Numpy

1. Numpy Array (Vector & Matrix)

In [2]:

```
import numpy as np

# vector
a = np.array([1,2,3,4,5])
b = np.array([1.5,2.5,3.5,4.5,5.5])
print(a)
print(b)
```

```
[1 2 3 4 5]
[1.5 2.5 3.5 4.5 5.5]
```

In [3]:

```
# arange
c = np.arange(1,10,1)
print(c)
```

```
[1 2 3 4 5 6 7 8 9]
```

In [4]:

```
# linspace
d = np.linspace(1,10,4)
print(d)
```

```
[ 1.  4.  7. 10.]
```

In [5]:

```
# matrix
e = np.array([ [1,2,3] , [4,5,6] ])
print(e)
print(e+1)
```

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

In [6]:

```
# zeros
f = np.zeros([5,5])
print(f)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

In [7]:

```
# ones
g = np.ones([5,5])
print(g)
```

```
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
```

In [8]:

```
# identity
h = np.identity(5)
print(h)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

2. Arithmetic Operations

In [9]:

```
a = np.array([1,2,3])
b = np.array([4,5,6])

c = a+b
print(c)
d = a-b
print(d)
e = a*b
print(e)
f = a/b
print(f)
g = a**2
print(g)
```

```
[5 7 9]
[-3 -3 -3]
[ 4 10 18]
[0.25 0.4  0.5 ]
[1 4 9]
```

In [10]:

```
a = np.array([[1,2,3],[4,5,6]])
b = np.array([[4,5,6],[1,2,3]])

print(a)
print(b)

c = a+b
print(c)
d = a-b
print(d)
e = a*b
print(e)
f = a/b
print(f)
g = a**2
print(g)
```

```
[[1 2 3]
 [4 5 6]]
[[4 5 6]
 [1 2 3]]
[[5 7 9]
 [5 7 9]]
[[-3 -3 -3]
 [ 3  3  3]]
[[ 4 10 18]
```

```
 [ 4 10 18]]
[[0.25 0.4  0.5 ]
 [4.   2.5  2.  ]]
[[ 1  4  9]
 [16 25 36]]
```

In [11]:

```
a = np.array([[1,2],[3,4]])
b = np.array([[5,6],[7,8]])

print("Matrix a :\n",a)
print("Matrix b :\n",b)

c = np.dot(a,b)
print("Perkalian Matrix :\n",c)

d = a.dot(b)
print("Perkalian Matrix :\n",d)
```

```
Matrix a :
 [[1 2]
 [3 4]]
Matrix b :
 [[5 6]
 [7 8]]
Perkalian Matrix :
 [[19 22]
 [43 50]]
Perkalian Matrix :
 [[19 22]
 [43 50]]
```

3. Access Items

In [60]:

```
a = np.arange(10)**2
print(a)

print("Element ke 1 : ",a[0])
print("Element terakhir : ",a[-1])
print("Element ke 1 sampai 6 : ",a[0:6])
print("Element ke 4 sampai akhir : ",a[4:])
print("Element ke awal sampai 5 : ",a[:5])
```

```
[ 0  1  4  9 16 25 36 49 64 81]
Element ke 1 :  0
Element terakhir :  81
Element ke 1 sampai 6 :  [ 0  1  4  9 16 25]
Element ke 4 sampai akhir :  [16 25 36 49 64 81]
Element ke awal sampai 5 :  [ 0  1  4  9 16]
```

In [13]:

```
for i in a:
    print('value : ',i)
```

```
value :  0
value :  1
value :  4
value :  9
value :  16
value :  25
value :  36
value :  49
value :  64
value :  81
```

In [34]:

```
a = np.array([[1,2,3],[4,5,6]])
print(a)
print(a[0,1])
print(a[1,1:3])
print(a[1,:])
```

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

4. Manipulation

In [14]:

```
a = np.array([[1,2,3],[4,5,6]])

print("Matrix a ukuran: ",a.shape)
print(a)

# transpose
print("Transpose Matrix a:")
print(a.transpose())
print(np.transpose(a))
print(a.T)
```

```
Matrix a ukuran:  (2, 3)
[[1 2 3]
 [4 5 6]]
Transpose Matrix a:
[[1 4]
 [2 5]
 [3 6]]
[[1 4]
 [2 5]
 [3 6]]
[[1 4]
 [2 5]
 [3 6]]
```

In [15]:

```
# flatten
print(a.ravel())
print(np.ravel(a))
```

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

In [35]:

```
# reshape
print(a.reshape(3,2))
print(a.reshape(3,-1))
```

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

In [17]:

```
# resize
a.resize(3,2)
```

```
print(a)
```

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

In [40]:

```
print(a)
a_del_col = np.delete(a,1,1)
print(a_del_col)
a_del_row = np.delete(a,1,0)
print(a_del_row)
```

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

5. Stacking

In [41]:

```
a = np.array([1,2,3])
b = np.array([4,5,6])

c = np.hstack([a,b])
print(c)
d = np.vstack([a,b])
print(d)
```

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

6. Sorting

In [19]:

```
a = np.random.randn(1,6)*10
b = np.floor(a)

print(b)
```

```
[[-10. -12.  -5.  -5.  19.  25.]]
```

In [20]:

```
print("Nilai Max : ",b.max())
print("Index Nilai Max : ",b.argmax())

print("Nilai Min : ",b.min())
print("Index Nilai Min : ",b.argmin())
```

```
Nilai Max :  25.0
Index Nilai Max :  5
Nilai Min :  -12.0
Index Nilai Min :  1
```

In [21]:

```
# sort
print(np.sort(b))
print(np.argsort(b))
```

```
print(-np.sort(-b))
```

```
[[-12. -10.  -5.  -5.  19.  25.]]
[[1 0 2 3 4 5]]
[[ 25.  19.  -5.  -5. -10. -12.]]
```

In [58]:

```
a = np.array([[4,5,6],[1,2,3],[3,6,1]])
print(a)
print(np.argsort(a[:,1]))
print(a[np.argsort(a[:,1])])
```

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

In [53]:

```
dtipe = [('nama','S50'),('tinggi',int)]
data = [('Adi',170),('Budi',145),('Cici',160)]
a = np.array(data, dtype = dtipe)

print(a)
print(np.sort(a, order = 'tinggi'))
print(np.sort(a, order = 'nama'))

print(np.sort(a, order = 'tinggi')[::-1])
```

```
[(b'Adi', 170) (b'Budi', 145) (b'Cici', 160)]
[(b'Budi', 145) (b'Cici', 160) (b'Adi', 170)]
[(b'Adi', 170) (b'Budi', 145) (b'Cici', 160)]
[(b'Adi', 170) (b'Cici', 160) (b'Budi', 145)]
```

7. Invers & Determinan

In [27]:

```
a = np.array([[1,-1],[1,1]])
print(a)

# invers
inv_a = np.linalg.inv(a)
print(inv_a)

print(np.dot(a,inv_a))

# determinan
det_a = np.linalg.det(a)
print(det_a)
```

```
[[ 1 -1]
 [ 1  1]]
[[ 0.5  0.5]
 [-0.5  0.5]]
[[1. 0.]
 [0. 1.]]
2.0
```

8. Median, Mean, Standar Deviasi & Varian

In [65]:

```
a = np.array([[1,2,3],[4,5,6]])

# median
std_a = np.mean(a)
print(std_a)

# mean
std_a = np.mean(a)
print(std_a)

# standar deviasi
std_a = np.std(a)
print(std_a)

# varian
var_a = np.var(a)
print(var_a)
```

```
3.5
3.5
1.707825127659933
2.9166666666666665
```

9. Read File

In [ ]:

```
import numpy as np

file = 'data.txt'
data = np.loadtxt(file, delimiter=',')
print(data)
data = np.loadtxt(file, delimiter=',', skiprows=1)
print(data)
```