

# **LAPORAN UTS KECERDASAN BUATAN**



Disusun oleh :

Muhammad Zidan Alif Oktavian (21091397045)

---

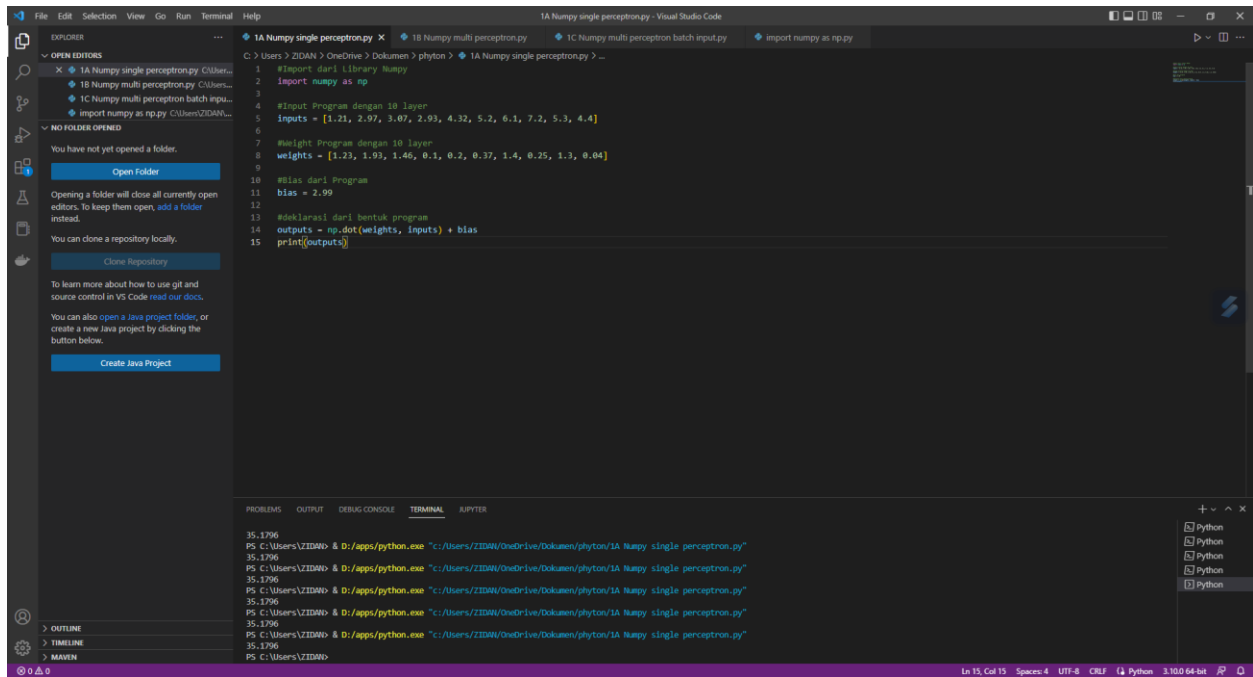
**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2022**

## No. 1A



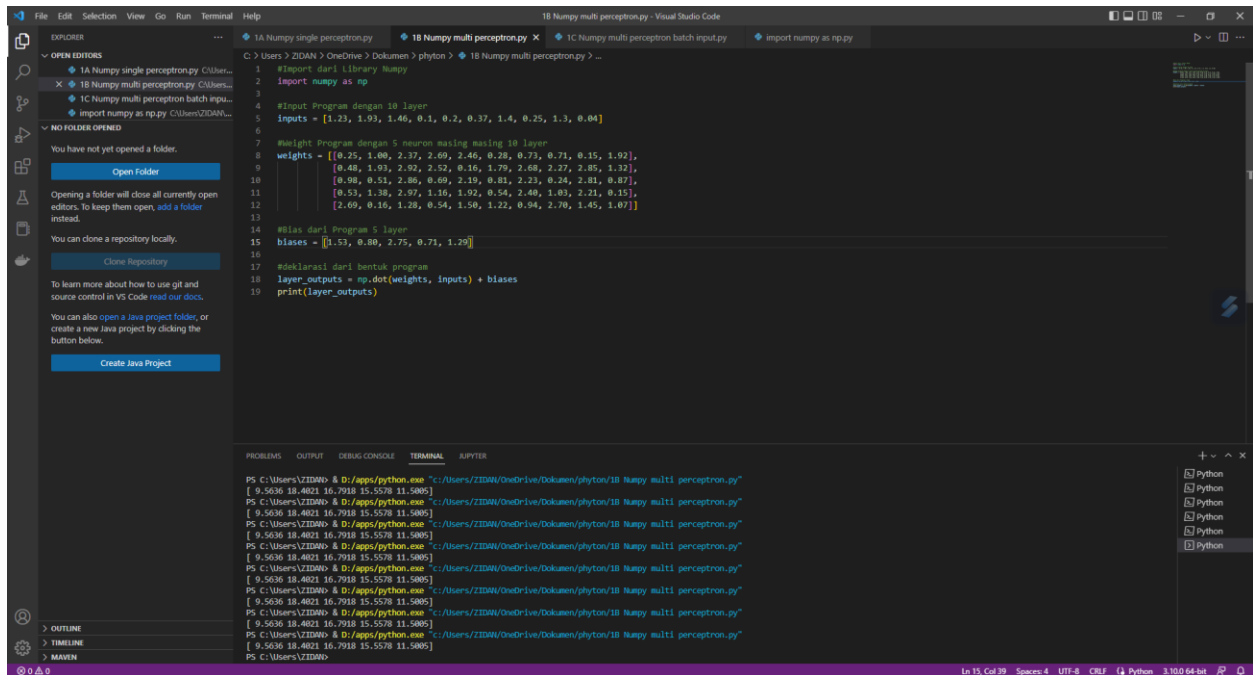
```
1 #Import dari Library Numpy
2 import numpy as np
3
4 #Input Program dengan 10 layer
5 inputs = [1.21, 2.97, 3.07, 2.93, 4.32, 5.2, 6.1, 7.2, 5.3, 4.4]
6
7 #Weight Program dengan 10 layer
8 weights = [1.23, 1.93, 1.46, 0.1, 0.2, 0.37, 1.4, 0.25, 1.3, 0.04]
9
10 #Bias dari Program
11 bias = 2.99
12
13 #Mendefinisikan dari bentuk program
14 outputs = np.dot(weights, inputs) + bias
15 print(outputs)
```

```
35.1796
PS C:\Users\ZIDAN> & D:/apps/python.exe "c:/Users/ZIDAN/OneDrive/Dokumen/python/1A Numpy single perceptron.py"
35.1796
PS C:\Users\ZIDAN> & D:/apps/python.exe "c:/Users/ZIDAN/OneDrive/Dokumen/python/1A Numpy single perceptron.py"
35.1796
PS C:\Users\ZIDAN> & D:/apps/python.exe "c:/Users/ZIDAN/OneDrive/Dokumen/python/1A Numpy single perceptron.py"
35.1796
PS C:\Users\ZIDAN> & D:/apps/python.exe "c:/Users/ZIDAN/OneDrive/Dokumen/python/1A Numpy single perceptron.py"
35.1796
PS C:\Users\ZIDAN> & D:/apps/python.exe "c:/Users/ZIDAN/OneDrive/Dokumen/python/1A Numpy single perceptron.py"
35.1796
PS C:\Users\ZIDAN>
```

### Analisis:

Single Neuron Perceptron adalah unit jaringan saraf tiruan paling dasar yang melakukan perhitungan yang tepat untuk mendeteksi fitur dalam data input dengan menambahkan weight dan bias.

## No. 1B



```
18 Numpy multi perceptron.py - Visual Studio Code
18 Numpy multi perceptron batch input.py
import numpy as np.py

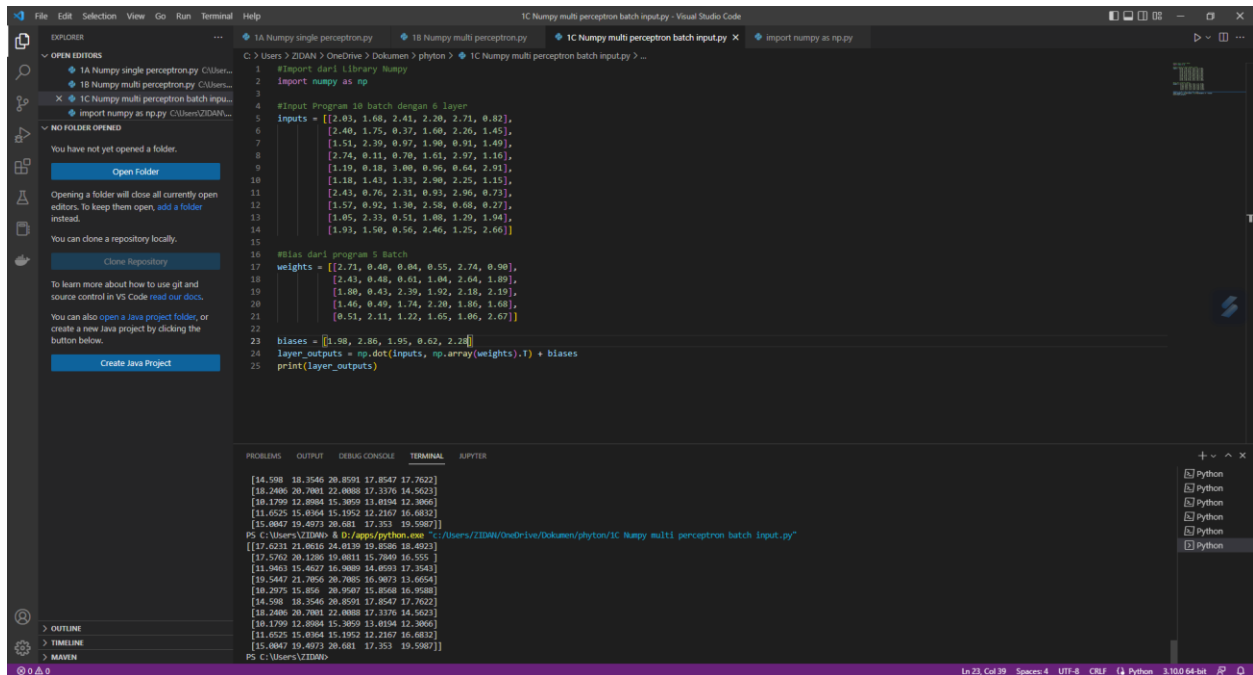
1 #Import dari library Numpy
2 import numpy as np
3
4 #Input Program dengan 10 layer
5 inputs = [1.23, 1.93, 1.46, 0.1, 0.2, 0.37, 1.4, 0.25, 1.3, 0.04]
6
7 #weight Program dengan 5 neuron masing masing 10 layer
8 weights = [[0.25, 1.00, 2.37, 2.69, 2.46, 0.20, 0.70, 0.71, 0.15, 1.02],
9            [0.40, 1.93, 2.02, 2.52, 0.10, 1.70, 2.66, 2.37, 2.05, 1.32],
10           [0.90, 0.51, 2.86, 0.09, 2.19, 0.81, 2.23, 0.24, 2.01, 0.87],
11           [0.53, 1.38, 2.97, 1.16, 1.92, 0.54, 2.40, 1.03, 2.21, 0.15],
12           [2.69, 0.16, 1.28, 0.54, 1.50, 1.22, 0.94, 2.70, 1.45, 1.07]]
13
14 #Bias dari Program 5 layer
15 biases = [1.53, 0.80, 2.75, 0.71, 1.29]
16
17 #deklaras dari bentuk program
18 layer_outputs = np.dot(weights, inputs) + biases
19 print(layer_outputs)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN> & D:/apps/python.exe "C:/Users/ZIDAN/OneDrive/Dokumen/python/18 Numpy multi perceptron.py"
[ 9.5636 18.4021 16.7918 15.5578 11.5005]
PS C:\Users\ZIDAN>
```

**Analisis:**  
Cara kerja multi layer adalah input layer menyuplai input vektor pada jaringan, kemudian input yang dimasukkan melakukan komputasi pada layer yang kedua, lalu output dari layer yang kedua digunakan sebagai input dari layer yang ketiga dan seterusnya.

## No. 1C



```
1 #Import dari library Numpy
2 import numpy as np
3
4 #Input Program 10 batch dengan 6 layer
5 inputs = [[2.03, 1.68, 2.41, 2.20, 2.71, 0.82],
6           [2.40, 1.75, 0.37, 1.60, 2.20, 1.45],
7           [1.51, 2.39, 0.97, 1.90, 0.91, 1.49],
8           [2.74, 0.11, 0.70, 1.51, 2.97, 1.16],
9           [1.19, 0.18, 1.00, 0.90, 0.54, 2.91],
10          [1.18, 1.43, 1.33, 2.90, 2.25, 1.15],
11          [2.43, 0.76, 2.31, 0.93, 2.90, 0.73],
12          [1.57, 0.92, 1.30, 2.58, 0.68, 0.27],
13          [1.05, 2.32, 0.51, 1.00, 1.20, 1.94],
14          [1.93, 1.50, 0.56, 2.46, 1.25, 2.66]]
15
16 #Bias dari program 5 batch
17 weights = [[2.71, 0.40, 0.04, 0.55, 2.74, 0.90],
18            [2.43, 0.40, 0.01, 1.04, 2.04, 1.80],
19            [1.00, 0.43, 2.39, 1.92, 2.18, 2.19],
20            [1.46, 0.49, 1.74, 2.20, 1.86, 1.68],
21            [0.51, 2.11, 1.22, 1.05, 1.06, 2.67]]
22
23 biases = [1.90, 2.80, 1.95, 0.62, 2.20]
24 layer_outputs = np.dot(inputs, np.array(weights).T) + biases
25 print(layer_outputs)
```

PROBLEMS OUTPUT DEBUG-CONSOLE TERMINAL JUPYTER

```
[16.598 18.3546 20.8591 17.8547 17.7622]
[18.2406 20.7801 22.8008 17.3376 14.5623]
[18.1799 12.8804 15.3059 13.0194 12.3066]
[11.6525 15.4064 15.1952 12.2167 16.6832]
[15.0047 19.4073 20.681 17.353 19.5987]
PS C:\Users\ZIDAN> & if ($?) {python.exe %*} else {python.exe %*}
[17.6231 21.0616 24.8139 19.8586 18.4923]
[17.5762 20.1286 19.0811 15.7869 16.555 ]
[11.9463 15.4027 16.8005 14.8593 17.3543]
[19.5447 21.7856 20.7085 16.9073 13.6654]
[18.2975 15.856 20.9567 15.8568 16.9588]
[14.550 20.2546 20.8591 17.8547 17.7622]
[18.2406 20.7801 22.8008 17.3376 14.5623]
[18.1799 12.8804 15.3059 13.0194 12.3066]
[11.6525 15.4064 15.1952 12.2167 16.6832]
[15.0047 19.4073 20.681 17.353 19.5987]
PS C:\Users\ZIDAN>
```

### Analisa:

Sebuah Perceptron Multilayer memiliki lapisan input dan output, dan satu atau lebih lapisan tersembunyi dengan banyak neuron ditumpuk bersama-sama. Dan sementara di Perceptron neuron harus memiliki fungsi aktivasi yang memberlakukan ambang batas, seperti ReLU atau sigmoid, neuron di Multilayer Perceptron dapat menggunakan fungsi aktivasi arbitrer apa pun.