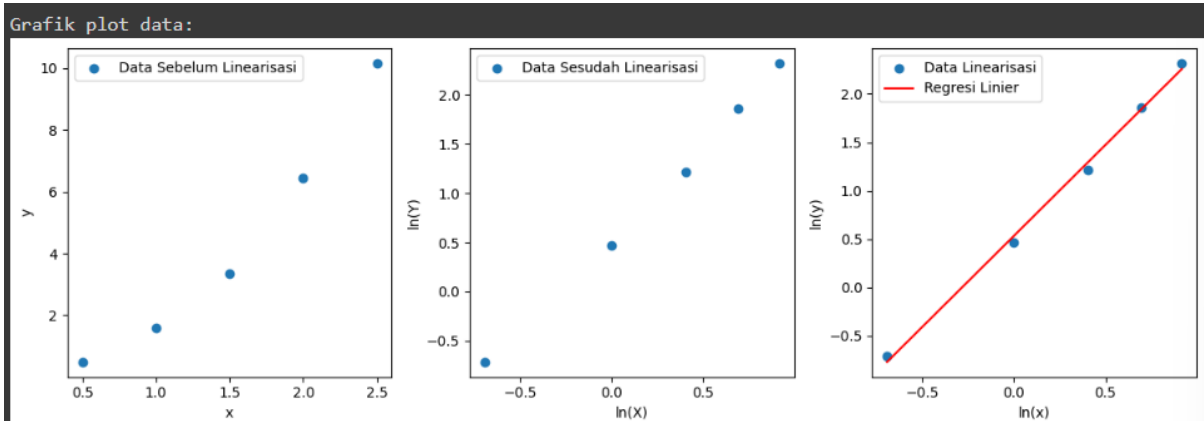
```

s ke i
0 0.003423
1 0.003902
2 0.006975
3 0.000662
4 0.003814
Jumlah x: 7.5
Jumlah y: 22.05
Jumlah lnx: 1.3217558399823195
Jumlah lny: 5.149581697609982
Jumlah ln(x)ln(y): 4.401244630640314
Jumlah (lnx)^2: 1.9648966870480429
Jumlah (lny)^2: 11.042834509164479

Maka,
Nilai a: 1.7031397871561753
Nilai b: 1.8817501813721564
Nilai ln(a): 0.532473481198709
Nilai S: 0.018775933950291812

```

Hasil Plot :



2. Tentukan S, a, dan b sehingga  $\gamma(t) = ae^{-bt}$  sesuai dengan data berikut,

t	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5
$\gamma$	1.00	0.94 4	0.99 0	0.98 5	0.97 79	0.97 7	0.97 2	0.96 9	0.96 7	0.96 0	0.95 6	0.95 2



Gunakan persamaan linear  $\ln(\gamma(t)) = \ln(a) - bt$  untuk menggantikan persamaan

$$\gamma(t) = ae^{-bt}$$

Kode Program :

```
# Nama : Ahmad Nayef Al Fito
# NIM : 082111733037
# Kelas : Komputasi Biomedis T1

# Soal 2

# Deklarasi awal
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import linregress

# Memasukkan data
t = np.array([0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5])
y = np.array([1.000, 0.994, 0.990, 0.985, 0.979, 0.977, 0.972, 0.969, 0.967, 0.960, 0.956, 0.952])

# Menyesuaikan data dengan persamaan regresi linier
Y = np.log(y)

# Menghitung nilai tY, y^2 dan x^2
tY = t * Y
t_squared = t ** 2
Y_squared = Y ** 2

# Menghitung nilai a dan b menggunakan regresi linier
slope, intercept, r_value, p_value, std_err = linregress(t, Y)
a = np.exp(intercept)
b = -slope

# Mengatur tampilan dataframe
data = {'No.': range(1, len(t) + 1), 't': t, 'y': y, 'ln(y)': Y, 't*ln(y)': tY, 't^2': t_squared, '(ln(y))^2': Y_squared}
df = pd.DataFrame(data)

# Menghitung jumlah nilai s
nilai_s = (Y - intercept - slope * t) ** 2
S = np.sum(nilai_s)
```

```

# Menampilkan nilai a dan b
print('Y = BX+A')
print('ln(y(t))= -bt + ln(a)')
print("Bentuk Persamaan Regresi Liniernya adalah: ln(y(t)) = {:.4f} -
({:.4f})t".format(intercept, slope))
print("Nilai a:", a)
print("Nilai b:", b)
print('\n')

# Menampilkan DataFrame
print('Tabel regresi linear dari data:')
print(df)
print('\n')

# Menampilkan nilai a, b dan S
print('Maka, ')
print("Nilai a:", a)
print("Nilai b:", b)
print("Nilai S:", S)
print('\n')

# Plot data dan hasil regresi
print('Grafik plot data:')
plt.scatter(t, Y, label="Data")
plt.plot(t, intercept + slope * t, color="red", label="Regresi Linier")
plt.xlabel("t")
plt.ylabel("ln(y)")
plt.legend()
plt.show()

```

Hasil Run :

$$Y = BX+A$$

$$\ln(y(t)) = -bt + \ln(a)$$

Bentuk Persamaan Regresi Liniernya adalah:  $\ln(y(t)) = -0.0016 - (-0.0086)t$

Nilai a: 0.9984157812833949

Nilai b: 0.008639549701453635

Tabel regresi linear dari data:

	No.	t	y	ln(y)	t*ln(y)	t^2	(ln(y))^2
0	1	0.0	1.000	0.000000	0.000000	0.00	0.000000
1	2	0.5	0.994	-0.006018	-0.003009	0.25	0.000036
2	3	1.0	0.990	-0.010050	-0.010050	1.00	0.000101
3	4	1.5	0.985	-0.015114	-0.022670	2.25	0.000228
4	5	2.0	0.979	-0.021224	-0.042447	4.00	0.000450
5	6	2.5	0.977	-0.023269	-0.058172	6.25	0.000541
6	7	3.0	0.972	-0.028399	-0.085198	9.00	0.000807
7	8	3.5	0.969	-0.031491	-0.110217	12.25	0.000992
8	9	4.0	0.967	-0.033557	-0.134227	16.00	0.001126
9	10	4.5	0.960	-0.040822	-0.183699	20.25	0.001666
10	11	5.0	0.956	-0.044997	-0.224987	25.00	0.002025
11	12	5.5	0.952	-0.049190	-0.270546	30.25	0.002420

Maka,

Nilai a: 0.9984157812833949

Nilai b: 0.008639549701453635

Nilai S: 1.623955062084946e-05

Hasil Plot :

Grafik plot data:

