

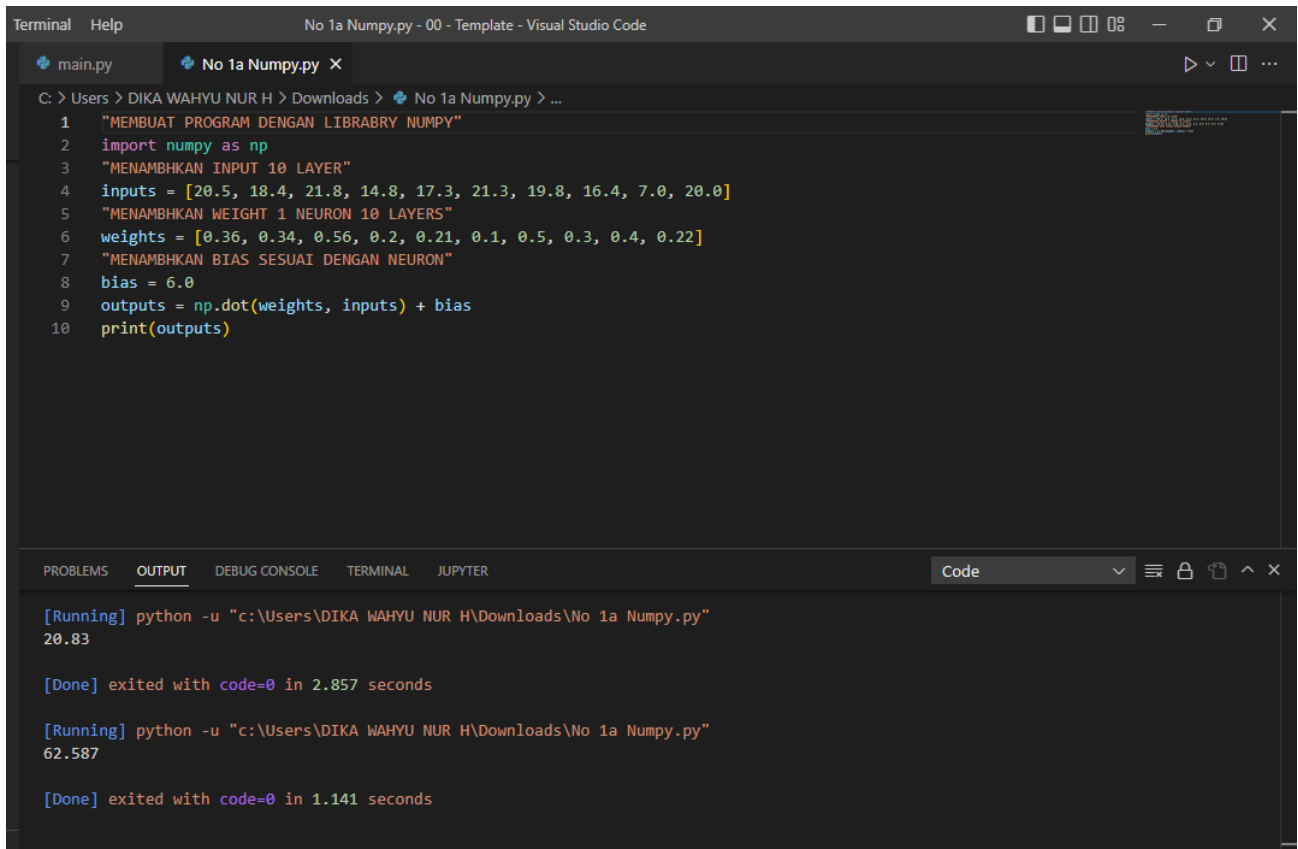
LAPORAN UTS  
Kecerdasan Buatan



Oleh :  
Dika Wahyu Nuralixsyah 21091397029  
2021A

PROGRAM VOKASI UNIVERSITAS NEGERI SURABAYA  
MANAJEMEN INFORMATIKA  
2021

## 1.A



The screenshot shows a Visual Studio Code window with a file named 'No 1a Numpy.py'. The code is a simple Python script using NumPy for a single-layer perceptron calculation. The script defines inputs, weights, and a bias, then calculates the output using a dot product and adds the bias. The output is printed. Below the code editor, the 'OUTPUT' panel shows the execution results of two runs. The first run outputs 20.83, and the second run outputs 62.587. Both runs completed successfully with code=0.

```
1 "MEMBUAT PROGRAM DENGAN LIBRARY NUMPY"
2 import numpy as np
3 "MENAMBAHKAN INPUT 10 LAYER"
4 inputs = [20.5, 18.4, 21.8, 14.8, 17.3, 21.3, 19.8, 16.4, 7.0, 20.0]
5 "MENAMBAHKAN WEIGHT 1 NEURON 10 LAYERS"
6 weights = [0.36, 0.34, 0.56, 0.2, 0.21, 0.1, 0.5, 0.3, 0.4, 0.22]
7 "MENAMBAHKAN BIAS SESUAI DENGAN NEURON"
8 bias = 6.0
9 outputs = np.dot(weights, inputs) + bias
10 print(outputs)
```

[Running] python -u "c:\Users\DIKA WAHYU NUR H\Downloads\No 1a Numpy.py"  
20.83

[Done] exited with code=0 in 2.857 seconds

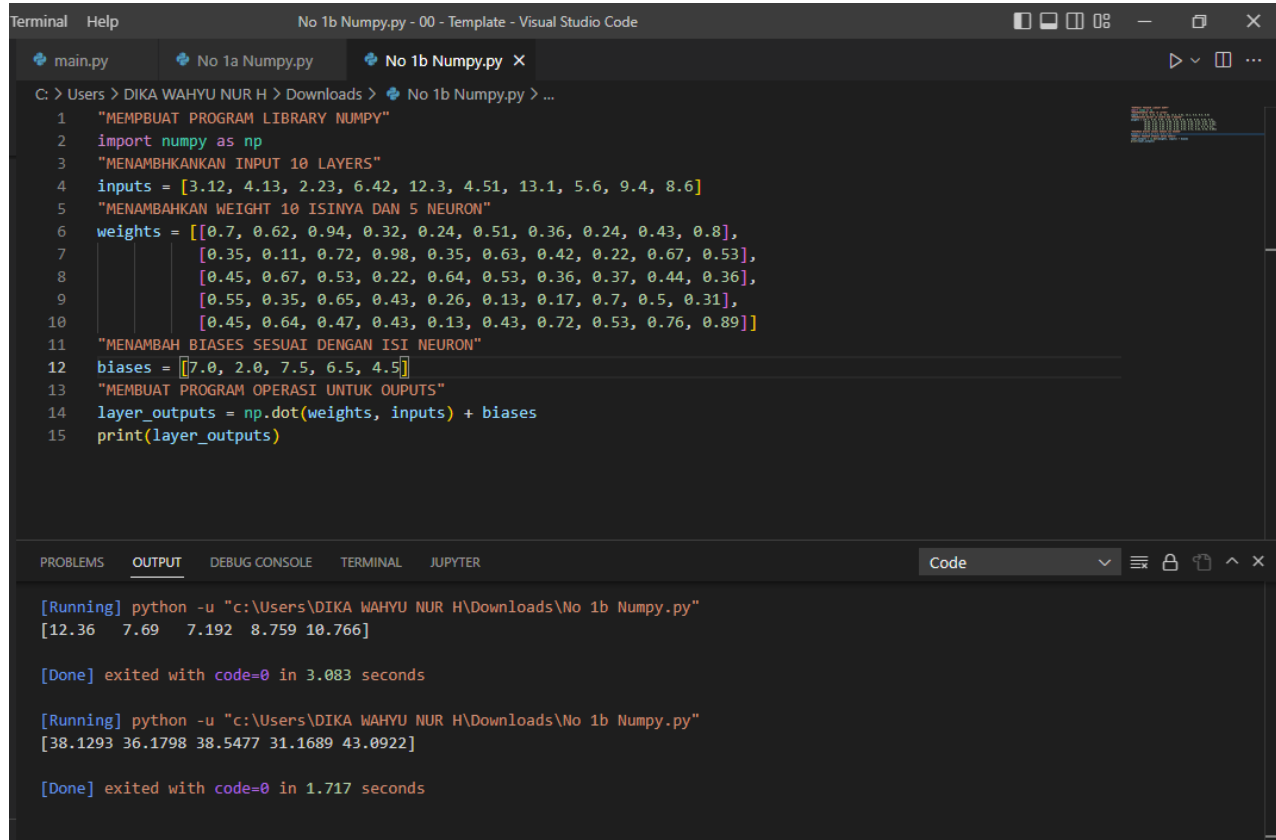
[Running] python -u "c:\Users\DIKA WAHYU NUR H\Downloads\No 1a Numpy.py"  
62.587

[Done] exited with code=0 in 1.141 seconds

### Analisa:

Neural Network paling sederhana adalah single layer perceptron, yang hanya memiliki single layer output node. Input data masuk langsung ke output neuron melalui serangkaian pembobot 'weight'. penjumlahan untuk semua perkalian dot antara input variable dan 'weight' ditiap koneksi neuron

## 1.B



```
Terminal  Help  No 1b Numpy.py - 00 - Template - Visual Studio Code

main.py  No 1a Numpy.py  No 1b Numpy.py X

C: > Users > DIKA WAHYU NUR H > Downloads > No 1b Numpy.py > ...
1  "MEMBUAT PROGRAM LIBRARY NUMPY"
2  import numpy as np
3  "MENAMBAHKAN INPUT 10 LAYERS"
4  inputs = [3.12, 4.13, 2.23, 6.42, 12.3, 4.51, 13.1, 5.6, 9.4, 8.6]
5  "MENAMBAHKAN WEIGHT 10 ISINYA DAN 5 NEURON"
6  weights = [[0.7, 0.62, 0.94, 0.32, 0.24, 0.51, 0.36, 0.24, 0.43, 0.8],
7             [0.35, 0.11, 0.72, 0.98, 0.35, 0.63, 0.42, 0.22, 0.67, 0.53],
8             [0.45, 0.67, 0.53, 0.22, 0.64, 0.53, 0.36, 0.37, 0.44, 0.36],
9             [0.55, 0.35, 0.65, 0.43, 0.26, 0.13, 0.17, 0.7, 0.5, 0.31],
10            [0.45, 0.64, 0.47, 0.43, 0.13, 0.43, 0.72, 0.53, 0.76, 0.89]]
11 "MENAMBAH BIASSES SESUAI DENGAN ISI NEURON"
12 biases = [7.0, 2.0, 7.5, 6.5, 4.5]
13 "MEMBUAT PROGRAM OPERASI UNTUK OUPUTS"
14 layer_outputs = np.dot(weights, inputs) + biases
15 print(layer_outputs)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER  Code

[Running] python -u "c:\Users\DIKA WAHYU NUR H\Downloads\No 1b Numpy.py"
[12.36  7.69  7.192  8.759 10.766]

[Done] exited with code=0 in 3.083 seconds

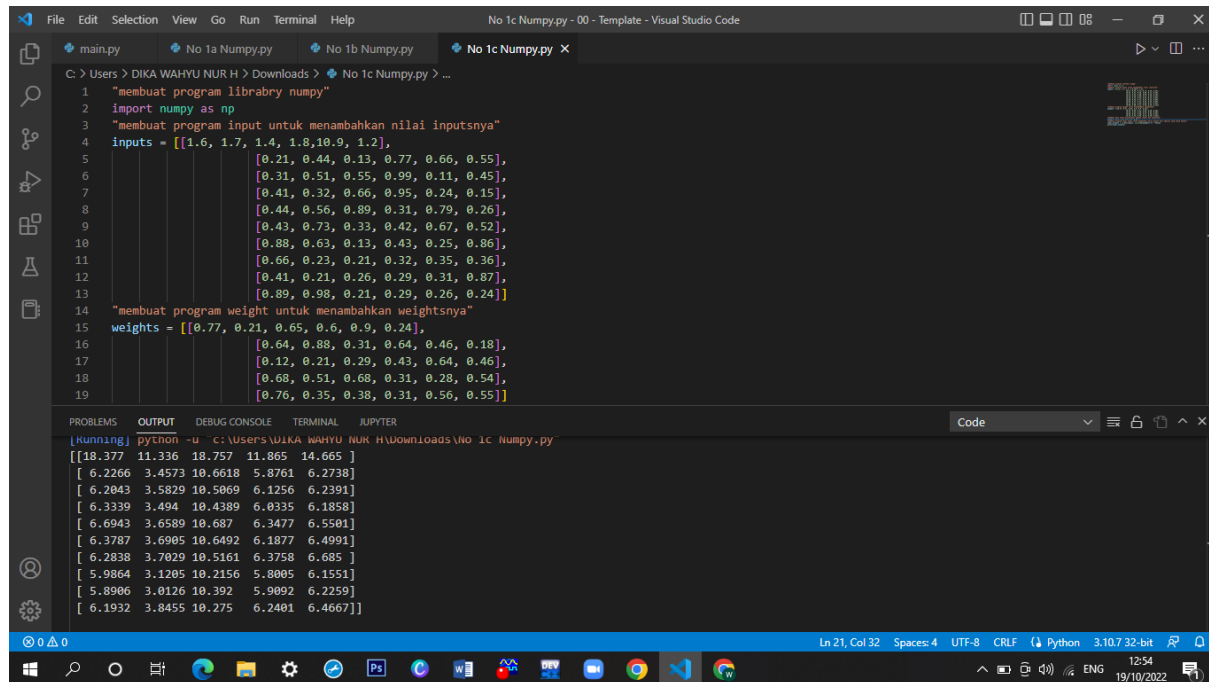
[Running] python -u "c:\Users\DIKA WAHYU NUR H\Downloads\No 1b Numpy.py"
[38.1293 36.1798 38.5477 31.1689 43.0922]

[Done] exited with code=0 in 1.717 seconds
```

Analisa:

Multi Layer Perceptron menghitung gradient dari [loss funtion](#) yang berhubungan dengan 'weight' tiap koneksi neuron dan akan meminimalkan loss saat 'weight' diubah

## 1.C



The screenshot shows a Visual Studio Code window with a Python file named 'No 1c Numpy.py'. The code defines input and weight matrices and prints their product. The output window shows the resulting 6x5 matrix.

```
1 "membuat program library numpy"
2 import numpy as np
3 "membuat program input untuk menambahkan nilai inputsnya"
4 inputs = [[1.6, 1.7, 1.4, 1.8, 10.9, 1.2],
5           [0.21, 0.44, 0.13, 0.77, 0.66, 0.55],
6           [0.31, 0.51, 0.55, 0.99, 0.11, 0.45],
7           [0.41, 0.32, 0.66, 0.95, 0.24, 0.15],
8           [0.44, 0.56, 0.89, 0.31, 0.79, 0.26],
9           [0.43, 0.73, 0.33, 0.42, 0.67, 0.52],
10          [0.88, 0.63, 0.13, 0.43, 0.25, 0.86],
11          [0.66, 0.23, 0.21, 0.32, 0.35, 0.36],
12          [0.41, 0.21, 0.26, 0.29, 0.31, 0.87],
13          [0.89, 0.98, 0.21, 0.29, 0.26, 0.24]]
14 "membuat program weight untuk menambahkan weightsnya"
15 weights = [[0.77, 0.21, 0.65, 0.6, 0.9, 0.24],
16            [0.64, 0.88, 0.31, 0.64, 0.46, 0.18],
17            [0.12, 0.21, 0.29, 0.43, 0.64, 0.46],
18            [0.68, 0.51, 0.68, 0.31, 0.28, 0.54],
19            [0.76, 0.35, 0.38, 0.31, 0.56, 0.55]]

[running] python -u "c:\Users\DIKA WAHYU NUR H\Downloads\No 1c Numpy.py"
[[18.377 11.336 18.757 11.865 14.665 ]
 [ 6.2266 3.4573 10.6618 5.8761 6.2738]
 [ 6.2043 3.5829 10.5069 6.1256 6.2391]
 [ 6.3339 3.494 10.4389 6.0335 6.1858]
 [ 6.6943 3.6589 10.687 6.3477 6.5501]
 [ 6.3787 3.6905 10.6492 6.1877 6.4991]
 [ 6.2838 3.7029 10.5161 6.3758 6.685 ]
 [ 5.9864 3.1205 10.2156 5.8005 6.1551]
 [ 5.8906 3.0126 10.392 5.9092 6.2259]
 [ 6.1932 3.8455 10.275 6.2401 6.4667]]
```

Analisa:

Mengubah weight menjadi bentuk yang sama dengan input menggunakan transpose lalu ditambah bias.