

```
In [1]: #membuat matrik menggunakan Llist
mat = [2.0,5.0,9.0]
mat
```

Out[1]: [2.0, 5.0, 9.0]

```
In [3]: mat = [[2.0,5.0,9.0],[2.0,5.0,9.0]]
mat
```

Out[3]: [[2.0, 5.0, 9.0], [2.0, 5.0, 9.0]]

```
In [5]: mat = [[[2.0,5.0,9.0],[2.0,5.0,9.0]],[[2.0,5.0,9.0],[2.0,5.0,9.0]]]
mat
```

Out[5]: [[[2.0, 5.0, 9.0], [2.0, 5.0, 9.0]], [[2.0, 5.0, 9.0], [2.0, 5.0, 9.0]]]

```
In [7]: mat = [2.0,5.0,9.0]
mat
```

Out[7]: [2.0, 5.0, 9.0]

```
In [9]: mat.append(10)
mat
```

Out[9]: [2.0, 5.0, 9.0, 10]

```
In [11]: mat.insert(3,-7)
mat
```

Out[11]: [2.0, 5.0, 9.0, -7, 10]

```
In [13]: len(mat)
```

Out[13]: 5

```
In [19]: mat = [[2.0,5.0,9.0],[2.0,5.0,9.0]]
mat
```

Out[19]: [[2.0, 5.0, 9.0], [2.0, 5.0, 9.0]]

```
In [21]: len(mat)
```

Out[21]: 2

```
In [23]: mat1 = [2.0,5.0,9.0]
mat2 = [2.2,5.5,9.9]
print("mat1 =",mat1)
print("mat2 =",mat2)
mat1+mat2
```

```
mat1 = [2.0, 5.0, 9.0]
mat2 = [2.2, 5.5, 9.9]
```

Out[23]: [2.0, 5.0, 9.0, 2.2, 5.5, 9.9]

```
In [25]: #membuat matrik menggunakan numpy
import numpy
from numpy import array
mat = array([2.0,5.0,9.0])
mat
```

```
Out[25]: array([2., 5., 9.])
```

```
In [27]: mat = array([[2.0,5.0,9.0],[2.0,5.0,9.0]])
mat
```

```
Out[27]: array([[2., 5., 9.],
               [2., 5., 9.]])
```

```
In [29]: mat = array([[[2.0,5.0,9.0],[2.0,5.0,9.0]],[[2.0,5.0,9.0],[2.0,5.0,9.0]]])
mat
```

```
Out[29]: array([[[2., 5., 9.],
                 [2., 5., 9.]],

               [[2., 5., 9.],
                 [2., 5., 9.]])
```

```
In [31]: mat1 = array([2.0,5.0,9.0])
mat2 = array([2.2,5.5,9.9])
print("mat1 =",mat1)
print("mat2 =",mat2)
mat1+mat2
```

```
mat1 = [2. 5. 9.]
mat2 = [2.2 5.5 9.9]
```

```
Out[31]: array([ 4.2, 10.5, 18.9])
```

```
In [37]: #membuat elemen matrik secara random dengan tipe data float
import numpy as mat
matRandomFloat1D = mat.random.rand(4)
print("matrik 1 dimensi, float")
print(matRandomFloat1D,"\n")

matRandomFloat2D = mat.random.rand(4,5)
print("matrik 2 dimensi, float")
print(matRandomFloat2D,"\n")

matRandomFloat3D = mat.random.rand(3,3,3)
print("matrik 3 dimensi, float")
print(matRandomFloat3D)
```

matrik 1 dimensi, float

```
[0.83922219 0.24629849 0.36091442 0.83700313]
```

matrik 2 dimensi, float

```
[[0.51847189 0.43803574 0.93417199 0.18544302 0.13211388]
 [0.02870569 0.63797534 0.36424615 0.90217699 0.9662448 ]
 [0.09303968 0.90538173 0.39478213 0.36977385 0.4936828 ]
 [0.81823293 0.25873988 0.164014 0.36439603 0.43203408]]
```

matrik 3 dimensi, float

```
[[[0.39853864 0.70289833 0.94747068]
 [0.20672559 0.58068323 0.19998849]
 [0.42573059 0.18407879 0.13395041]]
```

```
[[0.57837077 0.92023834 0.33573958]
 [0.78208583 0.60574657 0.71710215]
 [0.42974944 0.20839953 0.63606668]]
```

```
[[0.25450689 0.23976212 0.17119567]
 [0.56691834 0.76455058 0.99126259]
 [0.10994381 0.55967852 0.33643603]]]
```

```
In [41]: #membuat elemen matrik secara random dengan tipe data integer
matRandomInt1D = mat.random.randint(2,size=7)
print("matrik 1 dimensi, integer")
print(matRandomInt1D,"\n")

matRandomInt2D = mat.random.randint(6,size=(3,4))
print("matrik 2 dimensi, integer")
print(matRandomInt2D,"\n")

matRandomInt3D = mat.random.randint(10,size=(4,4,4))
print("matrik 3 dimensi, integer")
print(matRandomInt3D)
```

```
matrik 1 dimensi, integer
[0 1 1 1 0 1 1]
```

```
matrik 2 dimensi, integer
[[1 1 4 0]
 [4 3 0 3]
 [5 2 5 4]]
```

```
matrik 3 dimensi, integer
[[[4 8 4 4]
   [8 9 8 2]
   [8 2 0 7]
   [5 0 5 3]]
```

```
[[8 2 5 2]
 [4 6 3 5]
 [0 7 1 8]
 [1 7 4 8]]
```

```
[[6 4 6 0]
 [0 5 5 5]
 [2 2 7 8]
 [3 9 3 0]]
```

```
[[4 8 2 6]
 [9 9 6 6]
 [1 6 6 6]
 [3 9 0 2]]]
```

```
In [43]: # matrik nol, bujur sangkar, persegi panjang, diagonal, satuan, skalar
matNol = mat.zeros((2,2))
print("matrik nol =")
print(matNol, "\n")

matPersegi = mat.random.randint(10, size=(5,5))
print("matrik bujur sangkar =")
print(matPersegi, "\n")

matPP = mat.random.randint(10, size=(2,3))
print("matrik persegi panjang =")
print(matPP, "\n")

matDiagonal = mat.diag([1,2,3])
print("matrik diagonal =")
print(matDiagonal, "\n")

matIdentitas = mat.eye(3)
print("matrik satuan =")
print(matIdentitas, "\n")

matSkalar = 4*mat.eye(5)
print("matrik skalar =")
print(matSkalar)
```

```

matrik nol =
[[0. 0.]
 [0. 0.]]

matrik bujur sangkar =
[[2 6 6 8 6]
 [2 1 7 2 3]
 [5 8 3 5 5]
 [6 1 7 8 7]
 [5 4 0 9 8]]

matrik persegi panjang =
[[9 0 2]
 [5 6 7]]

matrik diagonal =
[[1 0 0]
 [0 2 0]
 [0 0 3]]

matrik satuan =
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

matrik skalar =
[[4. 0. 0. 0. 0.]
 [0. 4. 0. 0. 0.]
 [0. 0. 4. 0. 0.]
 [0. 0. 0. 4. 0.]
 [0. 0. 0. 0. 4.]]

```

```

In [53]: # operasi aljabar pada matrik
k = 2
print("k =",k)
A = mat.array([[2,4],[3,7]])
print("matrik A =")
print(A)
B = mat.array([[0,1],[2,3]])
print("matrik B =")
print(B,"\n")

print("perkalian skalar =")
matPerkalianSkalar = k*A
print("k*A =")
print(matPerkalianSkalar,"\n")

print("penjumlahan =")
plus = A+B
print("A+B =")
print(plus,"\n")

print("pengurangan =")
minus = A-B
print("A-B =")
print(minus,"\n")

print("perkalian =")
times = A@B

```

```
print("AxB =")
print(times)
```

```
k = 2
matrik A =
[[2 4]
 [3 7]]
matrik B =
[[0 1]
 [2 3]]
```

```
perkalian skalar =
k*A =
[[ 4  8]
 [ 6 14]]
```

```
penjumlahan =
A+B =
[[ 2  5]
 [ 5 10]]
```

```
pengurangan =
A-B =
[[2 3]
 [1 4]]
```

```
perkalian =
AxB =
[[ 8 14]
 [14 24]]
```

```
In [55]: # sifat matrik
print("matrik A =")
print(A)
print("matrik B =")
print(B)
C = mat.array([[5,6],[8,9]])
print("matrik C =")
print(C)
print("k =",k)
l = 3
print("l =",l,"\n")

SM1 = mat.array_equal(A+B,B+A)
print("A+B=B+A?", SM1)

SM2 = mat.array_equal(A+(B+C),(A+B)+C)
print("A+(B+C)=(A+B)+C?", SM2)

SM3 = mat.array_equal(k*(A+B),(k*A)+(k*B))
print("k(A+B)=kA+kB?", SM3)

SM4 = mat.array_equal((k+1)*A,(k*A)+(1*A))
print("(k+1)A=kA+1A?", SM4)

SM5 = mat.array_equal((k*1)*A,k*(1*A))
print("(k1)A=k(1A)?", SM5)

SM6_1 = mat.array_equal(k*(A@B),(k*A)@B)
SM6_2 = mat.array_equal((k*A)@B,A@(k*B))
```

```

SM6_3 = mat.array_equal(SM6_1,SM6_2)
print("k(AB)=kA(B)=A(kB)?",SM6_3)

SM7 = mat.array_equal(A@(B@C),(A@B)@C)
print("A(BC)=(AB)C?",SM7)

SM8 = mat.array_equal(A@(B+C),(A@B)+(A@C))
print("A(B+C)=AB+AC?",SM8)

SM9 = mat.array_equal((A+B)@C,(A@C)+(B@C))
print("(A+B)C=AC+BC?",SM9)

SM10 = mat.array_equal(A@B,B@A)
print("AB=BA?",SM10,"\n")

print("Jika AB=AC, belum tentu B=C,")
A = mat.zeros((2,2))
print("matrik A =")
print(A)
SM11_1 = mat.array_equal(A@B,A@C)
print("AB=AC?",SM11_1)
SM11_2 = mat.array_equal(B,C)
print("B=C?",SM11_2,"\n")

print("Jika AB=0, ada 2 kemungkinan :")
nol = mat.zeros((2,2))
SM12_1 = mat.array_equal(A@B,nol)
print(" - A=0 dan/atau B=0",(" ",SM12_1,""))
A = mat.array([[1,1],[2,2]])
print("matrik A =")
print(A)
B = mat.array([1,-1])
print("matrik B =")
print(B)
nol = mat.zeros(2)
SM12_2 = mat.array_equal(A@B,nol)
print(" - A!=0 dan B!=0",(" ",SM12_2,""))

```

```

matrik A =
[[2 4]
 [3 7]]
matrik B =
[[0 1]
 [2 3]]
matrik C =
[[5 6]
 [8 9]]
k = 2
l = 3

```

```

A+B=B+A? True
A+(B+C)=(A+B)+C? True
k(A+B)=kA+kB? True
(k+1)A=kA+lA? True
(kl)A=k(lA)? True
k(AB)=kA(B)=A(kB)? True
A(BC)=(AB)C? True
A(B+C)=AB+AC? True
(A+B)C=AC+BC? True
AB=BA? False

```

```

Jika AB=AC, belum tentu B=C,
matrik A =
[[0. 0.]
 [0. 0.]]
AB=AC? True
B=C? False

```

```

Jika AB=0, ada 2 kemungkinan :
- A=0 dan/atau B=0 ( True )
matrik A =
[[1 1]
 [2 2]]
matrik B =
[ 1 -1]
- A!=0 dan B!=0 ( True )

```

In []: