

Hands-On in

Hands-On ini digunakan pada kegiatan Microcredential Associate Data Scientist 2021

Definisi Matriks

Matriks dalam dunia matematika merupakan suatu bilangan, simbol, ataupun ekspresi yang disusun dalam baris dan kolom yang membentuk suatu bidang persegi/persegi panjang. Matriks memiliki ukuran/orde. Pada Python, matriks dapat dibuat sesuai kebutuhan dengan menentukan ukuran/orde dari matriks. Bentuk umum ukuran/orde matriks adalah mxn, m adalah banyak baris dan n adalah banyak kolom.

Contoh Membuat Matriks Orde 2x2

Berikut ini adalah syntax untuk membuat sebuah matriks yang diberi nama matriksA dengan ukuran/orde 2x2. Matriks orde 2x2 merupakan matriks yang memiliki nilai pada 2 baris dan 2 kolom. Pada contoh: matriksA memiliki: Baris 1 Kolom 1, nilai = 1 Baris 1 Kolom 2, nilai = 0 Baris 2 Kolom 1, nilai = 0 Baris 2 Kolom 2, nilai = 1

```
In [20]: #Matriks dengan ukuran 2x2
matriksA = [[1,0],[0,1]]
#Menampilkan matriks dengan nama matriksA
print (matriksA)

[[1, 0], [0, 1]]
```

Contoh Membuat Matriks Orde 3x3

Berikut ini adalah syntax untuk membuat sebuah matriks yang diberi nama matriksB dengan ukuran/orde 3x3. Matriks orde 3x3 merupakan matriks yang memiliki nilai pada 3 baris dan 3 kolom. Pada contoh: matriksB memiliki: Baris 1 Kolom 1, nilai = 1 Baris 1 Kolom 2, nilai = 0 Baris 1 Kolom 3, nilai = 1 Baris 2 Kolom 1, nilai = 0 Baris 2 Kolom 2, nilai = 1 Baris 2 Kolom 3, nilai = 0 Baris 3 Kolom 1, nilai = 1 Baris 3 Kolom 2, nilai = 1 Baris 3 Kolom 3, nilai = 1

```
In [20]: #Matriks dengan ukuran 3x3
matriksB = [[1,0,1],[0,1,0],[1,0,1]]
#Menampilkan matriks dengan nama MatriksB
print (matriksB)

[[1, 0, 1], [0, 1, 0], [1, 0, 1]]
```

Latihan (1)

Buatlah matriksC dengan orde 2x3, dan nilai yang diberikan pada matriksC adalah sebagai berikut: Baris 1 Kolom 1, nilai = 1 Baris 1 Kolom 2, nilai = 2 Baris 1 Kolom 3, nilai = 3 Baris 2 Kolom 1, nilai = 4 Baris 2 Kolom 2, nilai = 5 Baris 2 Kolom 3, nilai = 6

```
In [20]: #Latihan(1)
#Matriks dengan ukuran 2x3
MatriksC = [[1, 2, 3],
            [4, 5, 6]]
#Menampilkan matriks dengan nama MatriksC
print(MatriksC)

[[1, 2, 3], [4, 5, 6]]
```

Membuat Matriks dengan Menggunakan Perulangan

Matriks dapat juga dibuat dengan menggunakan fungsi perulangan. Fungsi perulangan atau biasa disebut looping atau iterasi memerlukan test kondisi. Bila hasil test kondisi True, maka blok kode kembali dieksekusi. Tapi jika False, maka keluar dari perulangan. Di python, perulangan bisa dilakukan dengan dua cara atau metode, yaitu: menggunakan For atau menggunakan While.

Contoh Membuat Matriks dengan Perulangan FOR

Pada contoh berikut ini, dibuat matriks dengan menggunakan perulangan for. Matriks mempunyai orde mxn, yaitu orde 2x3, dan nilai yang diberikan adalah Baris 1 Kolom 1, nilai = 1 Baris 1 Kolom 2, nilai = 1 Baris 1 Kolom 3, nilai = 1 Baris 2 Kolom 1, nilai = 1 Baris 2 Kolom 2, nilai = 1 Baris 2 Kolom 3, nilai = 1

```
In [210]: #Matriks dengan menggunakan fungsi perulangan
m = 2
n = 3
x = 0
#Fungsi perulangan dengan menggunakan for
for i in range(m):
    x+=1
    #Menampilkan matriks
    print (x)

[[1, 1, 1], [1, 1, 1]]
```

Latihan (2)

Buatlah matriks dengan orde 3x2, dengan nilai adalah: Baris 1 Kolom 1, nilai = 2 Baris 1 Kolom 2, nilai = 2 Baris 2 Kolom 1, nilai = 2 Baris 2 Kolom 2, nilai = 2 Baris 3 Kolom 1, nilai = 2 Baris 3 Kolom 2, nilai = 2

```
In [21]: #Latihan(2)
#Matriks dengan menggunakan fungsi perulangan
matriksD = [[0]*2]*3
#Fungsi perulangan dengan menggunakan for
for i in range(len(matriksD)):
    for j in range(len(matriksD[i])):
        matriksD[i][j] = 2
#Menampilkan matriks
print(matriksD)

[[2, 2], [2, 2], [2, 2]]
```

Membuat Matriks dengan Menggunakan Library Numpy

Library pada Python merupakan sebutan untuk kode program tambahan yang digunakan dalam kebutuhan tertentu. Python mempunyai lebih dari 140.000 library yang dikembangkan melalui open source project.

Library Numpy memiliki kegunaan untuk operasi vektor dan matriks. Fituranya hampir sama dengan SciKit-Learn dalam mengelola array dan array multidimensi. Numpy merupakan salah satu library yang digunakan oleh library lain seperti Matplotlib untuk keperluan analisis data.

Contoh Membuat Matriks dengan Numpy

Pada contoh berikut dibuat matriks dengan panjang elemen sebanyak 12 dengan nilai elemen dimulai dari indeks 0 sampai 11, dengan ukuran/orde matriks adalah 4x3.

```
In [212]: #Panggil Library Numpy
import numpy as np
#Menentukan panjang elemen sebanyak 12
matriks = range(12)
#Matriks orde adalah 4x3
matriks = np.reshape(matriks,(4,3))
#Menampilkan matriks
matriks

array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8],
       [9, 10, 11]])
```

Latihan (3)

Buatlah matriks dengan menggunakan Library Numpy dengan panjang elemen adalah 32 dan matriks memiliki orde 4x8

```
In [213]: #Latihan(3)
#Panggil Library Numpy
import numpy as np
#Menentukan panjang elemen sebanyak 30
mat = range(32)
#Matriks orde adalah 4x8
mat = np.reshape(mat, (4, 8))
#Menampilkan matriks
mat

array([[0, 1, 2, 3, 4, 5, 6, 7],
       [8, 9, 10, 11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20, 21, 22, 23],
       [24, 25, 26, 27, 28, 29, 30, 31]])
```

Membuat Matriks dengan Nilai Random

Jika ingin elemen pada matriks bernilai random, dapat menggunakan fungsi random yang tersedia pada numpy.

Contoh Membuat Matriks Random dengan Numpy

Pada contoh berikut ini dibuat matriks dengan ukuran/orde 3x4 dengan nilai elemen dimulai dari indeks 1 sampai 4 secara random.

```
In [214]: #Memanggil Library numpy dan diberikan nama alias np
import numpy as np
#Membuat matriks ukuran 3x4 dengan random
matriks = np.random.randint(1,4,(3,4))
#Menampilkan matriks
print (matriks)

[[3 3 3]
 [2 1 2]
 [2 2 2]]
```

Latihan (4)

Buatlah matriks dengan menggunakan Library Numpy dengan orde 3x30, dan nilai elemen dimulai dari indeks 1 sampai dengan 5 secara random.

```
In [215]: #Latihan(4)
#Memanggil Library numpy dan diberikan nama alias np
import numpy as np
#Membuat matriks ukuran 3x30 dengan random
matriks = np.random.randint(1, 5, (3, 30))
#Menampilkan matriks
matriks

array([[3, 1, 1, 2, 3, 4, 2, 3, 4, 2, 3, 1, 1, 3, 2, 2, 2, 4, 3, 1, 3, 3, 2, 3, 4, 1,
       3, 3, 4, 2, 3, 2, 4, 2, 4, 2, 4, 2, 4, 2, 2, 4, 2, 2, 1, 3, 4,
       3, 1, 1, 3, 2, 1, 3, 4],
       [2, 1, 2, 3, 3, 2, 2, 4, 2, 3, 4, 4, 1, 1, 1, 1, 1, 1, 4, 2, 2, 3, 3, 3,
       1, 1, 4, 1, 2, 4, 2, 2]),
       [1, 1, 1, 2, 4, 2, 2])
```

Penjumlahan Matriks

Penjumlahan matriks dilakukan dengan menjumlahkan setiap elemen, menggunakan tanda plus (+). Hasil penjumlahan tersebut akan menjadi elemen baru. Masing - masing matriks diakses setiap elemennya pada koordinat yang sama kemudian dijumlahkan untuk mendapatkan elemen baru. Penjumlahan matriks dilakukan pada dua buah matriks yang memiliki orde sama.

```
In [216]: #Penjumlahan Matriks
#Contoh 1
mat1 = [
    [5, 0],
    [2, 6],
]
mat2 = [
    [1, 0],
    [4, 2],
]
for x in range(0, len(mat1)):
    for y in range(0, len(mat1[0])):
        print (mat1[x][y] + mat2[x][y], end= ' '),
        print

6 0 6 8
```

```
In [216]: #Penjumlahan Matriks
#Contoh 2
import numpy as np
matriksA = np.random.randint(1, 4,(3, 4))
matriksB = np.random.randint(1, 3,(3, 4))
print (matriksA)
print ( ) #memberikan spasi baris pada matriks yang tercetak
print (matriksB)
print ( )

#Penjumlahan matriksA dan matriks B
for x in range(0, len(matriksA)):
    print ('Hasil Penjumlahan matriksA dan matriksB', 'Baris ke-', x+1)
    print ( )
    for y in range(0, len(matriksA[0])):
        print (matriksA[x][y] + matriksB[x][y], end= ' '),
        print ( )

[[1 3 1 2]
 [2 2 2 3]
 [1 2 2 2]]

[[1 1 2 1]
 [2 1 2 2]
 [1 2 2 2]]

Hasil Penjumlahan matriksA dan matriksB Baris ke- 1
2 4 3
Hasil Penjumlahan matriksA dan matriksB Baris ke- 2
4 3 4 5
Hasil Penjumlahan matriksA dan matriksB Baris ke- 3
2 2 4 4
```

```
In [218]: #Penjumlahan Matriks
#Contoh 3
import numpy as np
matriksA = np.random.randint(1, 4,(6, 10))
matriksB = np.random.randint(1, 3,(6, 10))
print (matriksA)
print ( ) #memberikan spasi baris pada matriks yang tercetak
print (matriksB)
print ( )

#Penjumlahan matriksA dan matriks B
for x in range(0, len(matriksA)):
    print ( )
    for y in range(0, len(matriksA[0])):
        print (matriksA[x][y] + matriksB[x][y], end= ' '),
        print ( )

[[3 3 3 1 1 2 1 2 2 2]
 [1 2 3 1 1 2 2 3 3 2]
 [1 2 1 3 1 1 3 3 2 2]
 [2 2 1 3 3 2 2 3 3 1]
 [2 1 2 2 3 1 2 1 2 1]
 [2 3 1 1 1 3 3 3 2 1]]

[[2 2 2 2 1 2 2 2 2 1]
 [1 2 1 2 1 2 1 1 1 1]
 [1 1 1 2 1 2 2 1 1 1]
 [1 2 1 2 2 2 1 1 2 3]
 [1 1 2 1 1 2 1 1 1 1]
 [1 2 1 1 2 2 1 1 1 2]]

5 5 5 5 2 3 4 3 4 3
2 4 4 2 3 3 4 4 4 3
2 3 5 2 3 5 4 3 3
3 4 2 5 5 3 2 5 2
3 2 3 4 3 4 2 3 2 3
3 5 2 2 3 5 4 4 3 3
```

Latihan (5)

Buatlah penjumlahan dua matriks dengan menggunakan library Numpy, dengan matriks berukuran 7x9 yang nilainya didapatkan secara random.

```
In [219]: #Latihan(5)
#Penjumlahan Matriks
matA = np.random.randint(10, 50, (7, 9))
matB = np.random.randint(10, 20, (7, 9))
result = matA + matB
#penjumlahan matriksA dan matriks B
result

array([[18, 18, 13, 18, 17, 19, 21, 14, 16],
       [20, 18, 15, 21, 15, 17, 16, 16, 19],
       [20, 14, 16, 14, 18, 15, 14, 19, 20],
       [15, 16, 20, 18, 20, 14, 17, 20, 18],
       [13, 16, 16, 19, 20, 17, 15, 11, 14],
       [14, 17, 16, 15, 16, 19, 23, 16, 20],
       [15, 21, 20, 14, 16, 13, 15, 17, 21]])
```

Pengurangan Matriks

Pengurangan matriks menggunakan operator dengan tanda kurang (-). Matriks baru akan terbentuk sebagai hasil dari pengurangan setiap kedua elemen matriks. Pengurangan matriks dilakukan pada dua matriks yang memiliki orde yang sama.

```
In [220]: #Pengurangan Matriks
#Contoh 1
mat1 = [
    [5, 0],
    [2, 6],
]
mat2 = [
    [1, 0],
    [4, 2],
]
for x in range(0, len(mat1)):
    for y in range(0, len(mat1[0])):
        print (mat1[x][y] - mat2[x][y], end= ' '),
        print

4 0 -2 4
```

```
In [221]: #Pengurangan Matriks
#Contoh 2
mat1 = [
    [5, 0],
    [2, 6],
]
mat2 = [
    [1, 0],
    [4, 2],
]
for x in range(0, len(mat1)):
    for y in range(0, len(mat1[0])):
        print (mat1[x][y] - mat2[x][y], end= ' '),
        print ( )

4 0
-2 4
```

```
In [222]: #Pengurangan Matriks
#Contoh 3
import numpy as np
matriksA = np.random.randint(1, 4,(3, 4))
matriksB = np.random.randint(1, 3,(3, 4))
print (matriksA)
print ( ) #memberikan spasi baris pada matriks yang tercetak
print (matriksB)
print ( )

#Pengurangan matriksA dengan matriks B
for x in range(0, len(matriksA)):
    print ('Hasil Pengurangan matriksA dan matriksB', 'Baris ke-', x+1)
    print ( )
    for y in range(0, len(matriksA[0])):
        print (matriksA[x][y] - matriksB[x][y], end= ' '),
        print ( )

[[3 2 1 1]
 [2 3 1 3]
 [3 3 3 3]]

[[1 2 1 2]
 [2 2 1 1]
 [1 2 1 2]]

Hasil Pengurangan matriksA dan matriksB Baris ke- 1
2 0 0 -1
Hasil Pengurangan matriksA dan matriksB Baris ke- 2
0 1 0 2
Hasil Pengurangan matriksA dan matriksB Baris ke- 3
1 2 2 1
```

```
In [223]: #Pengurangan Matriks
#Contoh 4
import numpy as np
matriksA = np.random.randint(1, 4,(6, 10))
matriksB = np.random.randint(1, 3,(6, 10))
print (matriksA)
print ( ) #memberikan spasi baris pada matriks yang tercetak
print (matriksB)
print ( )

#Pengurangan matriksA dengan matriks B
for x in range(0, len(matriksA)):
    print ( )
    for y in range(0, len(matriksA[0])):
        print (matriksA[x][y] - matriksB[x][y], end= ' '),
        print ( )

[[1 3 3 3 1 2 2 1 2 1 3]
 [1 2 3 3 2 1 2 1 2 1 1]
 [3 1 3 3 2 3 1 2 3 2]
 [2 3 3 1 3 2 3 1 2]
 [2 1 2 1 1 1 1 2 3]
 [1 2 2 1 3 3 1 1 2 3]]

[[2 1 2 2 2 1 2 1 2 1 1]
 [2 1 2 2 1 2 2 2 2 1 1]
 [1 1 1 1 2 2 1 1 1 1]
 [2 2 2 2 2 2 2 2 2 2]
 [1 2 1 1 1 2 2 1 1 1]
 [1 2 1 1 1 2 1 2 1 1]]

-1 1 2 1 -1 0 0 -1 1 2 2
-1 1 1 1 1 -1 0 -1 0 0
2 0 2 2 1 0 -1 1 2 1
0 1 1 0 1 0 1 1 0 1
1 -1 1 0 0 0 -1 -1 1 2
0 0 1 1 0 1 0 -1 1 2
```

Latihan (6)

Buatlah pengurangan dua matriks dengan menggunakan library Numpy, dengan matriks berukuran 7x9 yang nilainya didapatkan secara random.

```
In [224]: #Latihan(6)
#Pengurangan Matriks
matA = np.random.randint(15, 25, (7, 9))
matB = np.random.randint(10, 20, (7, 9))
#pengurangan matriksA dengan matriks B
result = matA - matB
result

array([[2, 9, 4, 13, 6, 9, 2, 6, 5],
       [5, 7, 0, 7, 6, 7, 7, 1, 1],
       [8, 6, 7, 8, -3, 2, 5, 3, 5],
       [-2, 10, 2, 6, -2, 8, 10, 2, 0],
       [8, 0, 0, 4, 4, 4, 3, 1, 9],
       [8, 11, 1, 6, 3, 0, 3, 11, 4],
       [7, 5, 5, -1, 3, 2, 6, 8, 3]])
```

Penggunaan Library Pandas

Latihan (7)

Lakukan import Library Pandas

```
In [225]: #Latihan(7)
#Import Library Pandas
import pandas as pd
```

Latihan (8)

Panggil file dengan format .csv yang bernama "cloth_data"

```
In [226]: #Latihan(8)
#Manggil file bernama cloth_data.csv
df = pd.read_csv('cloth_data.csv')
```

Latihan (9)

Tampilkan data dari "cloth_data"

```
In [227]: #Latihan(9)
#Tampilkan data dari dataset cloth_data
df.head(11)
```

	weight	age	height	size
0	62	280	17272	XL
1	59	360	16764	L
2	61	340	16510	M
3	65	270	17526	L
4	62	450	17272	M
5	50	270	16002	S
6	53	650	16002	M
7	51	330	16002	XXS
8	54	260	16764	M
9	53	320	16510	S
10	63	300	17018	XXXL
11	77	350	17272	XXXL
12	64	260	16510	L
13	52	280	16002	M
14	65	330	16510	L
15	63	300	16764	L
16	54	210	16764	XXS
17	63	270	17272	M
18	63	300	16764	M
19	54	200	16764	S
20	55	320	15494	S
21	55	370	16002	M
22	55	500	16510	S
23	50	430	16002	M
24	74	290	17780	XL
25	64	320	16002	XL
26	58	290	16764	XXS
27	54	470	16510	M
28	58	310	15494	XL
29	65	270	17272	L
30	47	430	15748	S
31	61	270	16510	M
32	58	290	17272	M
33	86	480	17272	XXL
34	68	360	16256	XL
35	65	270	17018	L
36	65	340	17018	L
37	54	400	15748	M
38	52	450	15494	M
39	52	300	17018	S
40	58	520	16256	XXL
41	78	370	17526	XXXL
42	61	340	17272	L
43	49	240	16764	S
44	61	280	16256	M
45	64	340	16256	XXXL
46	63	250	16764	S
47	56	320	15748	M
48	50	270	17018	S
49	77	290	17272	XXXL

Latihan (10)

Tampilkan data dari "cloth_data" mulai dari indeks 0 (nol) sampai dengan indeks 10 (sepuluh)

```
In [228]: #Latihan(10)
#Tampilkan data dari "cloth_data" mulai dari indeks 0 (nol) sampai dengan indeks 10 (sepuluh)
df.head(11)
```

	weight	age	height	size
0	62	280	17272	XL
1	59	360	16764	L
2	61	340	16510	M
3	65	270	17526	L
4	62	450	17272	M
5	50	270	16002	S
6	53	650	16002	M
7	51	330	16002	XXS
8	54	260	16764	M
9	53	320	16510	S
10	63	300	17018	XXXL

Latihan (11)

Tampilkan data dari "cloth_data" untuk kolom age dan size

```
In [229]: #Latihan(11)
#Tampilkan data dari "cloth_data" untuk kolom age dan size
df[['age', 'size']]
```

	age	size
0	280	XL
1	360	L
2	340	M
3	270	L
4	450	M
5	270	S
6	650	M
7	330	XXS
8	260	M
9	320	S
10	300	XXXL

Penggunaan Library Matplotlib

Latihan (13)

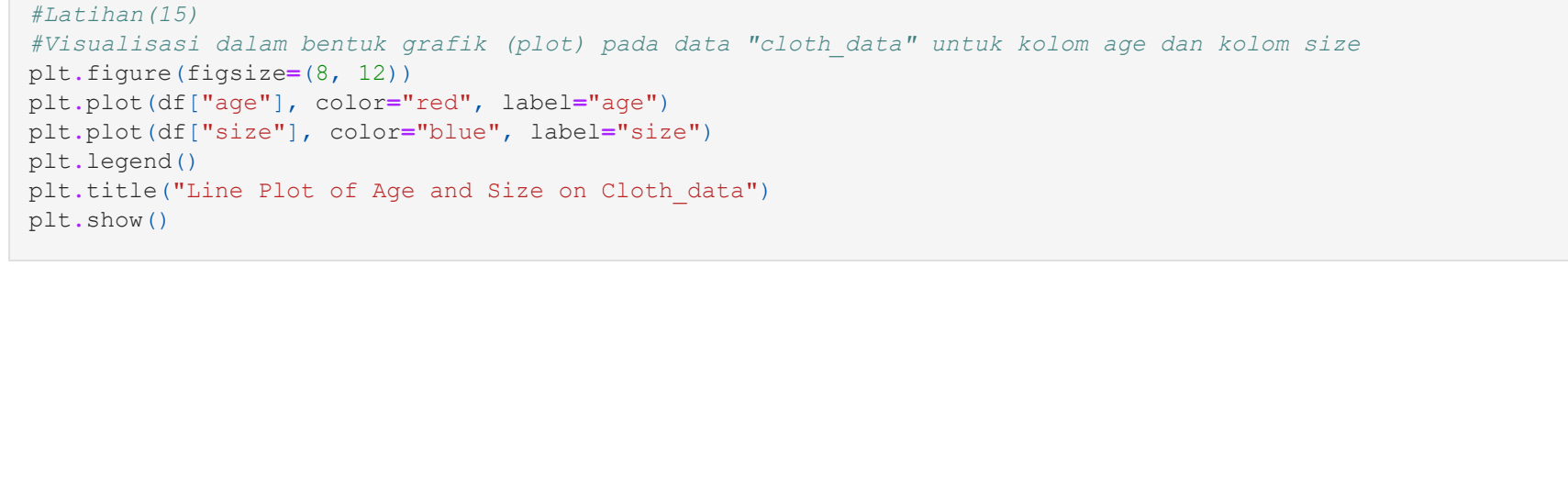
Lakukan import Library Matplotlib

```
In [231]: #Latihan(13)
#Import Library Matplotlib
import matplotlib.pyplot as plt
```

Latihan (14)

Buatlah visualisasi dalam bentuk histogram pada data "cloth_data" untuk kolom size

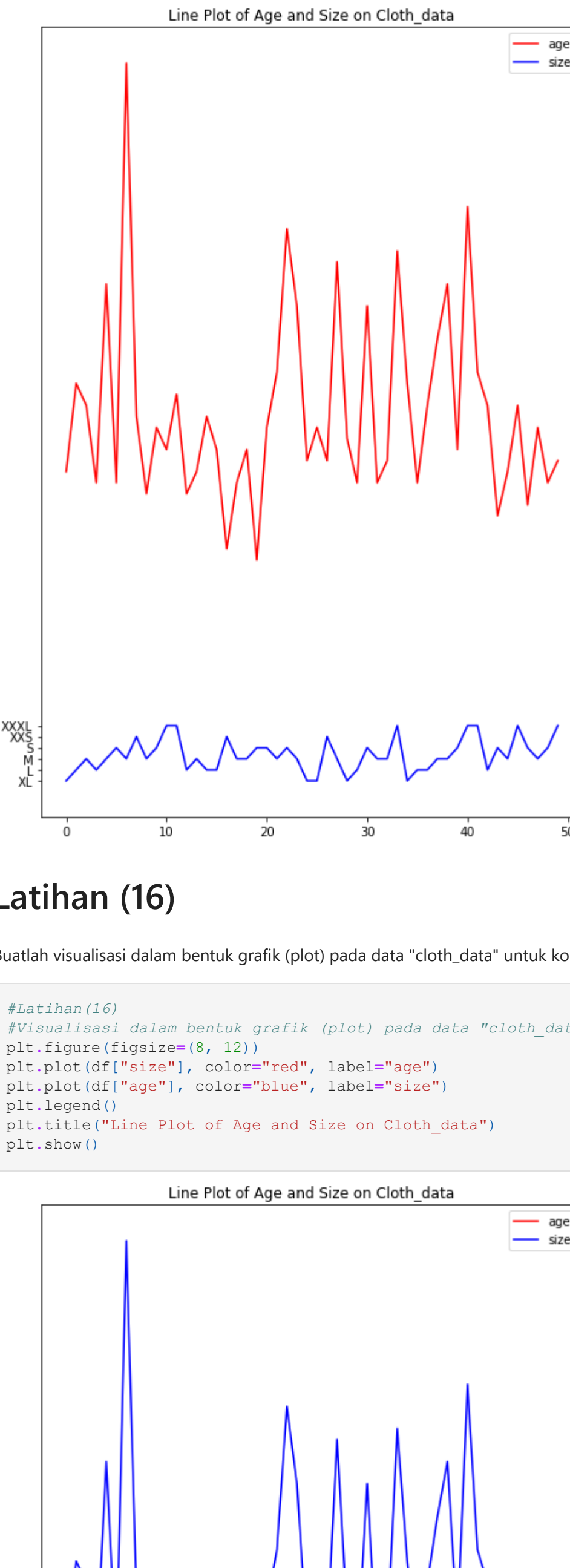
```
In [232]: #Latihan(14)
#Buat histogram untuk kolom size
plt.hist(df['size'])
plt.xlabel("Size")
plt.ylabel("Frequency")
plt.title("Every Size Frequency on Cloth_data")
plt.show()
```



Latihan (15)

Buatlah visualisasi dalam bentuk grafik (plot) pada data "cloth_data" untuk kolom age dan kolom size

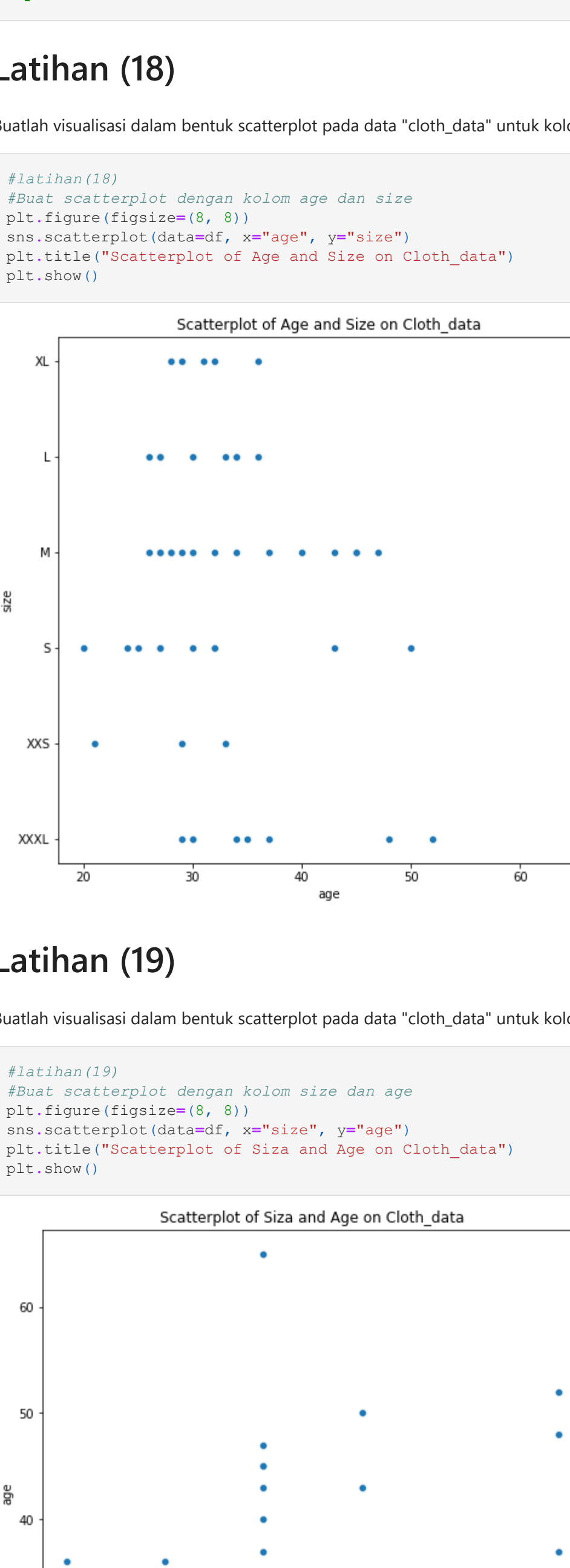
```
In [233]: #Latihan(15)
#Visualisasi dalam bentuk grafik (plot) pada data "cloth_data" untuk kolom age dan kolom size
plt.figure(figsize=(8, 12))
plt.plot(df['age'], color='red', label='age')
plt.plot(df['size'], color='blue', label='size')
plt.legend()
plt.title("LinePlot of Age and Size on Cloth_data")
plt.show()
```

Latihan (16)

Buatlah visualisasi dalam bentuk grafik (plot) pada data "cloth_data" untuk kolom size dan kolom age

```
In [234]: #Latihan(16)
#Visualisasi dalam bentuk grafik (plot) pada data "cloth_data" untuk kolom size dan kolom age
plt.figure(figsize=(8, 10))
plt.plot(df["size"], color="red", label="age")
plt.plot(df["age"], color="blue", label="size")
plt.legend()
plt.title("Line Plot of Age and Size on Cloth_data")
plt.show()
```



Latihan (17)

Import library seaborn

```
In [235]: #Latihan(17)
#Import library seaborn
import seaborn as sns
```

Latihan (18)

Buatlah visualisasi dalam bentuk scatterplot pada data "cloth_data" untuk kolom age dan kolom size

```
In [236]: #Latihan(18)
#Buat scatterplot dengan kolom age dan size
plt.figure(figsize=(8, 8))
sns.scatterplot(data=df, x="age", y="size")
plt.title("Scatterplot of Age and Size on Cloth_data")
plt.show()
```



Latihan (19)

Buatlah visualisasi dalam bentuk scatterplot pada data "cloth_data" untuk kolom size dan kolom age

```
In [237]: #Latihan(19)
#Buat scatterplot dengan kolom size dan age
plt.figure(figsize=(8, 8))
sns.scatterplot(data=df, x="size", y="age")
plt.title("Scatterplot of Size and Age on Cloth_data")
plt.show()
```



Latihan (20)

Buatlah visualisasi dalam bentuk histogram pada data "cloth_data" untuk kolom age

```
In [238]: #Latihan(20)
#Buat histogram untuk kolom size
plt.figure(figsize=(8, 8))
sns.histplot(data=df, x="age", kde=True, bins=10)
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Every Age Frequency on Cloth_data")
plt.show()
```

