Nama : Mutiara Khairunnisa

NIM : 23/517062/PA/22149

TUGAS 3 PRA-UTS GEOSTATISTIKA

TASK 1

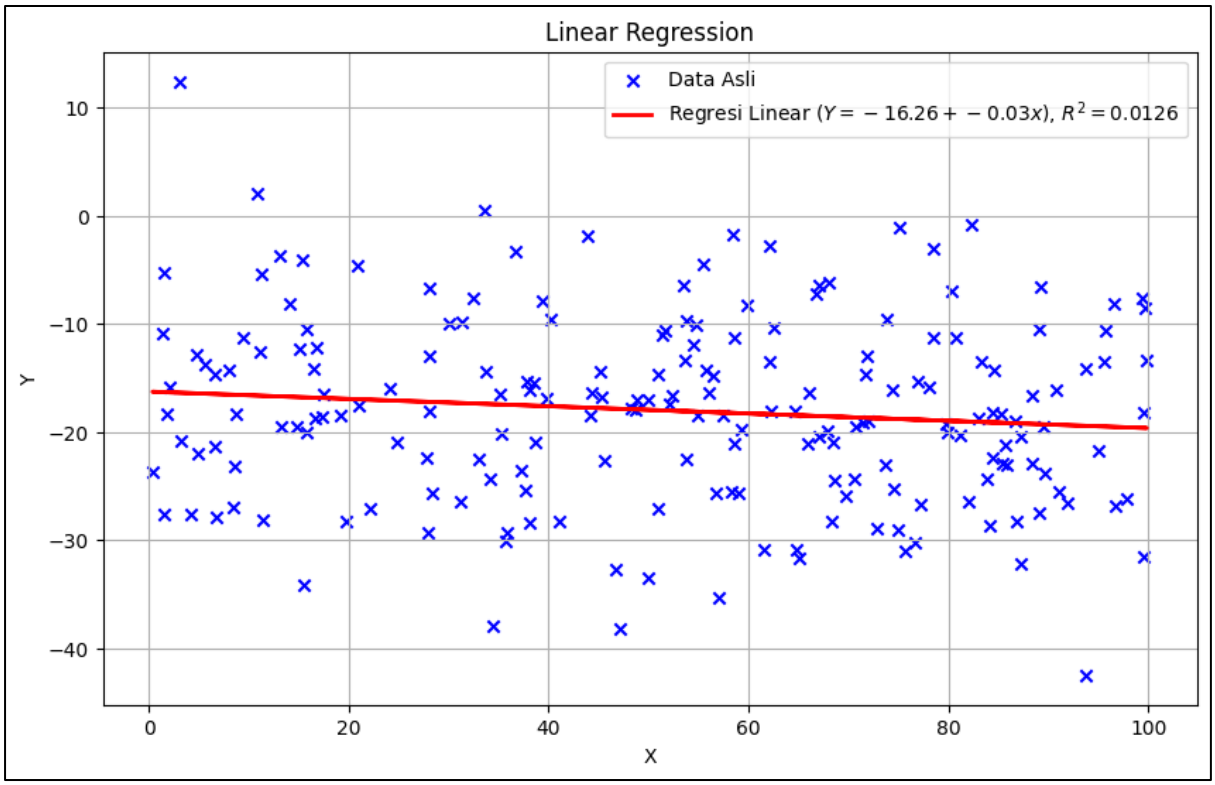
Pada task 1, diberikan dataset yang berisi sejumlah 200 angka *random* dengan variabel X dan Y yang diperoleh dari web program *octave.online.* Dengan dataset tersebut, diminta untuk menerapkan regresi linear dengan program apapun, termasuk pemrograman python.

Proses analisis regresi linear dilakukan dengan beberapa langkah utama. Pertama, hitung nilai rata-rata dari variabel X dan Y, kemudian digunakan untuk menghitung *covariance* dan *Pearson correlation coefficient* (r_{xy}). *Covariance* digunakan untuk menunjukkan hubungan antara kedua variabel (X dan Y), sedangkan *Pearson correlation coefficient* (r_{xy}) menunjukkan seberapa erat hubungan linear antara kedua variabel dan arahnya (naik atau turun). *Covariance* dapat dihitung dengan persamaan $cov(X,Y) = \frac{\Sigma(X-\bar{X})(Y-\bar{Y})}{n-1}$. Sedangkan *Pearson correlation coefficient*dapat dihitung dengan persamaan $r_{xy} = \frac{\Sigma(X-\bar{X})(Y-\bar{Y})}{\sqrt{\Sigma(X-\bar{X})^2}.\sqrt{\Sigma(Y-\bar{Y})^2}}$ atau $r_{xy} = \frac{cov(X,Y)}{\sqrt{\Sigma(X-\bar{X})^2}.\sqrt{\Sigma(Y-\bar{Y})^2}}$.

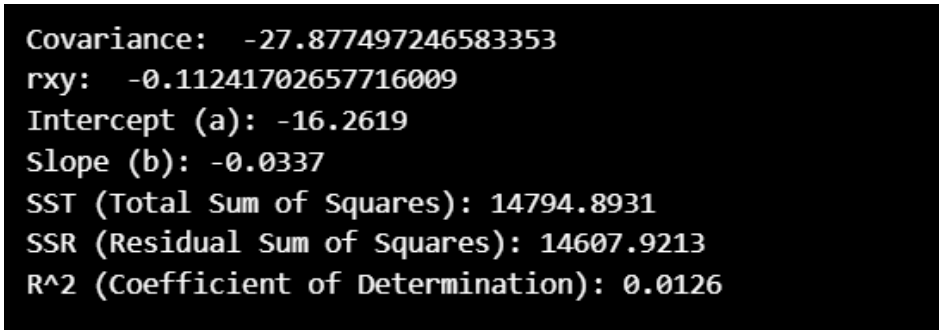
Kemudian, parameter regresi dihitung menggunakan metode *Least Squares* dengan menambahkan kolom bias ($X_0 = 1$) ke matriks X. Koefisien regresi (β) dihitung untuk mendapatkan *intercept(a)* dan *slope (b)* dengan menggunakan persamaan $\beta = (X^T X)^{-1} X^T Y$. Dengan persamaan tersebut, dapat diperoleh nilai *a* dan *b* terbaik denfan meminimalkan selisih antara nilai prediksi dan data aktual. Selanjutnya, nilai koefisien determinasi (R^2) dihitung untuk mengukur seberapa baik model regresi dalam menjelaskan variasi dalam data. Nilai R^2 dapat dihitung dengan persamaan $R^2 = 1 - \frac{\sum(Y-\hat{Y})^2}{\sum(Y-\bar{Y})^2}$. Proses dan data asli beserta regresi linearnya ditampilkan sebagaimana gambar berikut.

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

df = pd.read_csv('GEOSTAT 3.csv')
X = df['X']
Y = df['Y']

X_mean = np.mean(X)
Y_mean = np.mean(Y)

numerator = np.sum((X - X_mean) * (Y - Y_mean))
denominator = np.sqrt(np.sum((X - X_mean)**2)) * np.sqrt(np.sum((Y - Y_mean)**2))
r_xy = numerator / denominator
cov = np.sum((X - X_mean) * (Y - Y_mean)) / (len(X) - 1)

X_matrix = np.column_stack((np.ones(len(X)), X))
# koefisien beta menggunakan least squares regression
beta = np.linalg.inv(X_matrix.T @ X_matrix) @ X_matrix.T @ Y

a, b = beta

Y_pred = a + b * X

SST = np.sum((Y - np.mean(Y)) ** 2)  # Total sum of squares
SSR = np.sum((Y - Y_pred) ** 2)  # Residual sum of squares
R_squared = 1 - (SSR / SST)
```

Gambar 1. *Code* untuk proses regresi linear



Gambar 2. Hasil Regresi Linar pada Dataset



Gambar 3. Hasil Perhitungan Regresi Linear

Berdasarkan hasil perhitungan, diperoleh nilai *intercept (a)* sebesar -16.26, yang menunjukkan bahwa ketika $X = 0$, nilai prediksi Y adalah sekitar -16.26. Kemudian, nilia s*lope (b)* sebesar -0.03, yang berarti setiap peningkatan 1unit dalam X hanya menyebabkan perubahan sebesar -0.03 pada Y. Nilai *covariance* dan *Pearson correlation coefficient* menunjukkan hubungan yang lemah antara kedua variabel. Selain itu, diperoleh nilai R^2 yang sangat kecil, yaitu 0.0126, mengindikasikan bahwa hanya sekitar 1.26% variasi dalam Y yang dapat dijelaskan oleh X, sementara 98.74% variasi lainnya disebabkan oleh faktor lain yang tidak dimasukkan dalam model.

Dari visualisasi *scatter plot*, titik-titik data tersebar secara acak tanpa pola linier yang jelas. Garis regresi linear hampir horizontal, yang mengonfirmasi bahwa perubahan X hampir tidak berpengaruh terhadap Y. Hal imi berarti, model ini kurang efektif dalam menjelaskan hubungan antara variabel, sehingga memerlukan pendekatan lain seperti regresi non-linear atau penggunaan variabel tambahan mungkin lebih sesuai untuk menganalisis dataset ini. Secara keseluruhan, regresi linear pada data ini menunjukkan bahwa hubungan antara X dan Y sangat lemah, dan model yang lebih kompleks mungkin diperlukan untuk menangkap pola yang lebih signifikan.

TASK 2

Pada task 2, diberikan dataset yang berisi sejumlah 200 angka *random* dengan variabel X dan Y yang diperoleh dari web program *octave.online.* Dengan dataset tersebut, diminta untuk menerapkan *polynomial fitting* dengan program apapun, termasuk pemrograman python. *Polynomial fittin*g dilakukan dengan orde berapapun dengan meminimalisir *underfitted* dan *overfitted.*

Pada pengolahan *polynomial fittin*g kali ini, digunakan pendekatan *Least Squares* untuk menemukan model polinomial orde 2 yang paling sesuai dengan data. Model yang digunakan pada *polynomial fitting* tersebut berbentuk persamaan polinomial orde 2 $y = ax^2 + bx +$c, dengan nilai a, b, dan c diperoleh melalui perhitungan matriks Vandermonde. Pada polynomial

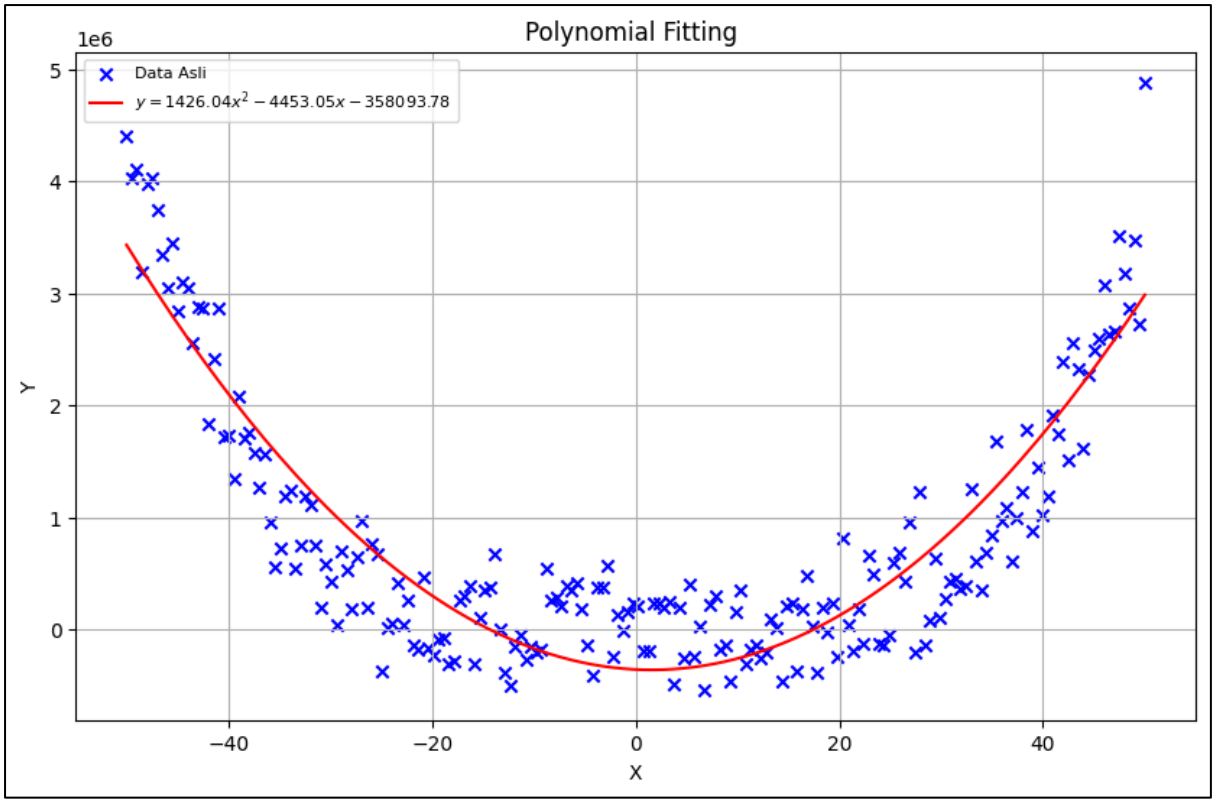
orde 2, matriks ini dibentuk dengan tiga kolom utama, yaitu dan $x^2, x,$ dan konstanta 1 yang kemudian digunakan untuk menyusun sistem persamaan linear. Kemudian, dengan menerapkan metode Least Squares, diperoleh nilai penyelesaian melalui persamaan $\beta = (X^T X)^{-1} X^T Y$, yang menghitung koefisien berdasarkan minimisasi error kuadrat antara data asli dan hasil prediksi. *Code* yang digunakan dan hasil yang diperoleh terdapat pada gambar berikut.

```
# Matriks Vandermonde untuk polinomial orde 2
X = np.vstack([x**2, x, np.ones_like(x)]).T

# Hitung koefisien polinomial dengan Least Squares
beta = np.linalg.inv(X.T @ X) @ X.T @ y
a, b, c = beta
print("a: ", a)
print("b: ", b)
print("c: ", c)

# Prediksi nilai Y dengan polinomial orde 2
y_pred = a * x**2 + b * x + c

SST = np.sum((y - np.mean(y)) ** 2)  # Total sum of squares
SSR = np.sum((y - y_pred) ** 2)  # Residual sum of squares
R_squared = 1 - (SSR / SST)
RMSE = np.sqrt(SSR / len(x))

print("SST (Total Sum of Squares): ", SST)
print("SSR (Residual Sum of Squares): ", SSR)
print("RMSE (Root Mean Squared Error): ", RMSE)
print("R^2 (Coefficient of Determination): ", R_squared)
```

Gambar 4. *Code* untuk *Polynomial Fitting* Orde 2

```
a:  1426.0443593450855
b:  -4453.051804999832
c:  -358093.78266570135
SST (Total Sum of Squares):  281043027268391.94
SSR (Residual Sum of Squares):  47201672096499.805
RMSE (Root Mean Squared Error):  485806.9168738739
R^2 (Coefficient of Determination):  0.832048236331361
```

Gambar 5. Hasil Perhitungan Komponen *Polynomial Fitting* Orde 2



Gambae 6. Hasil *Polynomial Fitting* Orde 2

Dari hasil komputasi, diperoleh nilai $a = 1426.04, b = -4453.05,$ $dan\ c = -358093.78$, yang kemudian dapat digunakan untuk membuat fungsi prediktif dari model polynomial fitting ini. Besar nilai $a = 1426.06$ yang positif menunjukkan bahwa kurva berbentuk parabola terbuka ke atas, yang berarti hubungan antara X dan Y memiliki tren yang menurun sebelum mencapai titik minimum dan kemudian meningkat kembali. Kemudian, diperoleh besar koefisien $b = -4453.05$ dan $c = -358093.78$, sehingga pada saat $X = 0$, nilai prediksi Y akan berada di sekitar -358093.78.

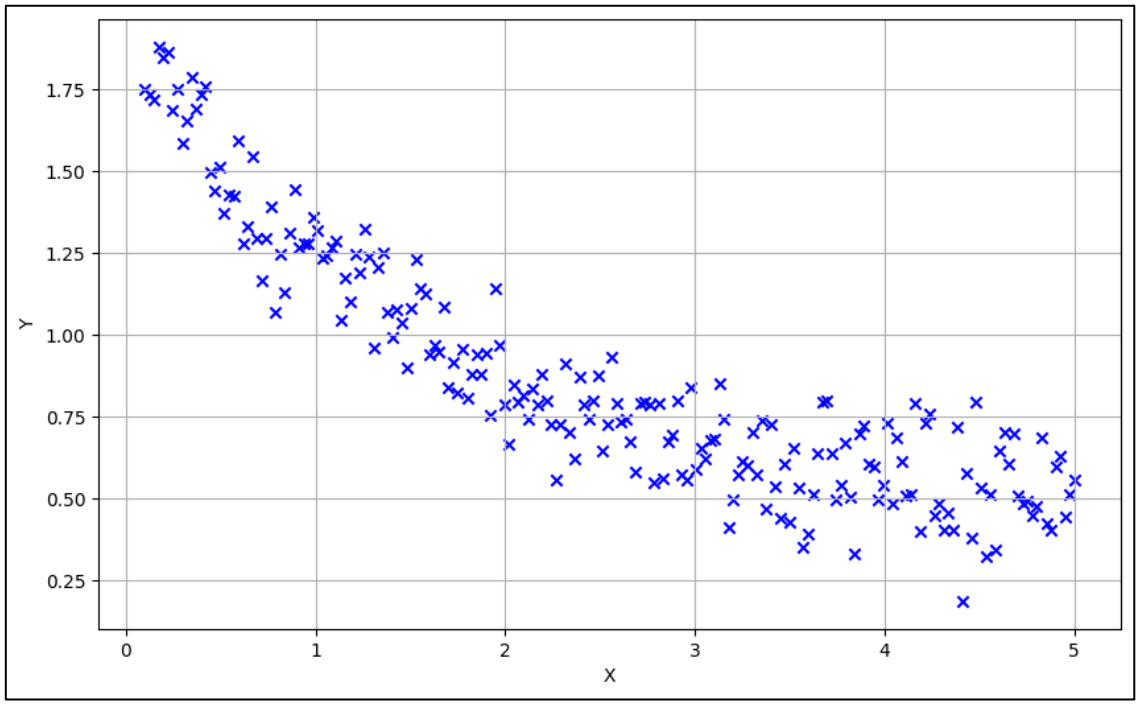
Setelah model *polynomial fitting* diterapkan, dilakukan perhitungan koefisien determinasi R^2. Perhitungan ini bertujuan untuk mengukur seberapa baik model dapat menjelaskan variasi dalam data. Dalam hal ini, Total *Sum of Squares* (SST) dihitung untuk menentukan total variabilitas dalam data asli, sementara *Residual Sum of Squares* (SSR) digunakan sebagai parameter error antara nilai prediksi dan data asli. Pengukuran besar kesalahan prediksi atau eror juga dapat dilakukan dengan menggunakan RMSE atau *Root Mean Square Error* yang diperoleh dari akar SSR. Nilai koefisien determinasi dapat diperoleh dengan persamaan berikut $R^2 = 1 - \frac{SSR}{SST}$, di mana nilai mendekati 1 menunjukkan bahwa model dapat menjelaskan sebagian besar variabilitas data dengan baik. Dari hasil perhitungan, diperoleh nilai RMSE atau *Root Mean Square Error* sebesar 485806.9168738739. Sedangkan nilai R^2

cukup tinggi, yaitu 0.832048236331361. Dari hasil tersebut, menunjukkan bahwa model *polynomial fitting* ini memiliki kecocokan yang baik dengan data yang diberikan.

Visualisasi hasil fitting menunjukkan dua elemen utama, yaitu scatter plot data asli (berwarna biru) dan kurva hasil fitting (berwarna merah). Titik-titik biru pada grafik mewakili data asli yang tersebar, dengan pola umum menyerupai bentuk parabola. Kurva merah yang diperoleh dari p*olynomial fitting* berhasil menangkap tren utama dalam data, menunjukkan bahwa pendekatan polinomial orde 2 cukup efektif untuk merepresentasikan hubungan antara X dan Y. Secara keseluruhan, polynomial fitting dengan *Least Squares* dalam analisis ini memberikan model yang cukup akurat dalam merepresentasikan hubungan antara X dan Y.

TASK 3

Pada task 3, diberikan dataset yang berisi sejumlah 200 angka *random* dengan variabel X dan Y yang diperoleh dari web program *octave.online.* Dengan dataset tersebut, diminta untuk menerapkan *function fitting* dengan program apapun, termasuk pemrograman python. *Fitting* ini dilakukan dengan fungsi dan parameter apapun dengan meminimalisir *underfitted* dan *overfitted.* Data mentah disajikan dengan *scatter plot* pada gambar 7 di bawah ini. Dari data mentah tersebut, dilakukan analisis bentuk fungsi yang paling mendekati dengan plot tersebut. Hasil analisis dan *trial error* menunjukkan fungsi yang paling mendekati yang dapat digunakan untuk *fitting* data tersebut adalah fungsi *exponential decay* dengan bentuk persamaan umum, yaitu $y = ae^{-bX}$.



Gambar 7. Data Mentah Task 3

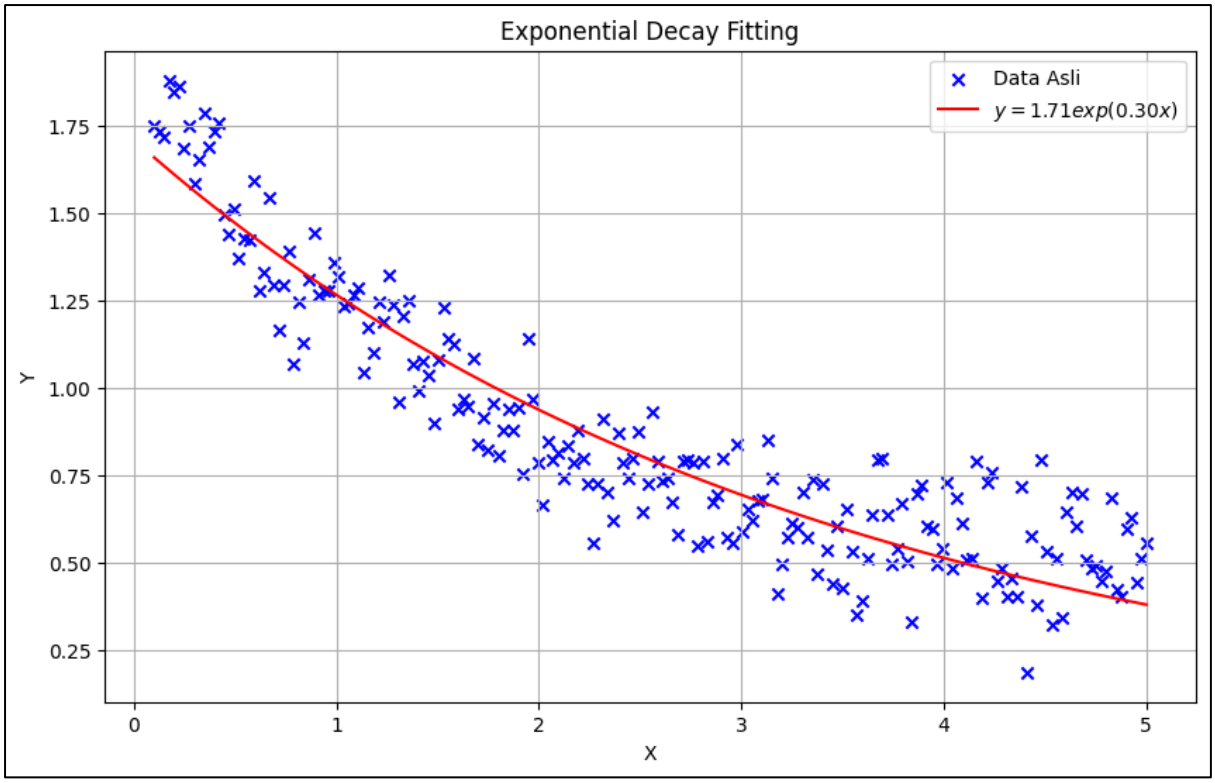
Proses fitting data eksponensial dalam analisis ini dilakukan melalui beberapa tahapan utama. Pertama, model eksponensial yang digunakan didefinisikan dengan persamaan $y = ae^{-bX}$. Nilai parameter *a* dan *b* pada fungsi ini dicari dengan *trial,* di mana berbagai kombinasi nilai diuji untuk melihat sejauh mana kurva hasil model mendekati data asli. Dengan melakukan iterasi terhadap parameter ini, didapatkan nilai yang dianggap paling sesuai. Setelah ditemukan parameter yang sesuai, kemudian prediksi nilai y dihitung berdasarkan dataset x yang tersedia. Terakhir, dilakukan *fitting* yang divisualisasikan dalam bentuk grafik. Data asli diplot menggunakan titik-titik biru, sementara hasil prediksi model diplot sebagai garis merah. *Code* dan hasil *fitting* dapat diamati pada gambar berikut.

```
# Fungsi eksponensial decay: y = a * exp(-b * x)
def decay_func(x, a, b):
    return a * np.exp(-b * x)

# trial-error a dan b
a_trial = 1.71
b_trial = 0.3
print(f"a: {a_trial:.2f}")
print(f"b: {b_trial:.2f}")

# Buat prediksi
x_fit = np.linspace(min(x_data), max(x_data), 100)
y_fit = decay_func(x_fit, a_trial, b_trial)

y_pred = decay_func(x_data, a_trial, b_trial)
SSR = np.sum((y_data - y_pred) ** 2)
SST = np.sum((y_data - np.mean(y_data)) ** 2)
RMSE = np.sqrt(SSR / len(x_data))
print(f"SSR: {SSR:.4f}")
print(f"SST: {SST:.4f}")
print(f"R^2: {1 - SSR / SST:.4f}")
print(f"RMSE: {RMSE:.4f}")
```

Gambar 8. *Code* untuk *Fitting* dengan *Exponential Decay*

```
a: 1.71
b: 0.30
SSR: 3.6535
SST: 28.9631
R^2: 0.8739
RMSE: 0.1352
```

Gambar 8. Hasil Perhitungan *Exponential Decay*



Gambar 10. Hasil *Fitting* dengan *Exponential Decay*

Hasil *fitting* dengan *exponential decay* dapat diamati pada gambar di atas. Pada persamaan tersebut, digunakan parameter $a = 1.71$ dan $b = 0.30$ yang dipilih berdasarkan RMSE yang paling kecil. Pada task ini juga dilakukan perhitungan koefisien determinasi R^2 untuk mengukur seberapa baik model dapat menjelaskan variasi dalam data. Dalam hal ini, Total *Sum of Squares* (SST) dihitung untuk menentukan total variabilitas dalam data asli, sementara *Residual Sum of Squares* (SSR) digunakan sebagai parameter error antara nilai prediksi dan data asli. Pengukuran besar kesalahan prediksi atau eror juga dapat dilakukan dengan menggunakan RMSE atau *Root Mean Square Error* yang diperoleh dari akar SSR. Hasil perhitungan dapat diamati pada gambar 9 di atas.

Besarnya nilai koefisien determinasi (R^2), yaitu 0.873 menunjukkan bahwa sekitar 87.39% variasi dalam data dapat dijelaskan oleh model eksponensial ini, meskipun masih menyisakan sekitar 12.61% variasi yang tidak terjelaskan, yang mungkin disebabkan oleh noise atau faktor lain yang tidak terakomodasi dalam model. Kemudian, dengan nilai RMSE yang diperoleh, yaitu 0.1352 menunjukkan bahwa model cukup akurat dengan rata-rata kesalahan yang relatif kecil dibandingkan skala data.

Berdasarkan hasil fitting dan analisis parameter, dapat disimpulkan bahwa model eksponensial yang digunakan mampu menangkap tren utama dari data asli. Nilai $a = 1.71$ dan $b = 0.30$ yang dipilih dapat *fitted* dan sesuai dengan model dataset secara umum meskipun

masih terdapat penyimpangan. Dari *trial and error* berbagai fungsi dan parameter *a* dan *b,* hasil paling baik diperoleh dengan fungsi *exponential decay.*