



## Hands-On

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 [1]: #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 = 0 Baris 3 Kolom 3, nilai = 1

```
In [2]: #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 [3]: #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 tes kondisi. Bila hasil tes 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 [4]: #Matriks dengan menggunakan fungsi perulangan
m = 2
n = 3
x = [0]*m
#Fungsi perulangan dengan menggunakan for
for i in range(m):
    x[i] = [1]*n
#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 [9]: #Latihan(2)
#Matriks dengan menggunakan fungsi perulangan
m = 3
n = 2
x = [0]*m
#Fungsi perulangan dengan menggunakan for
for i in range(m):
    x[i] = [2]*n

#Menampilkan matriks
print(x)

[[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. Fiturnya hampir sama dengan MATLAB dalam mengelola array dan array multidimensi. Numpy merupakan salah satu library yang digunakan oleh library lain seperti Scikit-Learn 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 [11]: #Panggil Library Numpy
from numpy import *
#Menentukan panjang elemen sebanyak 12
matriks = range(12)
#Matriks orde adalah 4x3
matriks = reshape(matriks,(4,3))
#Menampilkan matriks
print (matriks)
```

```
[[ 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 [3]: #Latihan(3)
#Panggil Library Numpy
from numpy import *

#Menentukan panjang elemen sebanyak 30
matriks = range(32)

#Matriks orde adalah 4x8
matriks = reshape(matriks,(4,8))

#Menampilkan matriks
print(matriks)
```

```
[[ 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 [4]: #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)
```

```
[[1 3 3 3]
 [3 2 3 1]
 [2 1 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 [6]: #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
print(matriks)

[[1 3 4 1 1 3 1 1 4 1 4 1 2 4 1 4 1 1 4 2 2 3 2 2 2 4 2 2 2 1 4]
 [4 1 4 1 4 2 1 3 4 3 3 1 4 3 2 4 2 4 2 1 2 4 3 1 2 4 3 1 1 4]
 [4 1 3 2 1 1 3 2 1 4 3 4 1 4 4 1 3 2 4 3 1 3 3 2 4 3 4 4 4 4]]
```

## 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 [7]: #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 [7]: #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 ()
```

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

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

```
Hasil Penjumlahan matriksA dan matriksB Baris ke- 1
```

```
3 3 4 3
```

```
Hasil Penjumlahan matriksA dan matriksB Baris ke- 2
```

```
4 3 4 5
```

```
Hasil Penjumlahan matriksA dan matriksB Baris ke- 3
```

```
5 2 5 3
```

```
In [5]: #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 ()
```

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

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

2 5 3 2 5 4 5 3 3 3

2 5 5 2 3 3 5 4 4 3

2 4 2 4 3 3 5 3 2 4

4 5 3 4 4 5 4 2 5 5

5 4 4 4 4 3 4 4 5 2

3 3 3 3 2 5 5 4 3 2

## Latihan (5)

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

```
In [11]: #Latihan(5)
#Penjumlahan Matriks
import numpy as np
matriksA = np.random.randint(1,4,(7,9))
matriksB = np.random.randint(1,3,(7,9))
print (matriksA)
print ()
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 2 2 3 3 2 2 1 2]
 [1 3 1 2 2 3 3 2 1]
 [1 1 2 1 1 3 1 1 3]
 [1 3 1 1 3 1 1 3 3]
 [2 1 2 3 3 3 1 2 2]
 [1 1 2 2 1 2 2 1 3]
 [1 2 2 2 1 2 1 2 2]]
```

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

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

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

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

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

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

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

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

## 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 [12]: #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 [13]: #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 [14]: #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 ()
```

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

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

```
Hasil Pengurangan matriksA dan matriksB Baris ke- 1
```

```
0 -1 1 1
```

```
Hasil Pengurangan matriksA dan matriksB Baris ke- 2
```

```
1 0 0 2
```

```
Hasil Pengurangan matriksA dan matriksB Baris ke- 3
```

```
0 0 2 -1
```

```
In [15]: #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 ()
```

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

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

1 0 1 1 0 1 0 -1 -1 -1 -1

1 1 1 0 1 -1 0 0 0 2

0 -1 2 0 2 1 2 0 1 0

0 0 0 0 1 0 1 -1 0 1

2 1 0 1 0 0 0 0 0 0

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

## Latihan (6)

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

```
In [19]: #Latihan(6)
#Pengurangan Matriks
import numpy as np
matriksA = np.random.randint(1,5,(7,9))
matriksB = np.random.randint(1,4,(7,9))
print(matriksA)
print()
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()
```

```
[[2 4 3 1 3 4 1 2 1]
 [1 2 3 4 3 2 2 3 4]
 [3 3 3 4 2 2 2 4 3]
 [2 3 1 4 1 1 2 4 1]
 [4 4 3 2 1 3 3 3 1]
 [3 4 3 4 2 4 1 3 2]
 [3 2 4 2 2 3 3 3 1]]
```

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

```
1 3 2 -1 0 1 0 0 -1
-2 1 0 2 2 -1 1 0 3
0 2 2 1 -1 0 -1 2 1
1 0 0 1 0 -2 -1 3 -1
2 2 1 -1 -1 0 2 1 -2
0 1 1 3 1 1 -2 0 1
1 -1 2 -1 -1 2 0 1 -2
```

## Penggunaan Library Pandas

### Latihan (7)

Lakukan import Library Pandas

```
In [8]: #Latihan(7)
#Import Library Pandas
import pandas as pd
```

### Latihan (8)

Panggil file dengan format .csv yang bernama "cloth\_data"

```
In [9]: #Latihan(8)
#Panggil file bernama cloth_data.csv

df = pd.read_csv("cloth_data.csv")
```

## Latihan (9)

Tampilkan data dari "cloth\_data"

```
In [10]: #latihan(9)
#Tampilkan data dari dataset cloth_data
df
```

Out[10]:

	weight	age	height	size
0	62	28.0	172.72	XL
1	59	36.0	167.64	L
2	61	34.0	165.10	M
3	65	27.0	175.26	L
4	62	45.0	172.72	M
5	50	27.0	160.02	S
6	53	65.0	160.02	M
7	51	33.0	160.02	XXS
8	54	26.0	167.64	M
9	53	32.0	165.10	S
10	63	30.0	170.18	XXXL
11	77	35.0	172.72	XXXL
12	64	26.0	165.10	L
13	52	28.0	160.02	M
14	65	33.0	165.10	L
15	63	30.0	167.64	L
16	54	21.0	167.64	XXS
17	63	27.0	172.72	M
18	63	30.0	167.64	M
19	54	20.0	167.64	S
20	55	32.0	154.94	S
21	55	37.0	160.02	M
22	55	50.0	165.10	S
23	50	43.0	160.02	M
24	74	29.0	177.80	XL
25	64	32.0	160.02	XL
26	58	29.0	167.64	XXS
27	54	47.0	165.10	M
28	58	31.0	154.94	XL
29	65	27.0	172.72	L
30	47	43.0	157.48	S
31	61	27.0	165.10	M
32	58	29.0	172.72	M
33	86	48.0	172.72	XXXL
34	68	36.0	162.56	XL
35	65	27.0	170.18	L
36	65	34.0	170.18	L
37	54	40.0	157.48	M
38	52	45.0	154.94	M
39	52	30.0	170.18	S
40	58	52.0	162.56	XXXL
41	78	37.0	175.26	XXXL

	weight	age	height	size
42	61	34.0	172.72	L
43	49	24.0	167.64	S
44	61	28.0	162.56	M
45	64	34.0	162.56	XXXL
46	63	25.0	167.64	S
47	56	32.0	157.48	M
48	50	27.0	170.18	S
49	77	29.0	172.72	XXXL

## Latihan (10)

Tampilkan data dari "cloth\_data" mulai dari indeks 0 (nol) sampai dengan indeks 10 (sepuluh)

```
In [112]: #Latihan(10)
#Tampilkan data dari "cloth_data" mulai dari indeks 0 (nol) sampai dengan indeks 10 (sepuluh)

df.iloc[0:11]
```

Out[112]:

	weight	age	height	size
0	62	28.0	172.72	XL
1	59	36.0	167.64	L
2	61	34.0	165.10	M
3	65	27.0	175.26	L
4	62	45.0	172.72	M
5	50	27.0	160.02	S
6	53	65.0	160.02	M
7	51	33.0	160.02	XXS
8	54	26.0	167.64	M
9	53	32.0	165.10	S
10	63	30.0	170.18	XXXL

## Latihan (11)

Tampilkan data dari "cloth\_data" untuk kolom age dan size

```
In [12]: #Latihan(11)
#Tampilkan data dari "cloth_data" untuk kolom age dan size

df[["age", "size"]]
```

Out[12]:

	age	size
0	28.0	XL
1	36.0	L
2	34.0	M
3	27.0	L
4	45.0	M
5	27.0	S
6	65.0	M
7	33.0	XXS
8	26.0	M
9	32.0	S
10	30.0	XXXL
11	35.0	XXXL
12	26.0	L
13	28.0	M
14	33.0	L
15	30.0	L
16	21.0	XXS
17	27.0	M
18	30.0	M
19	20.0	S
20	32.0	S
21	37.0	M
22	50.0	S
23	43.0	M
24	29.0	XL
25	32.0	XL
26	29.0	XXS
27	47.0	M
28	31.0	XL
29	27.0	L
30	43.0	S
31	27.0	M
32	29.0	M
33	48.0	XXXL
34	36.0	XL
35	27.0	L
36	34.0	L
37	40.0	M
38	45.0	M
39	30.0	S
40	52.0	XXXL
41	37.0	XXXL

	age	size
42	34.0	L
43	24.0	S
44	28.0	M
45	34.0	XXXL
46	25.0	S
47	32.0	M
48	27.0	S
49	29.0	XXXL

## Latihan (12)

Tampilkan data dari "cloth\_data" untuk kolom age dan size, mulai dari indeks 0 (nol) sampai dengan indeks 10 (sepuluh)

```
In [113]: #Latihan(12)
#Tampilkan data dari "cloth_data" untuk kolom age dan size, mulai dari indeks 0 (nol) sampai dengan indeks 10 (sepuluh)
df[["age","size"]].iloc[:11]
```

Out[113]:

	age	size
0	28.0	XL
1	36.0	L
2	34.0	M
3	27.0	L
4	45.0	M
5	27.0	S
6	65.0	M
7	33.0	XXS
8	26.0	M
9	32.0	S
10	30.0	XXXL

## Penggunaan Library Matplotlib

### Latihan (13)

Lakukan import Library Matplotlib

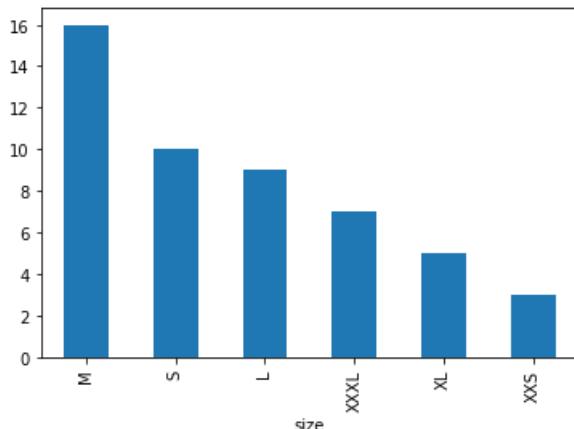
```
In [14]: #Latihan(13)
#Import Library Matplotlib
import matplotlib.pyplot as plt
```

### Latihan (14)

Buatlah visualisasi dalam bentuk histogram pada data "cloth\_data" untuk kolom size

```
In [110]: #Latihan(14)
#Buat histogram untuk kolom size
df.value_counts('size').plot(kind="bar")
```

Out[110]: <AxesSubplot:xlabel='size'>

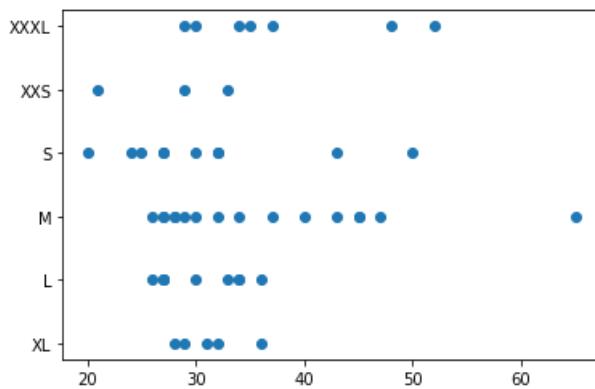


## Latihan (15)

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

```
In [93]: #Latihan(15)
#Visualisasi dalam bentuk grafik (plot) pada data "cloth_data" untuk kolom age dan kolom size
age = df['age']
size = df['size']

plt.plot(age, size, 'o')
plt.show()
```

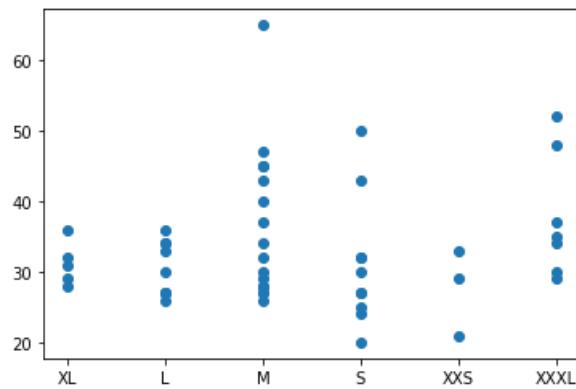


## Latihan (16)

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

```
In [97]: #Latihan(16)
#Visualisasi dalam bentuk grafik (plot) pada data "cloth_data" untuk kolom size dan kolom age
size = df['size']
age = df['age']

plt.plot(size, age, 'o')
plt.show()
```



## Latihan (17)

Import library seaborn

```
In [90]: #Latihan(17)
#Import library seaborn

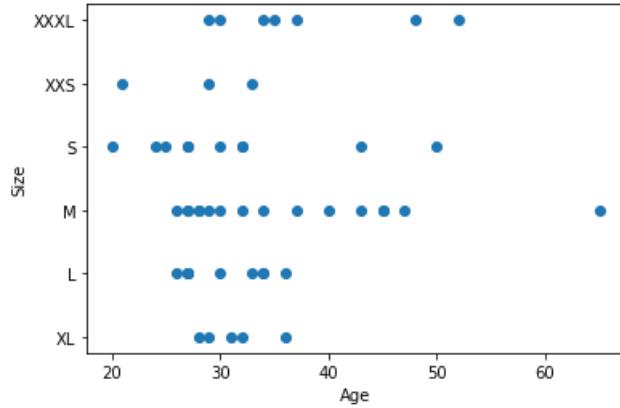
import seaborn as sns
```

## Latihan (18)

Buatlah visualisasi dalam bentuk scatterplot pada data "cloth\_data" untuk kolom age dan kolom size

```
In [107]: #latihan(18)
#Buat scatterplot dengan kolom age dan size
size = df['size']
age = df['age']

plt.xlabel('Age')
plt.ylabel('Size')
plt.scatter(age, size)
plt.show()
```

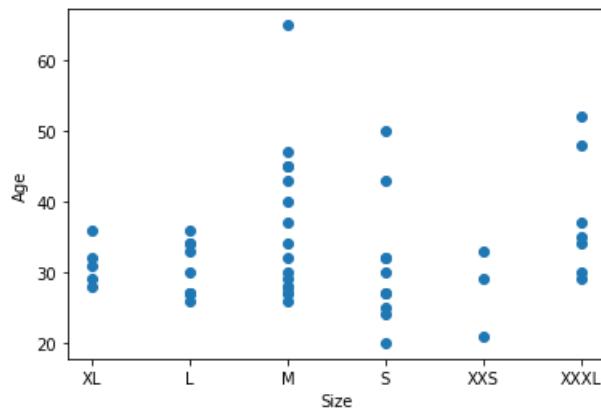


## Latihan (19)

Buatlah visualisasi dalam bentuk scatterplot pada data "cloth\_data" untuk kolom size dan kolom age

```
In [106]: #latihan(19)
#Buat scatterplot dengan kolom size dan age
size = df['size']
age = df['age']

plt.xlabel('Size')
plt.ylabel('Age')
plt.scatter(size, age)
plt.show()
```



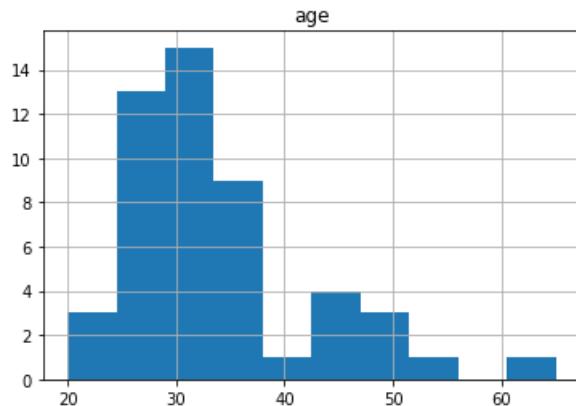
## Latihan (20)

Buatlah visualisasi dalam bentuk histogram pada data "cloth\_data" untuk kolom age

```
In [47]: #Latihan(20)
#Buat histogram untuk kolom age

df.hist('age')
```

```
Out[47]: array([[<AxesSubplot:title={'center':'age'}>]], dtype=object)
```



```
In [ ]:
```