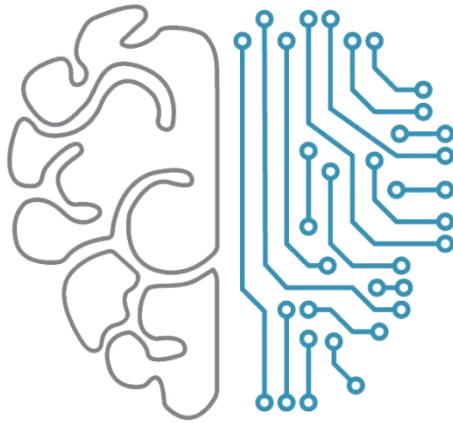


LAPORAN TEORI

PEMBELAJARAN MESIN



INTELLIGENT **COMPUTING**
LABORATORY

NAMA : PARISYA BOKINGS
NIM : 202131119
KELAS : B
DOSEN : Dr. Dra. DWINA KUSWARDANI, M.Kom.
NO.PC : 10
ASISTEN : 1. ARIFINA RESYUANTI
2. HAYA Q. LUTHFIYANINGSIH
3. VIANA SALSABILA FAIRUZ SYAHLA
4. MUHAMMAD HANIEF FEBRIANSYAH

INSTITUT TEKNOLOGI PLN
TEKNIK INFORMATIKA

2023

Mechine Learning Tutorial part-1
Beberapa jenis Mechine Learning Algorithms:
- Linear Regression
- Decision Tree
- Support Vector Mechine
Dalam video tersebut akan melakukan kasus penggunaan dimana kita akan mengklasifikasikan apakah resep mereka berupa cupcake atau muffin menggunakan SVM atau Support Vector Mechine.
Mengapa kita harus peduli akan Mechine Learning di masa yang akan mendatang?
dalam video menjelaskan bahwa dimasa yang akan datang mechine dapat melakukan semua hal baru ini untuk kami dengan baik. Contohnya, karena mechine seorang dapat mengemudikan mobil untuk melihat arah jalan dengan menggunakan Google dan kemudian kita bisa melakukan pemesanan kendaraan online melalui aplikasi
What is Mechine Learning?
Mechine Learning adalah ilmu yang membuat computer belajar dan bertindak seperti manusia dengan memasukkan data dan informasi tanpa deprogram secara eksplisit. Pada saat ini kita dapat menjalankan banyak hal di ponsel karena ponsel akan sangat terikat dan kemudian dengan adanya kecerdasan buatan dan Mechine Learning sekarang banyak yang mengambil data dan belajar untuk apa yang terjadi sebelumnya dan kemudian memprediksi apa yang akan terjadi selanjutnya .
Bagian saat ini yang terbesar dalam mechine learning adalah bagaimana kita menemukan solusi baru sehingga kita beralih dari kata-kata deskriptif , belajar tentang berbagai hal dan pemahamn bagaimana hal ini cocok untuk memprediksi apa yang akan dilakukan untuk memposting solusi baru .
Untuk membersihkan semua hal yang masuk dan melatih algoritma, dalam video ini memilih untuk bekerja sama dengan Support Vector Mechine karena akan melakukan uji model, apakah uji model ini berfungsi dan valid dengan apa yang kita lakukan, kemudian setelah mengujinya akan dijalankan prediksi, ingin menjalankan prediksi antara pemilihan kita atau keluaran papun yang akan dihasilkan. Setelah semuanya sudah diatur dan telah selesai melakukan banyak pengujian, maka dilanjutkan menerapkan modelnya, ini khusus domain.
Pertanyaan selanjutnya muncul, Apa yang kami sebut data tidak terlatih?
Apakah kita ingin menemukan struktur dalam data yang belum dijelajahi dan itu disebut pengelompokan, misalnya menemukan sekelompok pelanggan dengan perilaku serupa dengan database besar data dengan berisi demografi dan catatan pembelian masa lalu dan dalam hal ini

kita mungkin memperhatikan bahwa siapapun yang memakai sepatu tertentu akan pergi berbelanja di toko tertentu, mereka akan melakukan pembelian tertentu dengan memiliki informasi tersebut, hal ini membantu kita memasarkan atau mengelompokkan orang-orang apakah mereka berdasarkan bidang, investasi, dan latar belakang keuangan yang berbeda, apakah kita bisa memberi pinjaman atau tidak, bahkan kita dapat melihat apakah mereka pelanggan yang sah atau tidak.
Supervised Learning
Supervised Learning atau pembelajaran yang diawasi adalah metode yang digunakan untuk memungkinkan machine mengklasifikasikan masalah atau situasi objek yang diprediksi berdasarkan pada data label yang diumpankan ke machine, di video itu pun memiliki beberapa data yang sudah diberikan label pada masing-masing data. Sangat penting jika kita sedang Melakukan Supervised Learning kita sudah mengetahui jawaban atas banyak informasi yang masuk sehingga kita memiliki sekelompok besar data baru yang masuk. Jadi kami telah melatih model tersebut, model tersebut sekarang mengetahui perbedaan antara sebuah data, dan setelah kita melatihnya kita dapat mengirimkannya dalam kasus tertentu sesuai datanya.
Reinforcement Learning
Jenis machine learning yang penting atau agent yang harus berperilaku dalam suatu lingkungan dengan melakukan tindakan dan melihat hasil yang kita dapat. Dimana dalam video tersebut menggunakan bayi, sangat bagus jika menggunakan bayi menurut dari dalam video. Machine Learning terbesar diluar sana saat ini atau di masa depan dalam Reinforcement Learning bagaimana membuatnya berhasil bagi mereka.
Linear Regression
Linear Regression mungkin adalah salah satu yang paling terkenal dan algoritma yang dipahami dalam statistic dan machine learning.
Linear Regression adalah model linear misalnya model yang mengasumsikan hubungan linear antara variable masukan X dan variable keluaran Y, dan kita akan melihatnya jika kita mengingat dari kelas aljabar, $y = mx + c$ (bayangkan kita memperkirakan jarak yang ditempu Y dari kecepatan X, representasi model linear regression untuk soal tersebut adalah $y = m \cdot x + c$ 'jarak' = $m \cdot \text{kecepatan} + c$, Dimana m adalah koefisien).
Implementasi matematis linear regression akan mengambil data, jadi misalnya kita punya kumpulan data dimana memiliki kolom x dan y, kolom x = 1 2 3 4 5, kolom y = 3 2 2 4 3.
Ketika kita mengambilnya dan kita melanjutkan untuk memplot titik pada grafik kita dapat melihat garis ditengah-tengahnya, tetapi kita akan menghitung garis yang tepat untuk linear regression dan hal pertama yang kita lakukan kita memiliki mean(xi), dan ingat mean pada dasarnya adalah rata-rata, jadi kita menjumlahka semua yang ada di kolom x dan membaginya dengan 5 sehingga hasilnya menjadi tiga 'mean(xi)=3' dan lakukan hal yang sama pada kolom y dan menghasilkan 'mean(xi)=2.8', lalu hasil keduanya dimasukkan kedalam grafik dan mengurutkannya dengan garis putus-putus.

<p>Mari kita cari persamaan regresi untuk menemukan garis yang paling cocok. Kita ambil $y=mx+c$, jadi kita untuk menemukan persamaan ini untuk data kita perlu mencari kemiringan M dan koefisien c, dan kita punya $y=mx+c$ dimana m adalah jumlah rata-rata 'x-x*jumlah rata-rata y.</p>

Decision Trees

Cara yang sangat berbeda untuk memecahkan masalah dalam model linear regression. Decision
Trees adalah algoritma berbentuk pohon yang digunakan untuk menentukan suatu tindakan, setiap
cabang pohon mewakili kemungkinan keputusan atau arus atau raksi.

SVM atau Support Vector Mechine

Support Vector Mechine adalah algoritma klasifikasi yang banyak digunakan, Support Vector
Mechine algoritma menciptakan garis pemisah yang membagi kelas dengan cara terbaik.

What is Machine Learning ?

Ordinary System	>	With Srtificial Intelligence	>	Mechine Learning	>	Learns, Predics, Improves
-----------------	---	------------------------------	---	------------------	---	---------------------------

Types Mechine Learning:

- | |
|-----------------|
| - Supervised |
| - Reinforcement |
| - Un-Supervised |

Supervised VS Unsupervised

Labeled Data | Non-labeled Data

Direct feedback | NO feedback

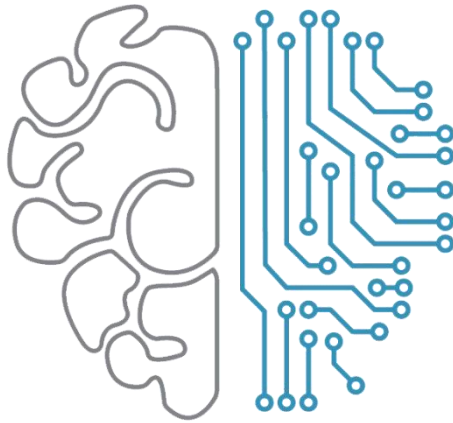
Predict output Find hidden structure in data
--

Machine Learning Algorithms:

- | | |
|---|------------------------|
| - | Linear Regression |
| - | Decision Trees |
| - | Support Vector Mechine |

LAPORAN PRAKTIKUM

PEMBELAJARAN MESIN



INTELLIGENT **COMPUTING**
LABORATORY

NAMA : PARISYA BOKINGS
NIM : 202131119
KELAS : B
DOSEN : Dr. Dra. DWINA KUSWARDANI, M.Kom.
NO.PC : 10
ASISTEN : 1. ARIFINA RESYUANTI
2. HAYA Q. LUTHFIYANINGSIH
3. VIANA SALSABILA FAIRUZ SYAHLA
4. MUHAMMAD HANIEF FEBRIANSYAH

INSTITUT TEKNOLOGI PLN
TEKNIK INFORMATIKA

2023

Praktikum Dataset Salary dan data-Penjualan

Documents/PARISYA BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... | RegresiLinear_Dataset - Jupyter | RegresiLinear_Dataframe - Jupyter | +

localhost:8889/notebooks/Documents/PARISYA%20BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... | Logout

jupyter RegresiLinear_Dataset (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

PARISYA BOKINGS 202131119

Import Library

```
In [31]: 1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
```

Dataset

```
In [12]: 1 data1 = pd.read_csv('salary_dataset.csv')
```

```
In [13]: 1 data1
```

```
Out[13]:
```

	no	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0

Documents/PARISYA BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... | RegresiLinear_Dataset - Jupyter | RegresiLinear_Dataframe - Jupyter | +

localhost:8889/notebooks/Documents/PARISYA%20BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... | Logout

jupyter RegresiLinear_Dataset (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

```
28 28 10.4 122392.0
29 29 10.6 121873.0
```

```
In [4]: 1 data1.head() #melihat data teratas
```

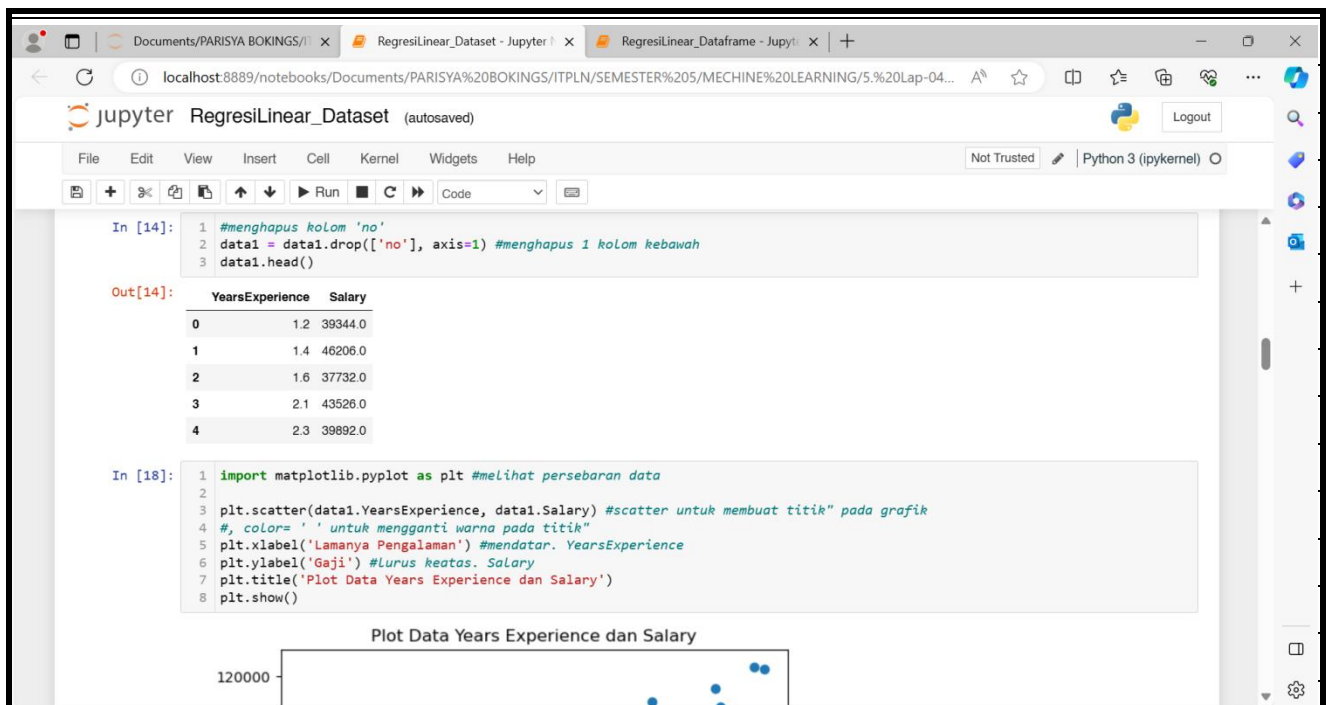
```
Out[4]:
```

	no	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0

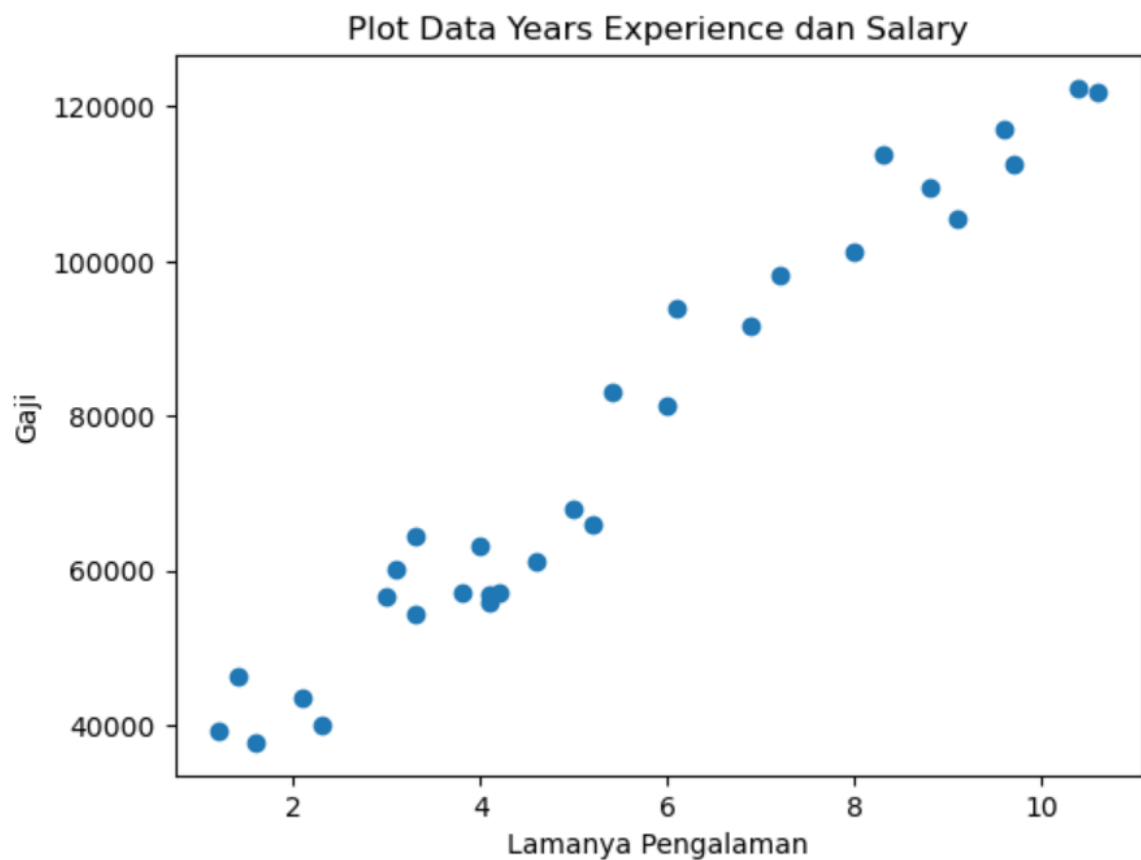
```
In [6]: 1 data1.tail() #melihat data terbawah
```

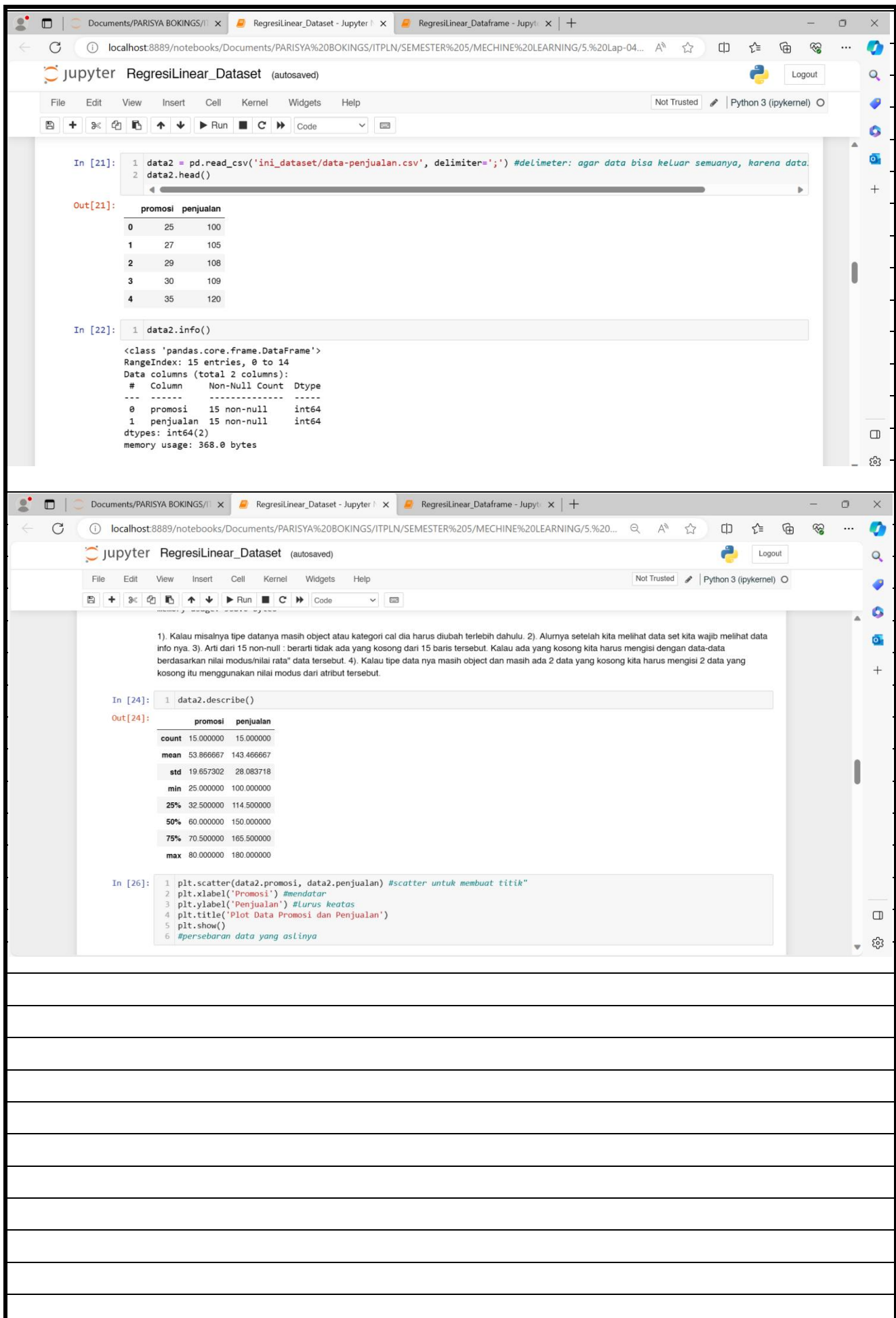
```
Out[6]:
```

	no	YearsExperience	Salary
25	25	9.1	105583.0
26	26	9.6	116970.0
27	27	9.7	112636.0
28	28	10.4	122392.0
29	29	10.6	121873.0



8 plt.show()





The screenshot displays a Jupyter Notebook interface with two visible code cells. The first cell, labeled 'In [21]:', contains the code to read a CSV file and display its first five rows. The output, 'Out[21]:', shows a table with two columns: 'promosi' and 'penjualan'. The second cell, labeled 'In [22]:', contains the code to get information about the DataFrame. The output, 'Out[22]:', shows the DataFrame's structure, including the number of entries, columns, and data types.

In [21]:

```
1 data2 = pd.read_csv('ini_dataset/data-penjualan.csv', delimiter=';') #delimiter: agar data bisa keluar semuanya, karena data.
2 data2.head()
```

Out[21]:

	promosi	penjualan
0	25	100
1	27	105
2	29	108
3	30	109
4	35	120

In [22]:

```
1 data2.info()
```

Out[22]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype  
---  -
0    promosi     15 non-null     int64  
1    penjualan   15 non-null     int64  
dtypes: int64(2)
memory usage: 368.0 bytes
```

1). Kalau misalnya tipe datanya masih object atau kategori cal dia harus diubah terlebih dahulu. 2). Alurnya setelah kita melihat data set kita wajib melihat data info nya. 3). Arti dari 15 non-null : berarti tidak ada yang kosong dari 15 baris tersebut. Kalau ada yang kosong kita harus mengisi dengan data-data berdasarkan nilai modus/nilai rata" data tersebut. 4). Kalau tipe data nya masih object dan masih ada 2 data yang kosong kita harus mengisi 2 data yang kosong itu menggunakan nilai modus dari atribut tersebut.

In [24]:

```
1 data2.describe()
```

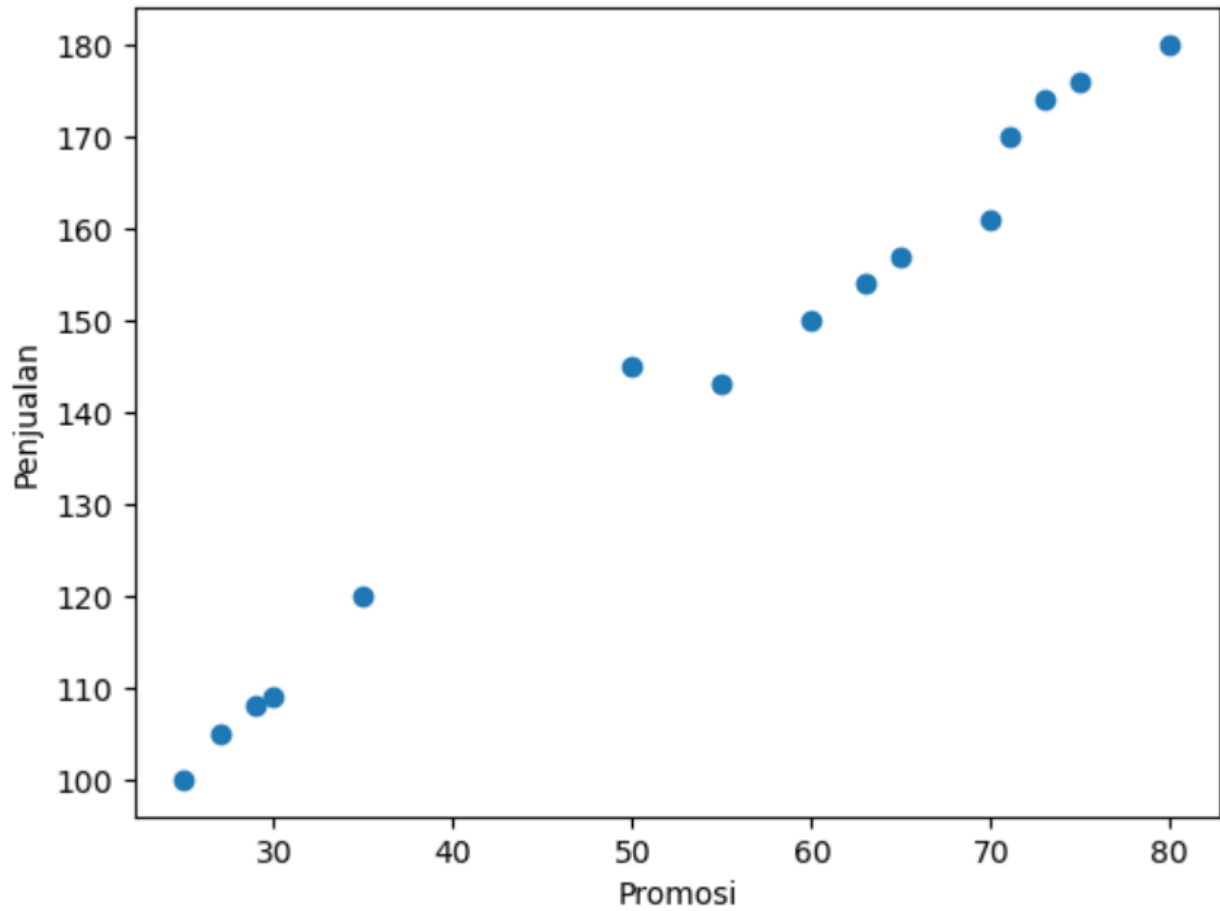
Out[24]:

	promosi	penjualan
count	15.000000	15.000000
mean	53.866667	143.466667
std	19.657302	28.083718
min	25.000000	100.000000
25%	32.500000	114.500000
50%	60.000000	150.000000
75%	70.500000	165.500000
max	80.000000	180.000000

In [26]:

```
1 plt.scatter(data2.promosi, data2.penjualan) #scatter untuk membuat titik"
2 plt.xlabel('Promosi') #mendatar
3 plt.ylabel('Penjualan') #Lurus keatas
4 plt.title('Plot Data Promosi dan Penjualan')
5 plt.show()
6 #persebaran data yang aslinya
```


Plot Data Promosi dan Penjualan



Documents/PARISYA BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... | RegresiLinear_Dataset - Jupyter | RegresiLinear_Dataframe - Jupyter

localhost:8889/notebooks/Documents/PARISYA%20BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... | Logout

jupyter RegresiLinear_Dataset (autosaved) | Not Trusted | Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Korelasi

Pentingnya korelasi untuk mencari regresi.

```
In [28]: 1 data1.corr() #melihat korelasi
```

```
Out[28]:
```

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

Kalo nilai korelasi nya tinggi dia bisa diolah menggunakan regresi, Tapi kalau dibawah 50% dia tidak bisa diolah menggunakan regresi karena tidak berhubungan antar x dan y. Sedangkan regresi didapatkan nilainya dari keterhubungan antara dua atribut.

Untuk melihat atribut mana yang akan dipakai: Cari yang korelasinya terbesar, acuan nya kepada nilai Y.

```
In [30]: 1 data2.corr()
```

```
Out[30]:
```

	promosi	penjualan
promosi	1.00000	0.99369
penjualan	0.99369	1.00000

Split Kolom

```
In [36]: 1 #data salary
2 years_exp = data1['YearsExperience'].values[:, np.newaxis] #x train nya
3 salary = data1['Salary'].values #y train nya
4
5 #data penjualan
6 promosi = data2['promosi'].values[:, np.newaxis] #x
7 penjualan = data2['penjualan'].values #y
8
9 #tidak pakai split train_test karena datanya masih sedikit, jadi semua datanya dipakai untuk training
10 # from sklearn.model_selection import train_test_split
11 # x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)
```

Modeling

```
In [38]: 1 from sklearn.linear_model import LinearRegression
2
3 model1 = LinearRegression() #menguji dataset1
4 model2 = LinearRegression() #menguji dataset2
5
6 model1.fit(years_exp, salary) #yang dipakai data trainingnya
7 model2.fit(promosi, penjualan)
```

Out[38]: LinearRegression()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

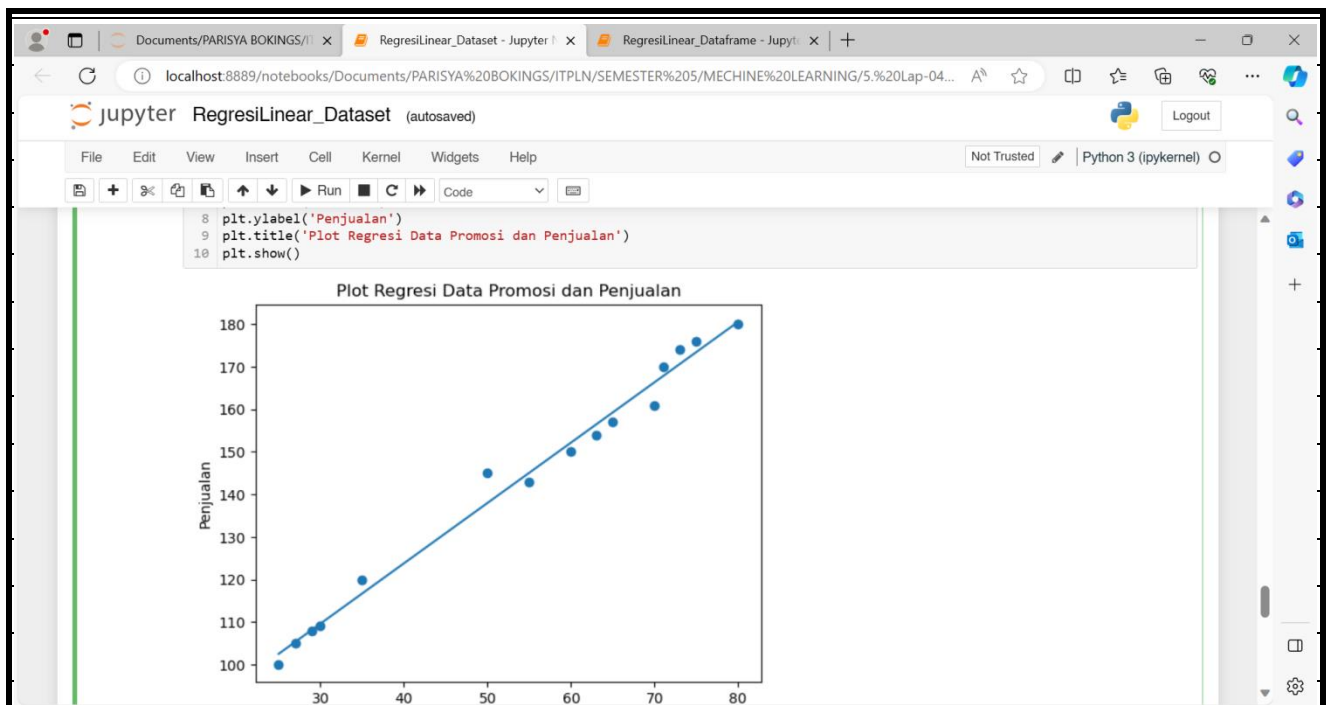
Testing

```
In [65]: 1 #dataset1
2 x_test_baru = [[1.1],[11],[7]]
3 pred_salary_baru = model1.predict(x_test_baru)
4
5 #dataset2
6 promosi_test = [[20],[100]]
7 pred_penjualan = model2.predict(promosi_test)
8
9 print("Prediksi Gaji: ", pred_salary_baru)
10 print("Prediksi Penjualan: ", pred_penjualan)
```

Prediksi Gaji: [35243.16252012 128797.78950253 90997.94021671]
Prediksi Penjualan: [95.38781949 208.95989944]

Visualisasi

```
In [59]: 1 import matplotlib.pyplot as plt
2
3 prediksi_penjualan = model2.predict(promosi)
4 dataframe_baru = pd.DataFrame({'promosi': data2.promosi, 'penjualan': prediksi_penjualan})
5 plt.scatter(data2.promosi, data2.penjualan)
6 plt.plot(dataframe_baru.promosi, dataframe_baru.penjualan)
7 plt.xlabel('Promosi')
8 plt.ylabel('Penjualan')
```



Documents/PARISYA BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04... RegresiLinear_Dataset - Jupyter RegresiLinear_Dataframe - Jupyter

localhost:8889/notebooks/Documents/PARISYA%20BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04...

Jupyter RegresiLinear_Dataset (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Cek Akurasi

```
In [74]: 1 from sklearn.metrics import r2_score
2 akurasi = r2_score(data2,dataframe_baru)
3 print(akurasi*100)
```

99.37098507592287

karena kalau akurasi 100% bisa dikatakan dia ada indikasi overfitting.

Kurang dari 70% itu kurang baik, tapi kalau 70-80% itu akurasi sudah baik.

Pembuktian Rumus

```
In [78]: 1 #Rumus Regresi y = a+bX (a = intercept, b = koefisien)
2
3 #dataset1
4 print("Koefisien: ", model1.coef_)
5 print("Intercept: ", model1.intercept_)
6
7 #dataset2
8 print("Koefisien: ", model2.coef_)
9 print("Intercept: ", model2.intercept_)
```

Koefisien: [9449.96232146]
Intercept: 24848.203966523193
Koefisien: [1.419651]
Intercept: 66.99479949720258

```
In [79]: 1 #testing rumus
2
3 #dataset1
4 y = model1.intercept_+ model1.coef_*2.1 #x nya adalah x testing
5 print(y)
6
7 #dataset2
8 y = model2.intercept_+ model2.coef_*27 #x nya adalah x testing
9 print(y)
```

[44693.12484158]
[105.32537648]

Praktikum DataFrame Tinggi dan Berat Badan

```
Documents/PARISYA BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04...
jupyter RegresiLinear_Dataframe Last Checkpoint: an hour ago (autosaved)
Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help
+ Run Code

PARISYA BOKINGS 202131119

In [1]: 1 #Import Library
        2 import pandas as pd
        3 from sklearn.model_selection import train_test_split
        4 from sklearn.linear_model import LinearRegression
        5 from sklearn.metrics import mean_squared_error, r2_score
        6 import matplotlib.pyplot as plt

In [2]: 1 #Membuat DataFrame Tinggi dan Berat badan
        2 data = {
        3     'Tinggi': [150, 155, 160, 165, 170],
        4     'Berat_Badan': [55, 60, 65, 70, 75]
        5 }
        6
        7 df = pd.DataFrame(data)

In [3]: 1 x = df[['Tinggi']]
        2 y = df[['Berat_Badan']]

In [4]: 1 #membagi data menjadi data Latih dan data diuji
        2 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

In [5]: 1 #Membuat model regresi linear
        2 model = LinearRegression()
```

```
Documents/PARISYA BOKINGS/ITPLN/SEMESTER%205/MECHINE%20LEARNING/5.%20Lap-04...
jupyter RegresiLinear_Dataframe Last Checkpoint: an hour ago (autosaved)
Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help
+ Run Code

In [6]: 1 #Melatih model dengan data Latih
        2 model.fit(x_train, y_train)

Out[6]: LinearRegression()

In [7]: 1 #Membuat prediksi
        2 y_pred = model.predict(x_test)

In [8]: 1 #Menganalisis model
        2 mse = mean_squared_error(y_test, y_pred)
        3 r2 = r2_score(y_test, y_pred)
        4
        5 print(f'Mean Squared Error: {mse}') #untuk mencetak nilai MSE dari model regresi linear.
        6 print(f'R-squared: {r2}') #mencetak nilai koefisien dari model regresi linear

Mean Squared Error: 0.0
R-squared: nan
```

