

LAPORAN UTS KECERDASAN BUATAN 2



Disusun oleh :

Rahmad firdiansyah (21091397023)

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

No. 1a2 : Multi Neuron Batch Input

Source code :

```
Al.py No 1a Numpy2.py x ... No 1a Numpy2.py x
No 1a Numpy2.py > ...
1  #menggunakan fungsi numpy
2  import numpy as np
3
4  #input data
5  inputs = [ [1.3, 3.3, 3.4, 5.5, 5.7, 7.9, 1.4, 3.9, 3.5, 4.9],
6             [0.5, 0.5, 1.3, 3.5, 3.5, 4.5, 1.5, 3.9, 5.9, 5.5],
7             [0.3, 7.5, 5.5, 9.7, 4.5, 5.7, 4.5, 0.9, 7.7, 5.5],
8             [1.1, 3.3, 4.1, 5.1, 5.3, 5.3, 5.9, 9.1, 4.3, 5.5],
9             [3.3, 4.4, 5.3, 1.5, 0.3, 0.5, 5.3, 7.3, 5.3, 3.1],
10            [4.7, 0.5, 0.4, 0.7, 0.1, 0.3, 5.3, 5.9, 5.9, 5.7]]
11
12  #panjang weights
13  weights1 = [[1.3, 3.3, 3.4, 4.5, 5.5, 5.5, 9.5, 5.5, 9.1, 7.3],
14              [5.7, 7.9, 9.1, 3.4, 3.5, 3.5, 3.3, 1.5, 3.5, 5.5],
15              [5.3, 5.3, 4.3, 1.9, 9.7, 7.5, 1.4, 3.1, 5.5, 5.5],
16              [5.5, 5.5, 5.4, 4.3, 3.3, 3.1, 5.5, 5.1, 3.3, 1.1],
17              [7.4, 5.4, 5.9, 5.1, 9.3, 7.3, 3.5, 5.3, 3.1, 0.3]]
18
19  #biases pada layer1
20  biases1 = [4.4, 1.3, 3.5, 4.5, 5.9]
21
22  #variable 3
23  weights3 = [ [0.1, 3.3, 3.4, 5.4, 4.5],
24               [4.3, 1.1, 3.5, 5.7, 5.5],
25               [3.5, 3.4, 9.7, 7.5, 5.4]]
26
27  #biases pada layer3
28  biases3 = [3.4, 5.1, 5.5]
29
30  #menghitung layer1 menggunakan inputs, weights1, dan biases 1 menggunakan perkalian dot
31  layer1 = np.dot(inputs, np.array(weights1).T) + biases1
32
33  #menghitung layer3 dari hasil perhitungan layer1 menggunakan dot
34  layer3 = np.dot(layer1, np.array(weights3).T) + biases3
35
36  #print output layer3
37  print(layer3)
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
NOVO/OneDrive/Desktop/vcode/No 1a Numpy2.py"
[[3297.485 4168.085 6078.342]
 [2259.134 3034.202 4366.005]
 [4056.194 5132.238 7409.899]
 [3760.796 4901.936 6939.453]
 [2832.706 3587.874 5059.334]
 [2065.572 2807.673 3919.738]]
PS C:\Users\LENOVO\OneDrive\Desktop\vcode>
```

Jelaskan cara kerja :

1.) Menggunakan fungsi numpy agar python mendukung penggunaan array

```
No 1a Numpy2.py > ...
1  #menggunakan fungsi numpy
2  import numpy as np
3
```

2.) inputkan data berjumlah 10 dan batch 6

```

3
4 #input data
5 inputs = [ [1.3, 3.3, 3.4, 5.5, 5.7, 7.9, 1.4, 3.9, 3.5, 4.9],
6            [0.5, 0.5, 1.3, 3.5, 3.5, 4.5, 1.5, 3.9, 5.9, 5.5],
7            [0.3, 7.5, 5.5, 9.7, 4.5, 5.7, 4.5, 0.9, 7.7, 5.5],
8            [1.1, 3.3, 4.1, 5.1, 5.3, 5.3, 5.9, 9.1, 4.3, 5.5],
9            [3.3, 4.4, 5.3, 1.5, 0.3, 0.5, 5.3, 7.3, 5.3, 3.1],
10           [4.7, 0.5, 0.4, 0.7, 0.1, 0.3, 5.3, 5.9, 5.9, 5.7]]
11

```

3.) dan weight 10x5

```

11
12 #panjang weights
13 weights1 = [[1.3, 3.3, 3.4, 4.5, 5.5, 5.5, 9.5, 5.5, 9.1, 7.3],
14            [5.7, 7.9, 9.1, 3.4, 3.5, 3.5, 3.3, 1.5, 3.5, 5.5],
15            [5.3, 5.3, 4.3, 1.9, 9.7, 7.5, 1.4, 3.1, 5.5, 5.5],
16            [5.5, 5.5, 5.4, 4.3, 3.3, 3.1, 5.5, 5.1, 3.3, 1.1],
17            [7.4, 5.4, 5.9, 5.1, 9.3, 7.3, 3.5, 5.3, 3.1, 0.3]]
18

```

4.) lalu gunakan fungsi np.dot untuk mengalikan weight dan input dan mentranspose keduanya lalu ditambahkan dengan biases

```

29
30 #menghitung layer1 menggunakan inputs, weights1, dan biases 1 menggunakan perkalian dot
31 layer1 = np.dot(inputs, np.array(weights1).T) + biases1
32

```

5.) lalu gunakan np.dot lagi untuk mengalikan layer1 hasil tadi dengan weight3 dan mentranspose keduanya lalu menambahkan dengan biases3

```

32
33 #menghitung layer3 dari hasil perhitungan layer1 menggunakan dot
34 layer3 = np.dot(layer1, np.array(weights3).T) + biases3
35

```

6.)lalu print hasil dari layer3 atau hasil dari semua proses

```

35
36 #print output layer3
37 print(layer3)

```