

**ITERA**

**Modul Praktikum 1**  
**Pembelajaran Mesin**

**Regresi Linear, Regresi Polinomial, Berganda, Logistik**

**Program Studi Sains Data**  
**Fakultas Sains**  
**Institut Teknologi Sumatera**

**2024**

## I. Tujuan Instruksional Umum

Praktikum ini bertujuan memberikan pemahaman dasar tentang regresi dalam konteks pembelajaran mesin.

## II. Tujuan Instruksional Khusus

1. Mahasiswa mampu menerapkan model regresi linier, polinomial, berganda, dan logistik menggunakan library scikit-learn.
  2. Mahasiswa mampu memahami perbedaan dan cara kerja masing-masing jenis regresi.
  3. Mahasiswa mampu mengevaluasi performa model regresi menggunakan metrik yang relevan.
- 

## A. Teori Dasar Regresi

Regresi adalah teknik statistik yang digunakan untuk memodelkan dan menganalisis hubungan antara variabel dependen (target) dan satu atau lebih variabel independen (fitur). Dalam konteks pembelajaran mesin, regresi digunakan untuk memprediksi nilai kontinu/nilai numerik berdasarkan input data/data historis.

Dalam praktikum ini, kita akan mempelajari dan mengimplementasikan 4 jenis regresi menggunakan library scikit-learn sebagai berikut:

### 1. Regresi Linear

Regresi Linear adalah metode dasar yang memodelkan hubungan antara variabel dependen ( $y$ ) dan satu variabel independen ( $X$ ) dengan persamaan garis lurus:

$$y = \beta_0 + \beta_1 X + \varepsilon \quad (1)$$

Di mana:

$y$  adalah variabel dependen (target).

$X$  adalah variabel independen (fitur).

$\beta_0$  adalah intersep (nilai  $y$  ketika  $X = 0$ ).

$\beta_1$  adalah koefisien kemiringan (slope).

$\varepsilon$  adalah error term.

Tujuan dari regresi linear adalah menemukan nilai optimal untuk  $\beta_0$  dan  $\beta_1$  yang meminimalkan kesalahan prediksi (biasanya menggunakan metode *Least Squares*).

Contoh Penggunaan:

- Prediksi harga rumah berdasarkan luas tanah.
- Prediksi konsumsi energi berdasarkan suhu.

## 2. Regresi Polinomial

Regresi Polinomial adalah perluasan dari regresi linear yang memungkinkan model untuk menangkap hubungan non-linear antara variabel dependen dan independen dengan memperkenalkan derajat polinomial (n):

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_n X^n + \varepsilon \quad (2)$$

Di mana n adalah derajat polinomial. Semakin tinggi nilai n, semakin kompleks model yang dihasilkan. Namun, ini juga meningkatkan risiko overfitting, di mana model terlalu sesuai dengan data latih dan tidak dapat digeneralisasi dengan baik ke data baru.

Contoh Penggunaan:

- Prediksi pertumbuhan populasi.
- Analisis kurva laju pertumbuhan bakteri.

## 3. Regresi Berganda

Regresi Berganda adalah bentuk regresi linear di mana terdapat lebih dari satu variabel independen. Model ini memodelkan hubungan antara variabel dependen (y) dan beberapa variabel independen ( $X_1, X_2, \dots, X_n$ ) dengan persamaan:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon \quad (3)$$

Regresi berganda digunakan ketika hasil dipengaruhi oleh beberapa faktor yang ingin diukur pengaruhnya secara simultan.

Contoh Penggunaan:

- Prediksi harga rumah berdasarkan beberapa fitur seperti luas tanah, jumlah kamar, dan lokasi.
- Prediksi kinerja karyawan berdasarkan pendidikan, pengalaman, dan usia.

## 4. Regresi Logistik

Regresi Logistik digunakan untuk memodelkan probabilitas kejadian dari suatu peristiwa biner (dua kelas). Berbeda dengan regresi linear yang memprediksi nilai kontinu, regresi logistik memprediksi nilai yang dibatasi antara 0 dan 1:

$$P(y=1|X) = 1 / (1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}) \quad (4)$$

Nilai yang dihasilkan diinterpretasikan sebagai probabilitas dari kejadian kelas positif (misalnya, kelas 1).

Contoh Penggunaan:

- Prediksi apakah seorang pasien memiliki penyakit tertentu (sembuh atau tidak).
- Klasifikasi email sebagai spam atau non-spam.

## Evaluasi Model

Setiap model regresi memerlukan evaluasi untuk menilai kinerja dan kemampuannya dalam membuat prediksi. Beberapa metrik evaluasi yang umum digunakan adalah:

- ✓ Mean Squared Error (MSE): Digunakan dalam regresi linear, polinomial, dan berganda untuk mengukur rata-rata kesalahan kuadrat antara nilai prediksi dan nilai aktual.
- ✓  $R^2$  (Koefisien Determinasi): Menunjukkan seberapa baik model menjelaskan variabilitas data.
- ✓ Akurasi: Dalam regresi logistik, digunakan untuk mengukur persentase prediksi benar dari total prediksi.

## B. Latihan Praktikum

Latihan praktikum ini bertujuan mahasiswa diharapkan dapat memodifikasi dan mengembangkan latihan ini untuk pemahaman yang lebih mendalam. Adapun langkah-langkah dalam praktikum modul 1 sebagai berikut.

### 1. Persiapan

Dalam modul praktikum ini menggunakan bahasa pemrograman python dengan menggunakan google colab yang dapat diakses di web: <https://colab.research.google.com/>.

#### Instalasi:

Instalasi pustaka yang diperlukan.

```
pip install numpy pandas matplotlib scikit-learn
```

#### Import Library:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, accuracy_score
```

### 2. Regresi Linear

Tujuan: Memahami regresi linear sebagai model dasar pembelajaran mesin.

Langkah-langkah:

a. Buat dataset:

```
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9]).reshape(-1, 1)
y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10])
```

b. Bagi data menjadi data latih dan data uji:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

c. Latih model regresi linear:

```
model = LinearRegression()  
model.fit(X_train, y_train)
```

d. Prediksi dan evaluasi model:

```
y_pred = model.predict(X_test)  
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))  
print("R2 Score:", r2_score(y_test, y_pred))
```

Hasil ouputnya :

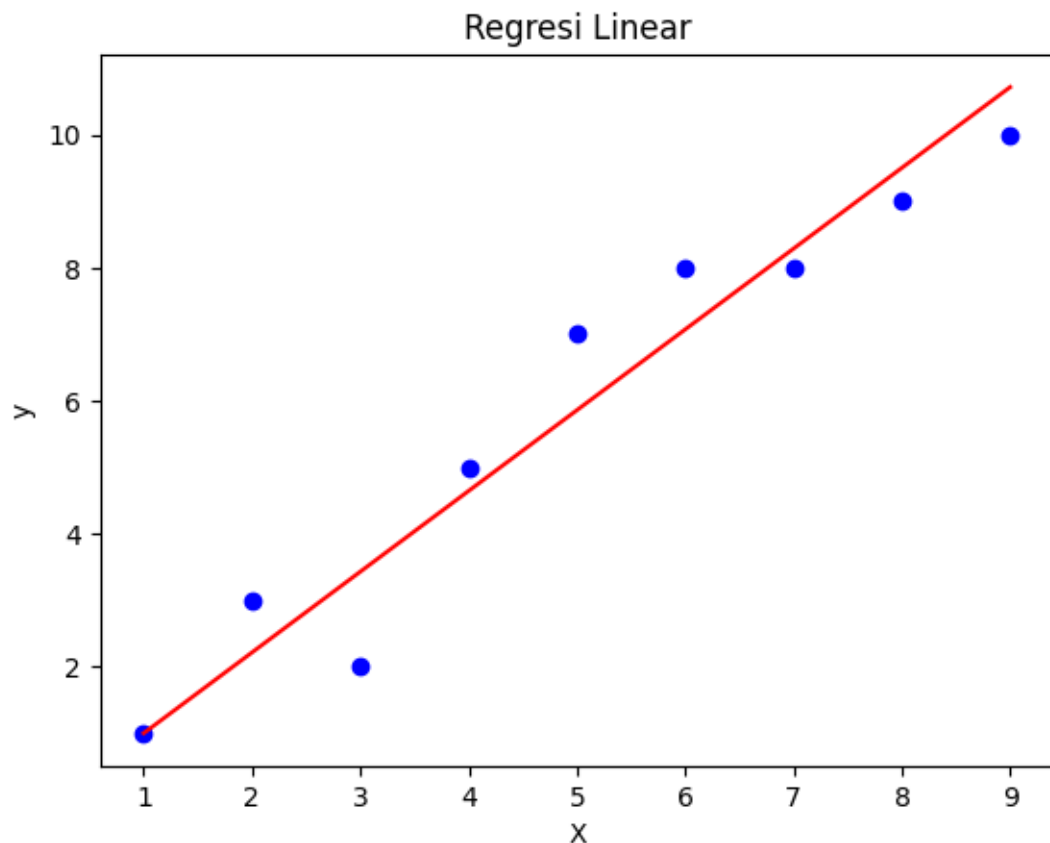
Mean Squared Error: 0.4336734693877543

R<sup>2</sup> Score: 0.9518140589569162

e. Visualisasi hasil:

```
plt.scatter(X, y, color='blue')  
plt.plot(X, model.predict(X), color='red')  
plt.title('Regresi Linear')  
plt.xlabel('X')  
plt.ylabel('y')  
plt.show()
```

Hasil ouputnya :



### 3. Regresi Polinomial

Tujuan: Memahami cara kerja regresi polinomial untuk menangani hubungan non-linear.

Langkah-langkah:

a. Transformasi fitur menjadi polinomial:

```
poly = PolynomialFeatures(degree=3)  
X_poly = poly.fit_transform(X)
```

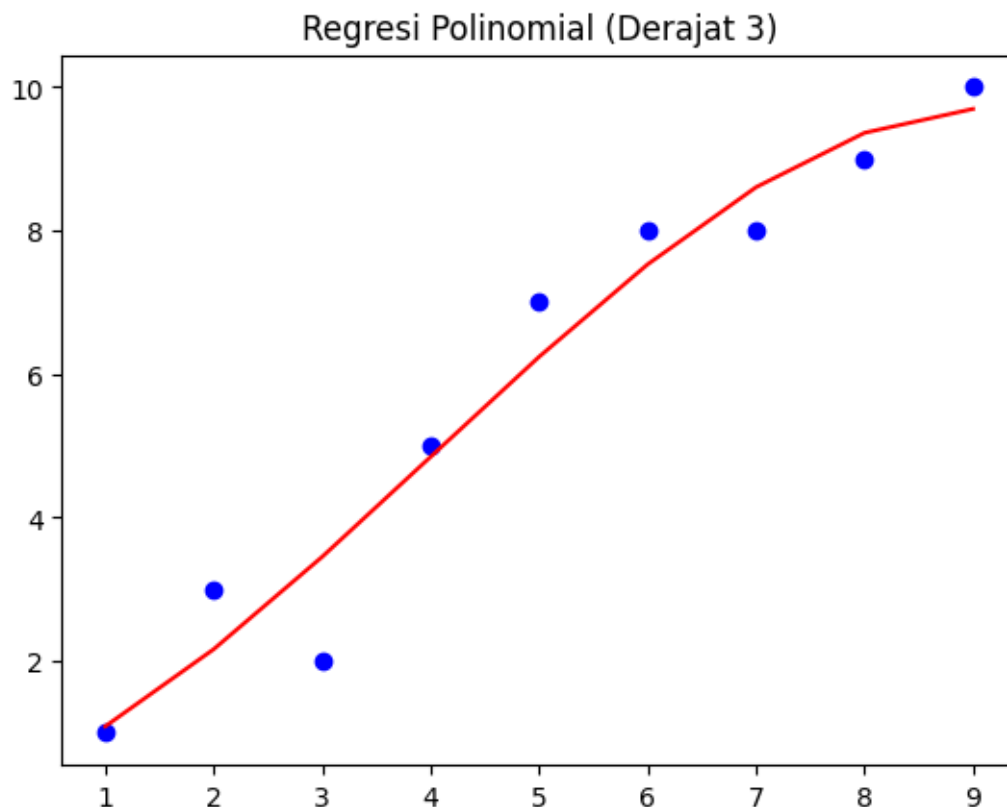
b. Latih model regresi polinomial:

```
model_poly = LinearRegression()  
model_poly.fit(X_poly, y)
```

c. Prediksi dan visualisasi:

```
plt.scatter(X, y, color='blue')  
plt.plot(X, model_poly.predict(poly.fit_transform(X)), color='red')  
plt.title('Regresi Polinomial (Derajat 3)')  
plt.show()
```

Hasil outputnya :



#### 4. Regresi Berganda

Tujuan: Mempelajari regresi dengan beberapa variabel independen (*multivariate regression*).

Langkah-langkah:

a. Buat dataset dengan lebih dari satu fitur:

```
X = np.array([[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10]])  
y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10])
```

b. Bagi data dan latih model:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
model = LinearRegression()  
model.fit(X_train, y_train)
```

c. Evaluasi model:

```
y_pred = model.predict(X_test)  
mse = mean_squared_error(y_test, y_pred)  
print(f'MSE: {mse}')
```

Hasil ouputnya :  
MSE: 0.1533203125

## 5. Regresi Logistik

Tujuan: Memahami regresi logistik sebagai model klasifikasi biner.

Langkah-langkah:

a. Buat dataset biner:

```
X = np.array([[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10]])  
y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1])
```

b. Bagi data dan latih model:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
model = LogisticRegression()  
model.fit(X_train, y_train)
```

c. Prediksi dan evaluasi:

```
y_pred = model.predict(X_test)  
accuracy = accuracy_score(y_test, y_pred)  
print(f'Akurasi: {accuracy}')
```

Hasil ouputnya :  
Akurasi: 1.0



## Referensi

- Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.
- Raschka, S., & Mirjalili, V. (2019). Python Machine Learning. Packt Publishing.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.
- Scikit-learn documentation: <https://scikit-learn.org/>
- Towards Data Science: <https://towardsdatascience.com/>