

**LAPORAN KECERDASAN BUATAN
UJIAN TENGAH SEMESTER**



Disusun oleh :

Diah Ayuning Tyas

21091397013

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA
2022**

- **Single Neuron**

Source Code

```
1 #Diah Ayuning Tyas_A_21091397013
2 #Membuat Program Single Neuron
3
4 #Inisialisasi Numpy
5 import numpy as np
6
7 #Inisialisasi variabel dengan panjang input layer = 10
8 inputs = [4,6,5.5,3,4.5,2,7,9,8,1]
9
10 #Inisialisasi bobot variabel sesuai dengan jumlah neuron = 1
11 weights = [0.5,1.6,-2.7,0.8,-4.3,-0.9,2.5,1.9,-2.4,1.7]
12
13 #Inisialisasi bias berdasarkan panjang neuron
14 bias = 8
15
16 # Menghitung output = (input*weight)+bias
17 output = np.dot(weights, inputs) + bias
18
19 # cetak output
20 print(output)
```

Output

```
PS C:\Users\user\Downloads> c::; cd 'c:\Users\user\Downloads'; & 'C:\Users\user\AppData\Local\Microsoft\Windows
Apps\python3.10.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugg
y\adapter\..\..\debugpy\launcher' '50007' '--' 'c:\Users\user\Downloads\013_Diah Ayuning Tyas_Single Neuron (2)
.py'
3.1000000000000023
```

Langkah Mengerjakan

1.

```
#Inisialisasi Numpy
import numpy as np
```

Sebelum menggunakan numPy pada python, kita harus mengimport library tersebut.
2.

```
#Inisialisasi variabel dengan panjang input layer = 10
inputs = [4,6,5.5,3,4.5,2,7,9,8,1]
```

Memasukkan input sesuai dengan panjang input layer yaitu 10
3.

```
#Inisialisasi bobot variabel sesuai dengan jumlah neuron = 1
weights = [0.5,1.6,-2.7,0.8,-4.3,-0.9,2.5,1.9,-2.4,1.7]
```

Memasukkan weights sesuai dengan input layer
4.

```
#Inisialisasi bias berdasarkan panjang neuron
bias = 8
```

Memasukkan bias sesuai dengan jumlah neuron yaitu 1
5.

```
15
16 # Menghitung output = (input*weight)+bias
17 output = np.dot(weights, inputs) + bias
18
```

Menghitung output dari hasil perkalian input dan weight dan ditambah bias

```
# cetak output
print(output)
```

6. Mencetak hasil output dengan perintah “Print”

- **Multi Neuron**

Source Kode

```
1 #Diah Ayuning Tyas_A_21091397013
2 #Membuat Program Multi Neuron
3
4 #inisialisasi numpy
5 import numpy as np
6
7 #inisialisasi variabel dengan input sesuai panjang input layer = 10
8 inputs = [2.0,1.5,2.2,3.4,1.0,3.0,4.5,5.0,2.4,7.0]
9
10 #inisialisasi bobot variabel dengan weight sesuai jumlah neuron = 5
11 weights = [[5.5,0.8,-1.2,3.5,4.8,-1.5,-2.2,0.4,5.5],
12            [3.6,0.7,-2.4,1.0,5.2,5.0,4,-0.6,9.0],
13            [2,1.5,2.0,-0.5,0.2,-2.2,1.0,7.5,5.0,5],
14            [1.2,4.8,1,3,0.2,8.0,7.5,-0.3,4.5],
15            [3.3,1.5,-4.2,3.0,5,7,0.1,2.4,1.5,6.5]]
16
17 #jumlah bias seperti jumlah neuron
18 bias = [1.2,0.4,5,2.5,0.2]
19
20 #menghitung output = (input*weight)+bias
21 output = np.dot(weights, inputs) + bias
22
23 #mencetak output
24 print(output)
```

Output

```
PS C:\Users\user\Downloads> cd 'c:\Users\user\Downloads'; & 'C:\Users\user\AppData\Local\Microsoft\Windows
Apps\python3.10.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugp
y\adapter\..\..\debugpy\launcher' '50023' '---' 'c:\Users\user\Downloads\013_Diah Ayuning Tyas_Multi Neuron (2).
py'
[ 72.37  99.59  63.75 139.78  97.56]
PS C:\Users\user\Downloads>
```

Langkah Mengerjakan

```
#inisialisasi Numpy
import numpy as np
```

1. Sebelum menggunakan numPy pada python, kita harus mengimport library tersebut.

```
#inisialisasi variabel dengan input sesuai panjang input layer = 10
inputs = [2.0,1.5,2.2,3.4,1.0,3.0,4.5,5.0,2.4,7.0]
```

2. Menginisialisasi variabel sesuai dengan panjang input layer = 10

```
#inisialisasi bobot variabel dengan weight sesuai jumlah neuron = 5
weights = [[5.5,0.8,-1.2,3.5,4.8,-1.5,-2.2,0.4,5.5],
           [3.6,0.7,-2.4,1.0,5.2,5.0,4,-0.6,9.0],
           [2,1.5,2.0,-0.5,0.2,-2.2,1.0,7.5,5.0,5],
           [1.2,4.8,1,3,0.2,8.0,7.5,-0.3,4.5],
           [3.3,1.5,-4.2,3.0,5,7,0.1,2.4,1.5,6.5]]
```

3. Memasukkan weights sesuai dengan jumlah neuron

```
#jumlah bias seperti jumlah neuron
bias = [1.2,0.4,5,2.5,0.2]
```

4. Memasukkan bias sesuai dengan jumlah neuron yaitu 5

```
5. #menghitung output = (input*weight)+bias
    output = np.dot(weights, inputs) + bias
```

Menghitung output dari hasil perkalian input dan weight dan ditambah bias

```
6. #mencetak output
    print(output)
```

Mencetak hasil output dengan perintah “Print”

- **Multi Neuron Batch Input**

Source Kode

```
#Diah Ayuning Tyas_A_21091397013
#Membuat Program Multi Neuron Batch Input

# inisialisasi numpy
import numpy as np

#inisialisasi variabel matriks 6x10 (jumlah batch = 6 dan input layer = 10 )
inputs = [[1.2,2.5,3.0,1.0,2.3,1.5,5.0,8.0,3.5,6.0],
          [1.2,4.2,5.5,3.5,8.0,6.5,2.5,1.0,4.5,5.0],
          [2.0,5.0,2.5,1.5,4.0,3.5,5.5,9.0,6.0,7.0],
          [3.3,5.0,2.4,1.2,5.0,3.2,7.5,6.0,3.0,2.5],
          [1.2,2.5,3.5,7.0,2.0,1.5,5.0,5.5,8.0,6.5],
          [3.5,2.5,1.0,2.2,4.0,2.0,8.0,5.0,3.5,7.5]]

#inisialisasi bobot variabel (panjang weight = panjang input dan jumlah weight = jumlah neuron)
weights = [[1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0],
           [1.2,2.2,4.2,3.4,5.0,1.0,2.5,5.2,4.7,8.0],
           [4.0,2.5,2.2,3.5,1.5,6.0,2.3,1.5,8.0,9.0],
           [3.0,5.5,2.5,4.2,1.0,2.6,7.5,9.0,2.0,6.7],
           [1.2,3.0,3.2,4.5,7.0,6.5,3.0,2.5,8.5,6.5]]

#inisialisasi bias (jumlah bias = jumlah neuron)
bias = [1.2,2.0,2.5,3.2,3.0]

#Menghitung output menggunakan rumus :
output = np.dot(inputs, np.array(weights).T) + bias

#Mencetak Output
print(output)
```

Output

```
y:\adapter\...\debugpy\launcher' '58018' '-c' 'c:\Users\user\Downloads\013_Diah Ayuning Tyas_Multi Neuron batc
h input (1).py'
[[231.4 156.49 141.6 195.15 155.64]
 [236.3 166.78 181.4 153.5 229.39]
 [302.2 199.25 197.9 243.5 221.4 ]
 [223.2 143.37 137.13 197.96 170.09]
 [280.9 189.64 200. 211.15 217.39]
 [256. 167.83 172.05 210.64 183.8 ]]
```

Langkah Mengerjakan

```
1. # inisialisasi numpy
    import numpy as np
```

Sebelum menggunakan numPy pada python, kita harus mengimport library tersebut.

```
2. #inisialisasi variabel matriks 6x10 (jumlah batch = 6 dan input layer = 10 )
    inputs = [[1.2,2.5,3.0,1.0,2.3,1.5,5.0,8.0,3.5,6.0],
              [1.2,4.2,5.5,3.5,8.0,6.5,2.5,1.0,4.5,5.0],
              [2.0,5.0,2.5,1.5,4.0,3.5,5.5,9.0,6.0,7.0],
              [3.3,5.0,2.4,1.2,5.0,3.2,7.5,6.0,3.0,2.5],
              [1.2,2.5,3.5,7.0,2.0,1.5,5.0,5.5,8.0,6.5],
              [3.5,2.5,1.0,2.2,4.0,2.0,8.0,5.0,3.5,7.5]]
```

Menginisialisasi variabel dengan jumlah batch = 6 dan input layer = 10

```
#inisialisasi bobot variabel (panjang weight = panjang input dan jumlah weight = jumlah neuron)
weights = [[1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0],
            [1.2,2.2,4.2,3.4,5.0,1.0,2.5,5.2,4.7,8.0],
            [4.0,2.5,2.2,3.5,1.5,6.0,2.3,1.5,8.0,9.0],
            [3.0,5.5,2.5,4.2,1.0,2.6,7.5,9.0,2.0,6.7],
            [1.2,3.0,3.2,4.5,7.0,6.5,3.0,2.5,8.5,6.5]]
```

3. Memasukkan weights sesuai dengan jumlah neuron

```
#inisialisasi bias (jumlah bias = jumlah neuron)
bias = [1.2,2.0,2.5,3.2,3.0]
```

4. Memasukkan bias sesuai dengan jumlah neuron yaitu 5

```
#Menghitung output menggunakan rumus :
output = np.dot(inputs, np.array(weights).T) + bias
```

5. Menghitung output dengan perkalian antara inputs dan weight lalu dikalikan dengan tranpose dan ditambahkan dengan bias

```
#Mencetak Output
print(output)
```

6. Mencetak hasil output dengan perintah “Print”