

## **LAPORAN UTS AI**



Disusun oleh :

Filusive Nathan Fernanda (21091397073)

---

---

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

## No. 1a : Single Neuron

### Source Code :

```
1 #UTS AI
2 #Filusive Nathan Fernanda 21091397073
3
4 #Single Neuron
5
6 #Inisialisasi Numpy
7 import numpy as np
8
9 #Inisialisasi Variabel
10 inputs = [2.0, 1.5, 1.0, 3.5, 3.0, 1.5, 4.0, 2.5, 3.0, 7.2]
11 weights = [1.0, 3.0, 4.0, 2.5, 1.0, 1.2, 4.0, 3.2, 2.0, 2.0]
12 bias = 5
13
14 #Output
15 outputs = np.dot(weights, inputs) + bias
16
17 #PrintOutput
18 print(outputs)
```

### Output :

73.45

### Jelaskan cara kerja :

- Menggunakan fungsi numpy agar python mendukung penggunaan array
- Inputkan data input, weight, dan biases
- Gunakan fungsi np.dot untuk mengalikan weight dan input lalu tambahkan dengan biases
- Gunakan fungsi print untuk menampilkan hasil

## No. 1b : Multi Neuron

### Source Code :

```
2  #Filusive Nathan Fernanda 21091397073
3
4  #Multi Neuron
5
6  #Inisialisasi Numpy
7  import numpy as np
8
9  #Inisialisasi Variabel
10 inputs = [4.8, 6.2, 1.1, 2.6, 3.1, 1.0, 2.5, 1.7, 3.3, 3.0]
11 weights = [
12     [2.3, 4.3, 7.4, 5.3, 9.5, 3.2, 1.4, 5.0, 2.3, 4.0],
13     [2.4, 5.2, 4.4, 2.0, 2.6, 7.0, 5.5, 1.5, 9.0, 4.5],
14     [2.0, 1.4, 4.5, 2.0, 5.0, 2.5, 4.5, 3.0, 2.5, 8.5],
15     [2.0, 5.2, 3.0, 1.0, 2.0, 4.0, 2.0, 4.5, 1.5, 6.5],
16     [4.3, 5.0, 4.2, 2.6, 3.7, 4.8, 3.0, 4.0, 3.0, 6.0],
17 ]
18 biases = [9.0, 1.0, 1.5, 4.4, 1.5]
19
20 #Output
21 outputs = np.dot(weights, inputs) + biases
```

### Output :

```
[132.86 129.36 98.03 99.44 122.99]
```

### Jelaskan cara kerja :

- Menggunakan fungsi numpy agar python mendukung penggunaan array
- Inputkan data input, weight, dan biases
- Gunakan fungsi np.dot untuk mengalikan weight dan input lalu tambahkan dengan biases
- Gunakan fungsi print untuk menampilkan hasil

### No. 1c : Multi Neuron Batch Input

#### Source Code :

```
7 import numpy as np
8
9 #Inisialisasi Variabel
10 inputs = [
11     [1.5, 3.5, 2.0, 7.1, 9.4, 2.8, 1.7, 3.2, 4.6, 2.5],
12     [3.2, 2.2, 2.1, 1.1, 2.3, 7.2, 1.5, 2.5, 3.0, 5.0],
13     [2.5, 9.3, 2.0, 10.0, 2.5, 19.5, 30.2, 10.5, 47.5, 30.0],
14     [3.8, 5.7, 4.3, 5.5, 1.5, 7.8, 6.1, 2.0, 6.0, 5.7],
15     [4.4, 8.3, 2.7, 6.3, 8.8, 9.1, 3.5, 5.6, 12.0, 9.5],
16     [10.0, 13.1, 19.4, 11.0, 20.0, 14.1, 33.2, 21.3, 10.0, 14
17         .0],
18 ]
19 weights = [
20     [3.2, 2.1, 3.2, 4.1, 1.0, 5.4, 1.0, 3.0, 5.1, 9.0],
21     [3.6, 3.7, 8.3, 1.0, 2.3, 5.5, 6.7, 3.2, 4.9, 7.0],
22     [7.2, 6.9, 4.3, 2.1, 1.0, 5.6, 7.5, 5.9, 1.9, 5.0],
23     [2.7, 7.9, 4.1, 9.8, 5.7, 7.1, 8.0, 9.0, 12.4, 10.1],
24     [2.5, 84.3, 11.5, 13.4, 23.4, 41.0, 34.1, 32.5, 25.0, 22.2]
25 ]
26 biases = [11.0, 2.5, 1.0, 2.0, 3.7]
```

#### Output :

```
[[ 140.44  143.24  137.41  309.63 1087.87]
 [ 147.57  153.33  149.88  233.84  903.47]
 [ 767.68  864.2   753.17 1569.22 5033.51]
 [ 209.06  241.38  240.36  391.52 1521.31]
 [ 301.92  314.72  304.14  590.18 2221.13]
 [ 547.93  820.14  832.54 1256.64 4934.8 ]]
```

#### Jelaskan cara kerja :

- Menggunakan fungsi numpy agar python mendukung penggunaan array
- Inputkan data input, weight, dan biases

- Gunakan fungsi np.dot untuk mengalikan weight dan input lalu tambahkan dengan biases
- Gunakan fungsi print untuk menampilkan hasil