

# **LAPORAN UTS KECERDASAN BUATAN**



Disusun oleh :

Rahmad firdiansyah (21091397023)

---

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2022**

## No. 1a : Single Neuron

### Source code :

```
Al.py ● No 1a Numpy.py × No 1b Numpy.py No 1c Numpy.py ● Org.jav
No 1a Numpy.py > ...
1  #menggunakan fungsi numpy
2  import numpy as np
3
4  #input data
5  inputs = [1.6, 1.5, 2.0, 2.4, 1.5, 1.0, 4.0, 3.5, 1.3, 2.0]
6  weights = [[0.2, 0.9, 0.5, 0.1, 0.2, 0.3, 0.5, 0.5, 0.6, 0.7],]
7  biases = [2.0, 2.0, 0.5, 1.5, 2.4]
8  #perkalian Weight dan inputs menggunakan dot lalu ditambah oleh biases
9  hasil = np.dot(weights, inputs) + biases
10 #print output
11 print(hasil)
```

### Output :

```
[11.44  9.94 10.94 11.84]
PS C:\Users\LENOVO\OneDrive\Desktop\vcode> & C:/Users/LENOVO/AppData/Local/Microsoft/Windows
LENOVO/OneDrive/Desktop/vcode/No 1a Numpy.py"
[11.44 11.44  9.94 10.94 11.84]
PS C:\Users\LENOVO\OneDrive\Desktop\vcode> █
```

### Jelaskan cara kerja :

1.) Menggunakan fungsi numpy agar python mendukung penggunaan array

```
Al.py ● No 1a Numpy.py × No 1b Numpy.py No 1c Numpy.py ● Org.jav
No 1a Numpy.py > ...
1  #menggunakan fungsi numpy
2  import numpy as np
3
```

2.) inputkan data

```
4  #input data
5  inputs = [1.6, 1.5, 2.0, 2.4, 1.5, 1.0, 4.0, 3.5, 1.3, 2.0]
6  weights = [[0.2, 0.9, 0.5, 0.1, 0.2, 0.3, 0.5, 0.5, 0.6, 0.7],]
7  biases = [2.0, 2.0, 0.5, 1.5, 2.4]
```

3.) lalu gunakan fungsi np.dot untuk mengalikan weight dan input lalu tambahkan dengan biases

```
8  #perkalian Weight dan inputs menggunakan dot lalu ditambah oleh biases
9  hasil = np.dot(weights, inputs) + biases
```

4.) gunakan fungsi print untuk menampilkan hasil

```
10 #print output
11 print(hasil)
```

## No. 1b : Multi Neuron

### Source code :

```
Al.py • No 1a Numpy.py No 1b Numpy.py X No 1c Numpy.py • Org.jav
No 1b Numpy.py > ...
1 #menggunakan fungsi numpy
2 import numpy as np
3
4 #input data
5 inputs = [2.3, 2.5, 2.0, 2.5, 2.5, 2.0, 3.0, 3.5, 2.3, 3.0]
6 weights = [[0.2, 0.8, 0.5, 0.2, 0.2, 0.3, 0.3, 0.5, 0.6, 0.7],
7            [0.5, 0.2, 0.26, 0.5, 0.22, 0.22, 0.23, 0.23, 0.25, 0.26],
8            [0.26, 0.27, 0.27, 0.87, 0.22, 0.28, 0.25, 0.28, 0.22, 0.23],
9            [0.23, 0.25, 0.26, 0.27, 0.28, 0.28, 0.30, 0.30, 0.32, 0.32],
10           [0.32, 0.32, 0.33, 0.33, 0.35, 0.36, 0.37, 0.38, 0.38, 0.33]]
11 biases = [2.0, 3.0, 0.5, 2.5, 2.5]
12 #menggunakan perkalian dot
13 hasil = np.dot(weights, inputs) + biases
14 #hasil output
15 print(hasil)
```

### Output :

```
PS C:\Users\LENOVO\OneDrive\Desktop\vcde> & C:/Users/LENOVO/AppData/Local/Microsoft/WindowsApps/
LENOVO/OneDrive/Desktop/vcode/No 1b Numpy.py"
[13.19 10.26 8.524 9.755 11.42 ]
PS C:\Users\LENOVO\OneDrive\Desktop\vcde> □
```

### Jelaskan cara kerja :

- 2.) Menggunakan fungsi numpy agar python mendukung penggunaan array
- 5-11.) inputkan data
- 13.) lalu gunakan fungsi np.dot untuk mengalikan weight dan input lalu tambahkan dengan biases
- 15.) gunakan fungsi print untuk menampilkan hasil

## No. 1c : Multi Neuron Batch Input

### Source kode :

```
Al.py • No 1a Numpy.py No 1b Numpy.py No 1c Numpy.py • Org.jav
No 1c Numpy.py > ...
1  #menggunakan numpy
2  import numpy as np
3  #semua input
4  inputs = [[1.1, 0.8, 0.4, 0.1, 0.1, 0.4],
5            [0.4, 0.81, 0.16, 0.4, 0.13, 0.14],
6            [0.16, 0.17, 0.17, 0.87, 0.77, 0.11],
7            [0.31, 0.33, 0.34, 0.34, 0.37, 0.38],
8            [0.11, 0.11, 0.13, 0.14, 0.14, 0.16],
9            [0.1, 0.8, 0.4, 0.1, 0.1, 0.4],
10           [0.4, 0.81, 0.16, 0.4, 0.13, 0.14],
11           [0.16, 0.17, 0.17, 0.87, 0.77, 0.11],
12           [0.31, 0.33, 0.34, 0.34, 0.37, 0.38],
13           [0.11, 0.11, 0.13, 0.14, 0.14, 0.16]]
14
15  weights = [[0.1, 0.8, 0.4, 0.1, 0.1, 0.4],
16             [0.4, 0.81, 0.16, 0.4, 0.13, 0.14],
17             [0.16, 0.17, 0.17, 0.87, 0.77, 0.11],
18             [0.31, 0.33, 0.34, 0.34, 0.37, 0.38],
19             [0.11, 0.11, 0.13, 0.14, 0.14, 0.16]]
20
21  biases = [1.0, 3.0, 0.4, 1.4, 1.0]
22  #kalikan
23  hasil = np.dot(inputs, np.array(weights)) + biases
24  #print hasil
25  print(hasil)
```

## Output :

```
PS C:\Users\LENOVO\OneDrive\Desktop\vcod> & C:/Users/LENOVO/AppData/Local/Microsoft/WindowsApps/python3.10.11/Python.exe C:\Users\LENOVO\OneDrive\Desktop\vcod\No 1c Numpy.py"
[[3.34  4.631  1.408  2.565  2.72 ]
 [3.131  4.4351 1.4715 2.3238 2.5565]
 [2.648  3.9715 2.0313 2.3587 2.583 ]
 [2.755  3.8238 1.3587 2.2244 2.4912]
 [2.51   3.5565 1.083  1.9912 2.3331]
 [3.14   4.131  1.148  2.255  2.51 ]
 [3.131  4.4351 1.4715 2.3238 2.5565]
 [2.648  3.9715 2.0313 2.3587 2.583 ]
 [2.755  3.8238 1.3587 2.2244 2.4912]
 [2.51   3.5565 1.083  1.9912 2.3331]]
PS C:\Users\LENOVO\OneDrive\Desktop\vcod> █
```

## Jelaskan cara kerja :

- 2.) Menggunakan fungsi numpy agar python mendukung penggunaan array
- 4-21.) inputkan data input, weight, dan biases
- 23.) lalu gunakan fungsi np.dot untuk mengalikan weight dan input lalu tambahkan dengan biases
- 25.) gunakan fungsi print untuk menampilkan hasil