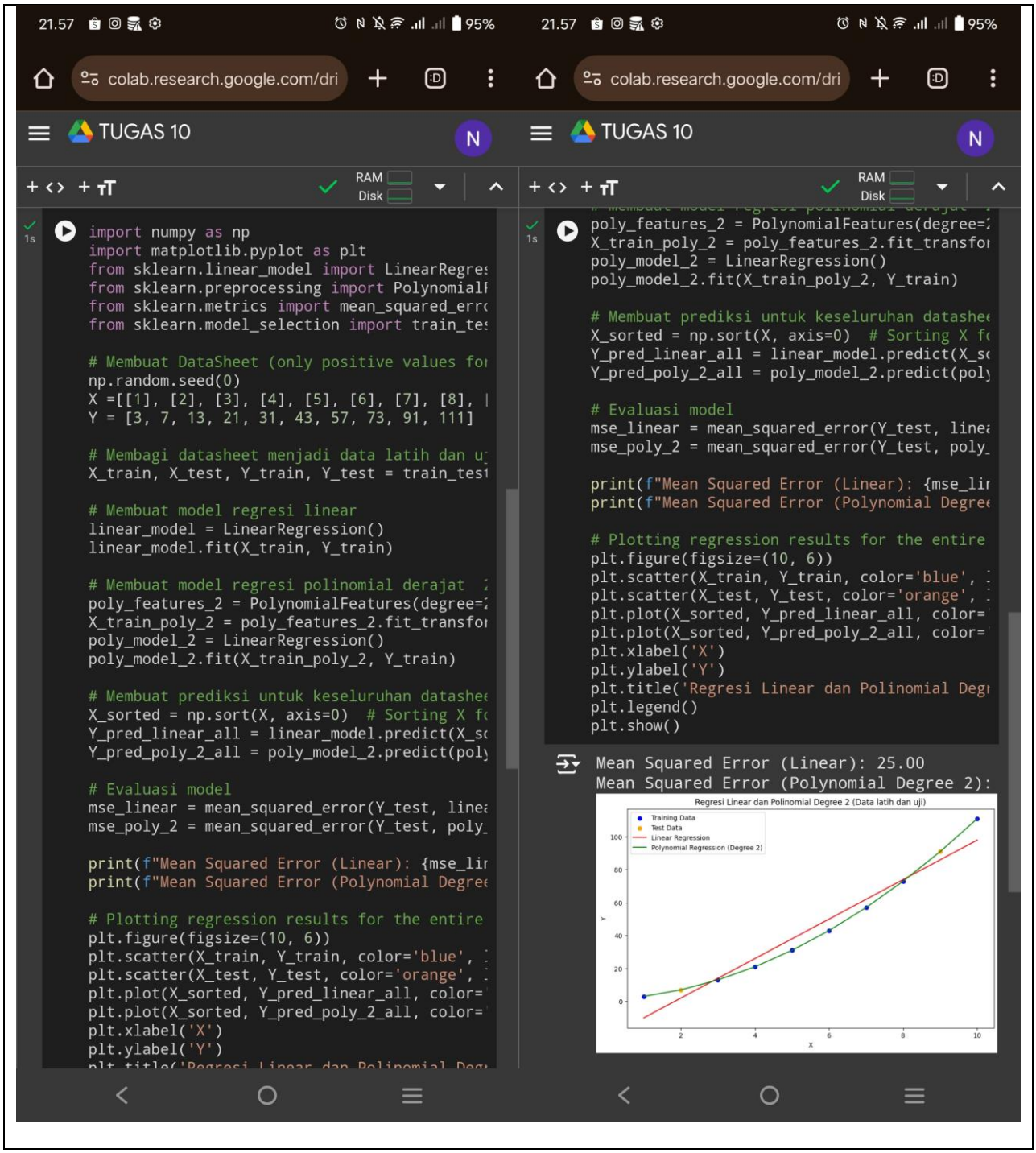
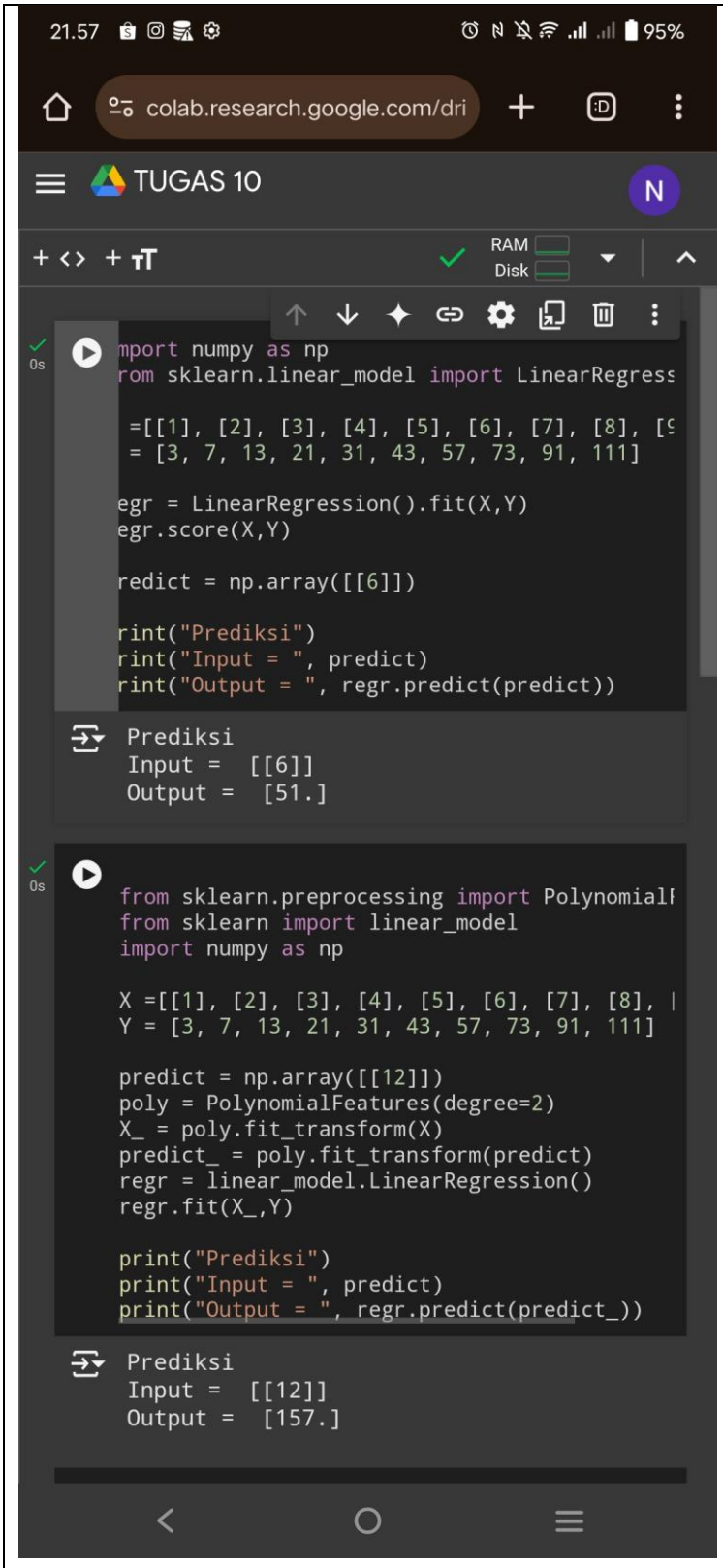
Nama : Novianti Zakiah

Nim : 1227030026

1-2



4


21.57
95%
colab.research.google.com/dri
TUGAS 10
N
RAM
Disk
0s
mport numpy as np
rom sklearn.linear_model import LinearRegress
=[[1], [2], [3], [4], [5], [6], [7], [8], [9
= [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]
egr = LinearRegression().fit(X,Y)
egr.score(X,Y)
redict = np.array([[6]])
rint("Prediksi")
rint("Input = ", predict)
rint("Output = ", regr.predict(predict))
Prediksi
Input = [[6]]
Output = [51.]
0s
from sklearn.preprocessing import PolynomialF
from sklearn import linear_model
import numpy as np
X =[[1], [2], [3], [4], [5], [6], [7], [8],
Y = [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]
predict = np.array([[12]])
poly = PolynomialFeatures(degree=2)
X_ = poly.fit_transform(X)
predict_ = poly.fit_transform(predict)
regr = linear_model.LinearRegression()
regr.fit(X_,Y)
print("Prediksi")
print("Input = ", predict)
print("Output = ", regr.predict(predict_))
Prediksi
Input = [[12]]
Output = [157.]

3.untuk Mean Squared Error (MSE) antara regresi linear dan regresi polinomial ini lebih akurat regresi linear karena nilai input dan outputnya lebih kecil itu menunjukan lebih akurat.

5.

- pertama masukkan library yang dibutuhkan :

import numpy as np , ini untuk perhitungan numerik

import matplotlib.pyplot as plt, untuk visualisasi grafik

from sklearn.linear_model import LinearRegression , untuk membuat model regresi linear

from sklearn.preprocessing import PolynomialFeatures, : Untuk memanipulasi data menjadi bentuk polinomial

from sklearn.metrics import mean_squared_error, untuk menghitung MSE

from sklearn.model_selection import train_test_split, untuk membagi datasheet menjadi data uji dan latih

- masukkan data sheet x dan y nya :

```
np.random.seed(0)

X = [[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]]

Y = [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]
```

- masukkan kode program untuk membagi data sheet :

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

- masukkan untuk model regresi linear

```
linear_model = LinearRegression()

linear_model.fit(X_train, Y_train)
```

- Masukkan untuk model regresi polinomial:

```
poly_features_2 = PolynomialFeatures(degree=2)

X_train_poly_2 = poly_features_2.fit_transform(X_train)

poly_model_2 = LinearRegression()

poly_model_2.fit(X_train_poly_2, Y_train)
```

- masukkan kode untuk prediksi seluruh data :

```
X_sorted = np.sort(X, axis=0)  # Mengurutkan X untuk plotting

Y_pred_linear_all = linear_model.predict(X_sorted)

Y_pred_poly_2_all = poly_model_2.predict(poly_features_2.transform(X_sorted))
```

- masukkan kode untuk evaluasi nilai erornya atau Mean Squared Eror (MSE)

```
mse_linear = mean_squared_error(Y_test, linear_model.predict(X_test))

mse_poly_2 = mean_squared_error(Y_test, poly_model_2.predict(poly_features_2.transform(X_test)))
```

- masukkan kode untuk menampilkan hasil evaluasi MSE :

```
print(f"Mean Squared Error (Linear): {mse_linear:.2f}")

print(f"Mean Squared Error (Polynomial Degree 2): {mse_poly_2:.2f}")
```

- masukkan kode untuk visualisasi grafik

```
plt.figure(figsize=(10, 6))

plt.scatter(X_train, Y_train, color='blue', label='Training Data')  # Data latih

plt.scatter(X_test, Y_test, color='orange', label='Test Data')  # Data uji

plt.plot(X_sorted, Y_pred_linear_all, color='red', label='Linear Regression')  # Regresi linear

plt.plot(X_sorted, Y_pred_poly_2_all, color='green', label='Polynomial Regression (Degree 2)')  # Regresi polinomial

plt.xlabel('X')

plt.ylabel('Y')

plt.title('Regresi Linear dan Polinomial Degree 2 (Data latih dan uji)')

plt.legend()

plt.show()
```