

**LAPORAN PRATIKUM UTS**  
**KECERDASAN BUATAN**  
**NEURAL NETWORKS**



Disusun oleh:  
Nila Gayatri

NIM/Kelas:  
21091397066/2021B

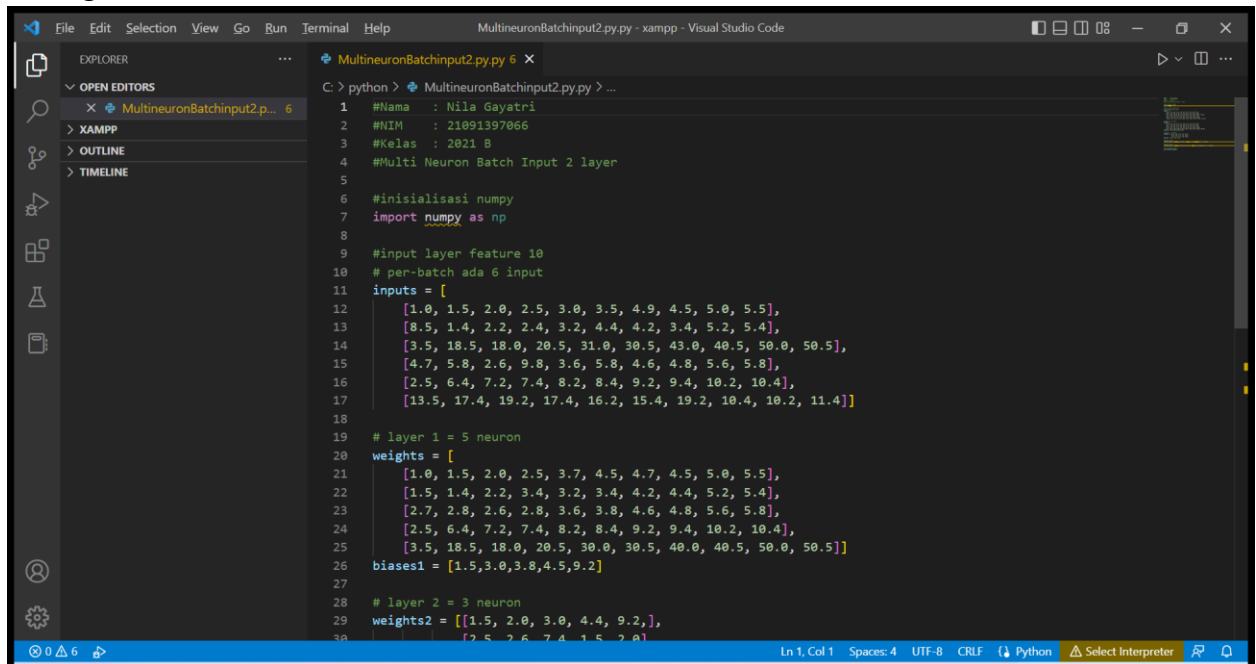
**D4 MANAJEMEN INFORMATIKA**  
**FAKULTAS VOKASI**  
**UNIVERSITAS NEGERI SURABAYA**  
**2022**

## UTS 2

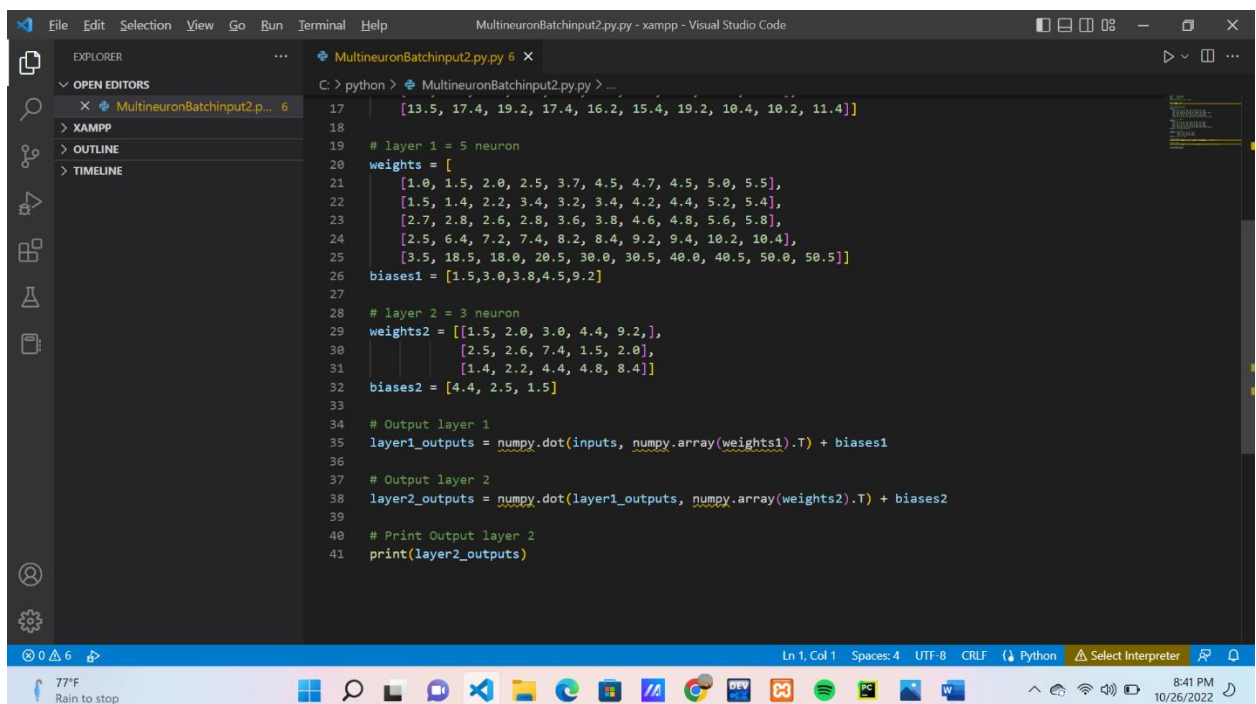
### 1. Multi Neuron Batch Input

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

### Codingan

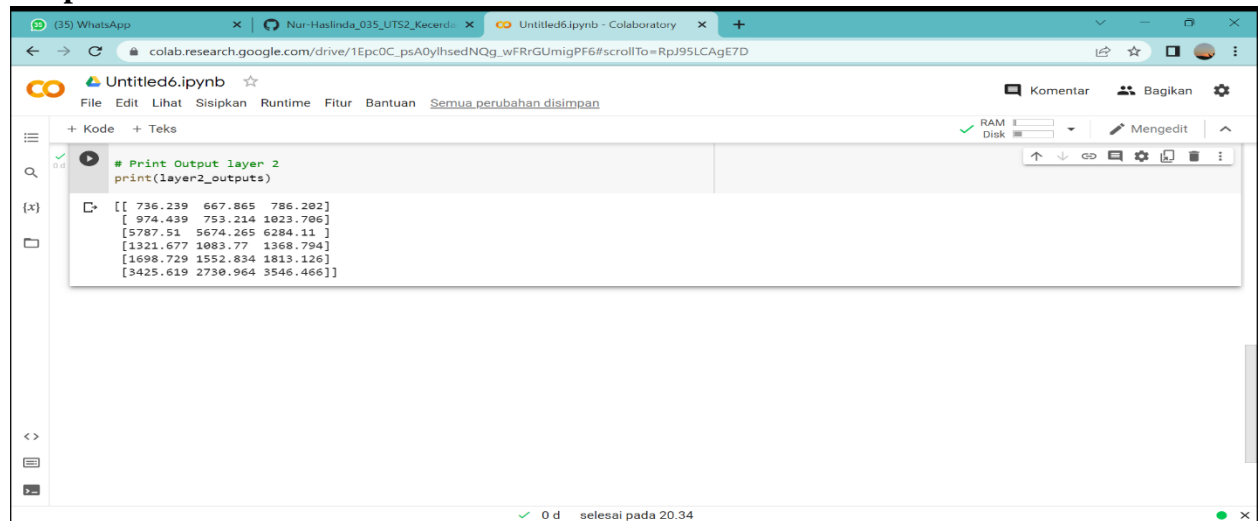


```
File Edit Selection View Go Run Terminal Help MultineuronBatchinput2.py.py - xampp - Visual Studio Code
EXPLORER
  OPEN EDITORS
    MultineuronBatchinput2.py.py 6
  XAMPP
  OUTLINE
  TIMELINE
C:\> python > MultineuronBatchinput2.py.py > ...
1 #Nama : Nila Gayatri
2 #NIM : 21091397066
3 #Kelas : 2021 B
4 #Multi Neuron Batch Input 2 layer
5
6 #inisialisasi numpy
7 import numpy as np
8
9 #input layer feature 10
10 # per-batch ada 6 input
11 inputs = [
12     [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.9, 4.5, 5.0, 5.5],
13     [8.5, 1.4, 2.2, 2.4, 3.2, 4.4, 4.2, 3.4, 5.2, 5.4],
14     [3.5, 18.5, 18.0, 20.5, 31.0, 30.5, 43.0, 40.5, 50.0, 50.5],
15     [4.7, 5.8, 2.6, 9.8, 3.6, 5.8, 4.6, 4.8, 5.6, 5.8],
16     [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
17     [13.5, 17.4, 19.2, 17.4, 16.2, 15.4, 19.2, 10.4, 10.2, 11.4]]
18
19 # layer 1 = 5 neuron
20 weights = [
21     [1.0, 1.5, 2.0, 2.5, 3.7, 4.5, 4.7, 4.5, 5.0, 5.5],
22     [1.5, 1.4, 2.2, 3.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
23     [2.7, 2.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
24     [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
25     [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]]
26 biases1 = [1.5, 3.0, 3.8, 4.5, 9.2]
27
28 # layer 2 = 3 neuron
29 weights2 = [[1.5, 2.0, 3.0, 4.4, 9.2,],
30             [2.5, 2.6, 7.4, 1.5, 2.0],
31             [1.4, 2.2, 4.4, 4.8, 8.4]]
32 biases2 = [4.4, 2.5, 1.5]
```



```
File Edit Selection View Go Run Terminal Help MultineuronBatchinput2.py.py - xampp - Visual Studio Code
EXPLORER
  OPEN EDITORS
    MultineuronBatchinput2.py.py 6
  XAMPP
  OUTLINE
  TIMELINE
C:\> python > MultineuronBatchinput2.py.py > ...
17 [13.5, 17.4, 19.2, 17.4, 16.2, 15.4, 19.2, 10.4, 10.2, 11.4]]
18
19 # layer 1 = 5 neuron
20 weights = [
21     [1.0, 1.5, 2.0, 2.5, 3.7, 4.5, 4.7, 4.5, 5.0, 5.5],
22     [1.5, 1.4, 2.2, 3.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
23     [2.7, 2.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
24     [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
25     [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]]
26 biases1 = [1.5, 3.0, 3.8, 4.5, 9.2]
27
28 # layer 2 = 3 neuron
29 weights2 = [[1.5, 2.0, 3.0, 4.4, 9.2,],
30             [2.5, 2.6, 7.4, 1.5, 2.0],
31             [1.4, 2.2, 4.4, 4.8, 8.4]]
32 biases2 = [4.4, 2.5, 1.5]
33
34 # Output layer 1
35 layer1_outputs = numpy.dot(inputs, numpy.array(weights1).T) + biases1
36
37 # Output layer 2
38 layer2_outputs = numpy.dot(layer1_outputs, numpy.array(weights2).T) + biases2
39
40 # Print Output layer 2
41 print(layer2_outputs)
```

## Output



```
# Print Output layer 2
print(layer2_outputs)
```

```
[[ 736.239  667.865  786.202]
 [ 974.439  753.214 1023.706]
 [5787.51  5674.265 6284.11 ]
 [1321.677 1083.77  1368.794]
 [1698.729 1552.834 1813.126]
 [3425.619 2730.964 3546.466]]
```

## Analisa

- Program di atas ialah merupakan program multineuron layer 2 dimana. Input yang dimiliki oleh multineuron layer 2 berjumlah 10 dan dengan jumlah 6 input. Terdapat pada layer 1 yang berjumlah 5 neuron berdasarkan neuron dan jumlah yang terdapat pada kolom weight ialah 10.
- Pada perhitungan dot. Product dimana terdapat pada layer 1 dan dengan menggunakan fungsi np dot. Kemudian di transpose dengan cara mengoprasikan input dan array pada transpose dengan menambahkan biases pada layer 1
- Pada perhitungan dot product yang terdapat pada layer 2 dengan menggunakan input dan array pada transpose dan dengan menambahkan biases pada layer 2
- Dimulai pada baris ke 6 untuk mengintalasi numpy gunannya untuk menghitung dot.product
- Pada baris ke 9 gunanya untuk input layer feature 10 dan pada baris ke 10 bachth input 6
- Kemudian pada baris ke 19 merupakan terdapat variable bias layer 1 yakni dari bias layer 1 dan weight 1 neuron 5
- Baris ke 28 merupakan terdapat variable bias layer 2 yakni dari layer 2 dan weight 2 dengan dari neuron 3
- Baris ke 34 untuk menghitung output dari layer 1 dengan menggunakan rumus numpy dan menghitung input, weight dan biasnya
- Baris ke 37 untuk menghitung output dari layer 2 dengan menggunakan rumus dengan yang sama dan mehitung input, weight dan biasnya.

## Cara kerja

### Input:

Dengan input yang berjumlah 10 dan bachth 6 jumlah baris harus sesuai dengan input pada layer dengan sesuai kolom jumlah (6x10)

```
inputs = [
    [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.9, 4.5, 5.0, 5.5],
    [8.5, 1.4, 2.2, 2.4, 3.2, 4.4, 4.2, 3.4, 5.2, 5.4],
    [3.5, 18.5, 18.0, 20.5, 31.0, 30.5, 43.0, 40.5, 50.0, 50.5],
    [4.7, 5.8, 2.6, 9.8, 3.6, 5.8, 4.6, 4.8, 5.6, 5.8],
    [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
    [13.5, 17.4, 19.2, 17.4, 16.2, 15.4, 19.2, 10.4, 10.2, 11.4]]
```

### Weight 1

- Weights yang pada baris pertama neuron1
- Weights yang pada baris ke dua neuron 2
- Weights yang pada baris ke tiga neuron 3
- Weights yang pada baris ke empat neuron 4
- Weights yang pada baris ke lima neuron 5

```
weights = [
    [1.0, 1.5, 2.0, 2.5, 3.7, 4.5, 4.7, 4.5, 5.0, 5.5],
    [1.5, 1.4, 2.2, 3.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
    [2.7, 2.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
    [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
    [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]]
```

### Biases layer 1

```
biases1 = [1.5, 3.0, 3.8, 4.5, 9.2]
```

**Rumus:**  $\text{numpy.dot}(\text{input}, \text{numpy.array}(\text{weights1}).T) + \text{biases1}$

### Output yang di hasilkan

```
↳ [[ 736.239  667.865  786.202]
    [ 974.439  753.214 1023.706]
    [5787.51  5674.265 6284.11 ]
    [1321.677 1083.77  1368.794]
    [1698.729 1552.834 1813.126]
    [3425.619 2730.964 3546.466]]
```

**Weights 2 ialah :**

Jumlah yang terdiri atas weight 2 jumlah kolom yang terdapat di weight 2 merupakan sama seperti layer 2 = 3 neuron tersebut dengan (5x3)

```
weights2 = [[1.5, 2.0, 3.0, 4.4, 9.2,],  
            [2.5, 2.6, 7.4, 1.5, 2.0],  
            [1.4, 2.2, 4.4, 4.8, 8.4]]
```

Weights 2 merupakan neuron 1 pada layer 2

Weights 2 merupakan neuron 2 pada layer ke 2

Weights 2 merupakan neuron 3 pada layer ke 2

**Biases pada layer ke 2**

```
biases2 = [4.4, 2.5, 1.5]
```

**Rumus:  $\text{numpy.dot}(\text{input}, \text{numpy.array}(\text{weights1}).T) + \text{biases2}$**

**Output yang di hasilkan pada layer ke 2**

```
↳ [[ 1785.691  1484.477  1905.934]  
    [ 2251.216  2118.346  2472.068]  
    [15423.295 13157.775 16576.07 ]  
    [ 2841.424  2572.082  3086.368]  
    [ 4100.156  3657.046  4441.728]]
```