

**UTS JARINGAN KOMPUTER  
PART 2**



**OLEH  
ACHMAD NURS SYURURI ARIFIN  
21091397024  
D4 MANAJEMEN INFORMATIKA**

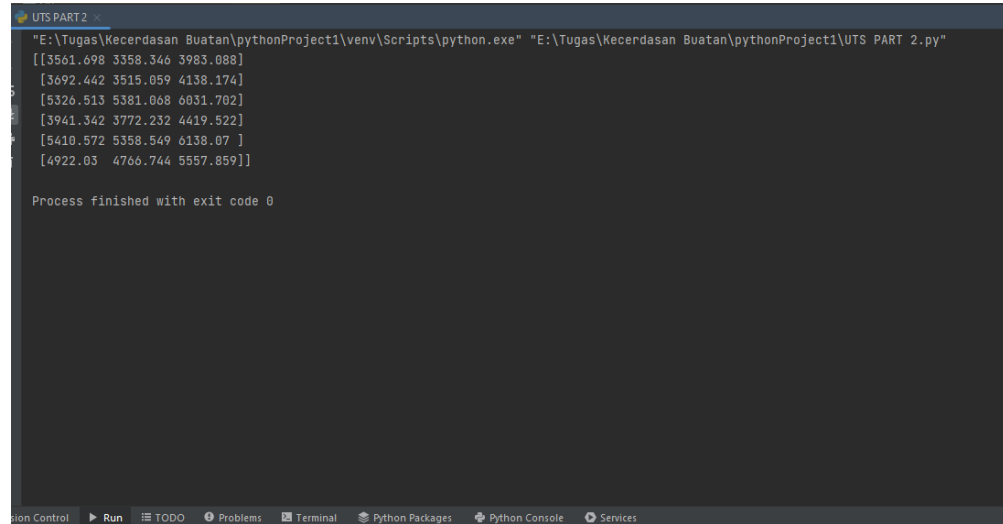
## Multi Neuron Batch Input

- Input layer feature 10
- Per batch nya 6 input
- Hidden layer 1, 5 neuron
- Hidden layer 2, 3 neuron

Input :

```
UTS PART 2.py
1  # Achmad Nurs Syururi Arifin (21091397024)
2
3  # inisialisasi numpys
4  import numpy as np
5
6  # inisialisasi variable
7  # Input layer feature 10
8  # Per batchnya 6 input
9  inputs = [[1.0, 1.5, 2.0, 2.5, 3.3, 3.5, 4.9, 4.5, 5.0, 5.5],
10           [1.7, 1.6, 2.4, 2.8, 3.4, 3.6, 4.4, 4.8, 5.2, 5.4],
11           [9.2, 4.2, 1.3, 8.2, 2.4, 8.4, 5.8, 7.4, 1.6, 9.3],
12           [2.8, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
13           [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 7.2, 9.4, 1.2, 3.4],
14           [2.3, 5.4, 2.4, 3.2, 3.4, 4.2, 6.4, 7.7, 8.2, 2.5]]
15
16  # inisialisasi bobot variable
17  # jumlah weight sesuai dengan jumlah neuron layer1, yaitu 5
18  weights1 = [[0.5, 0.7, 1.4, 2.7, 7.8, 9.4, 3.2, 4.6, 7.3, 9.4],
19             [1.5, 3.4, 0.9, 3.2, 0.4, 0.1, 2.8, 6.2, 8.4, 3.7],
20             [2.7, 1.3, 1.4, 7.2, 9.8, 0.2, 6.5, 8.4, 5.3, 6.4],
21             [0.1, 9.3, 4.2, 7.4, 0.3, 2.5, 1.3, 9.3, 8.2, 4.5],
22             [2.7, 8.5, 0.2, 1.5, 3.2, 1.9, 0.8, 4.3, 6.4, 4.8]]
23
24  # inisialisasi bias
25  # jumlah bias pada layer1, yaitu 5
26
27
28  [2.7, 8.5, 0.2, 1.5, 3.2, 1.9, 0.8, 4.3, 6.4, 4.8]]
29
30  # inisialisasi bias
31  # jumlah bias pada layer1, yaitu 5
32  bias1 = [2.3, 3.5, 0.1, 1.5, 3.9]
33
34  # panjang weights sesuai dengan neuron layer1, yaitu 5
35  # jumlah weights sesuai dengan jumlah neuron layer2, yaitu 3
36  weights2 = [[2.4, 7.3, 4.9, 1.3, 8.3],
37             [3.5, 2.6, 1.7, 6.4, 7.5],
38             [1.6, 7.9, 6.3, 2.6, 8.4]]
39
40  # jumlah bias pada layer2, yaitu 3 neuron
41  bias2 = [3.1, 1.2, 4.5]
42
43  # command untuk menghitung layer1 menggunakan inputs, weights1, dan biases1
44  layer1_outputs = np.dot(inputs, np.array(weights1).T) + bias1
45
46  # command untuk menghitung layer2 menggunakan hasil perhitungan pada layer1
47  layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + bias2
48
49  # mencetak output layer2
50  print(layer2_outputs)
```

Ouput :



```
"E:\Tugas\Kecerdasan Buatan\pythonProject1\venv\Scripts\python.exe" "E:\Tugas\Kecerdasan Buatan\pythonProject1\UTS PART 2.py"
[[3561.698 3358.346 3983.088]
 [3692.442 3515.059 4138.174]
 [5326.513 5381.068 6031.702]
 [3941.342 3772.232 4419.522]
 [5410.572 5358.549 6138.07 ]
 [4922.03 4766.744 5557.859]]

Process finished with exit code 0
```

Penjelasan:

Jumlah weight dan bias sesuai dengan banyaknya jumlah neuron. Jumlah angka pada weight sebanyak jumlah layer. Pada source code ini, input dioperasikan menggunakan operasi dot dan dikalikan dengan weight menggunakan array kemudian ditambahkan dengan bias. Terdapat dua penghitungan layer, yaitu pada layer 1 dan layer 2. Untuk output yang diminta adalah output layer 2 dari hasil perhitungan layer 1 dan dihitung menggunakan weights2 dan biases2. Lalu hasilnya akan dicetak melalui command print.