

**“LAPORAN UJIAN TENGAH SEMESTER KECERDASAN  
BUATAN”**



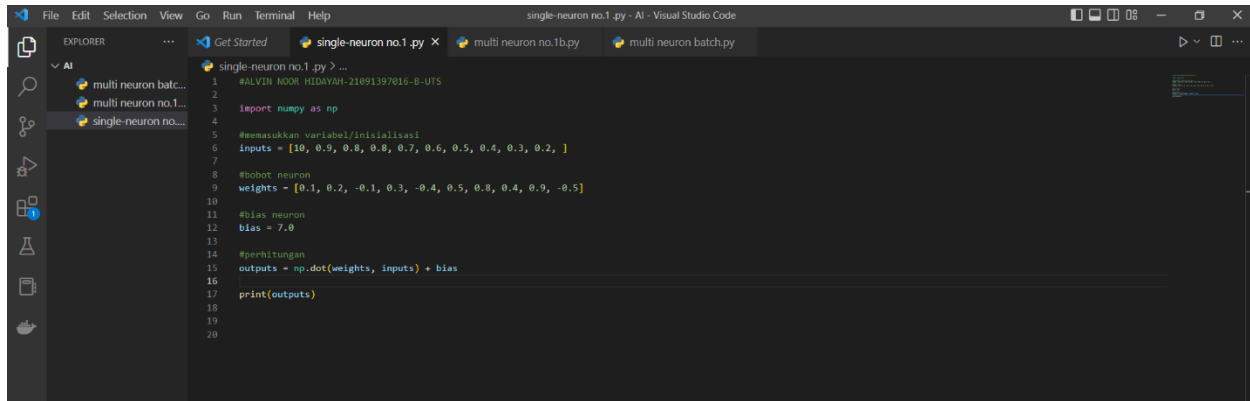
Oleh:

Alvin Noor Hidayah

21091397016

**UNIVERSITAS NEGERI SURABAYA  
FAKULTAS VOKASI  
PRODI D IV MANAJEMEN INFORMATIKA  
TAHUN AJARAN 2022/2023**

## Single neuron codingan

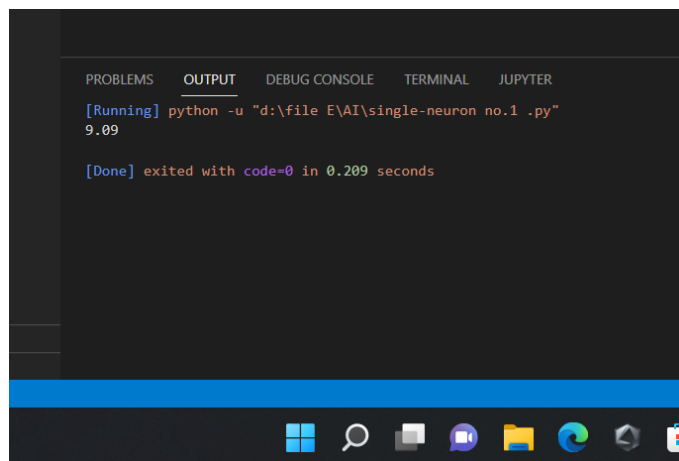


```
File Edit Selection View Go Run Terminal Help
single-neuron no.1 .py - AI - Visual Studio Code

EXPLORER
  AI
    multi-neuron batc...
    multi-neuron no.1...
    single-neuron no...

single-neuron no.1 .py > ...
1 #ALVIN HOOR. HIDAYAH-21091397016-B-UTS
2
3 import numpy as np
4
5 #memasukkan variabel/inisialisasi
6 inputs = [10, 0.9, 0.8, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, ]
7
8 #bobot neuron
9 weights = [0.1, 0.2, -0.1, 0.3, -0.4, 0.5, 0.8, 0.4, 0.9, -0.5]
10
11 #bias neuron
12 bias = 7.0
13
14 #perhitungan
15 outputs = np.dot(weights, inputs) + bias
16
17 print(outputs)
18
19
20
```

Output :

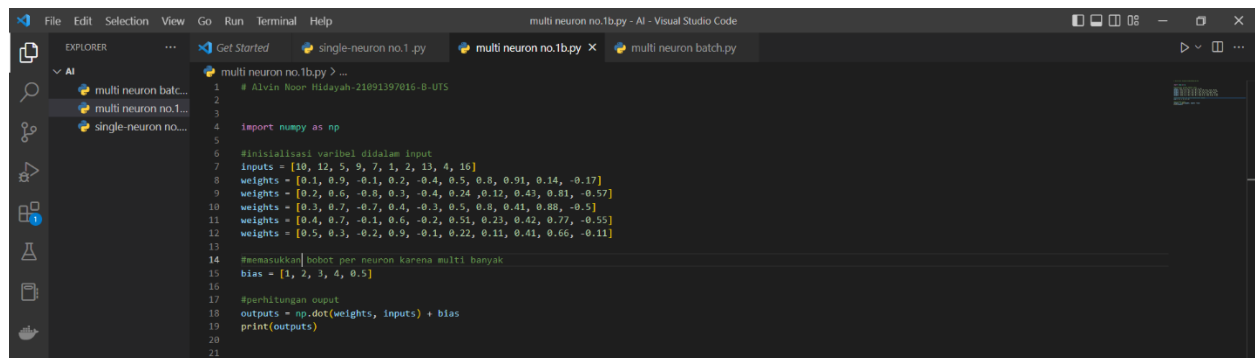


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
[Running] python -u "d:\file E\AI\single-neuron no.1 .py"
9.09

[Done] exited with code=0 in 0.209 seconds
```

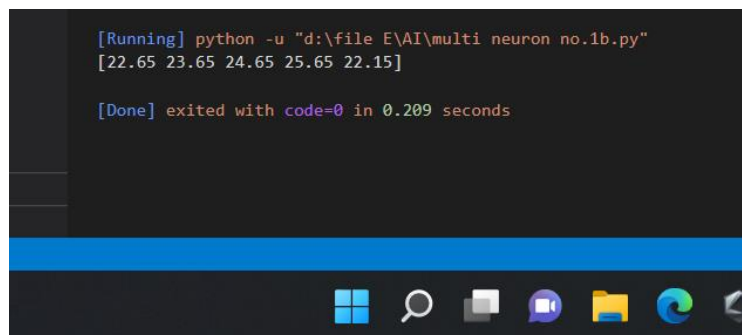
Analisa: dalam kode ini. 1 bobot saja dan 1 bias karena ini merupak single. Nah gunanya np.dot tersebut untuk perhitungan biar lebih sederhana.

## Multiple Neuron Codingan



```
1 # Alvin Noor Hidayah-21091397016-B-UTS
2
3
4 import numpy as np
5
6 #inisialisasi variabel didalam input
7 inputs = [10, 12, 5, 9, 7, 1, 2, 13, 4, 10]
8 weights = [0.1, 0.9, -0.1, 0.2, -0.4, 0.5, 0.8, 0.91, 0.14, -0.17]
9 weights = [0.2, 0.5, -0.8, 0.3, -0.4, 0.24, 0.12, 0.43, 0.81, -0.57]
10 weights = [0.3, 0.7, -0.7, 0.4, -0.3, 0.5, 0.8, 0.41, 0.88, -0.5]
11 weights = [0.4, 0.7, -0.1, 0.6, -0.2, 0.51, 0.23, 0.42, 0.77, -0.55]
12 weights = [0.5, 0.3, -0.2, 0.9, -0.1, 0.22, 0.11, 0.41, 0.66, -0.11]
13
14 #memasukkan bobot per neuron karena multi banyak
15 bias = [1, 2, 3, 4, 0.5]
16
17 #perhitungan output
18 outputs = np.dot(weights, inputs) + bias
19 print(outputs)
20
21
```

Ouput:

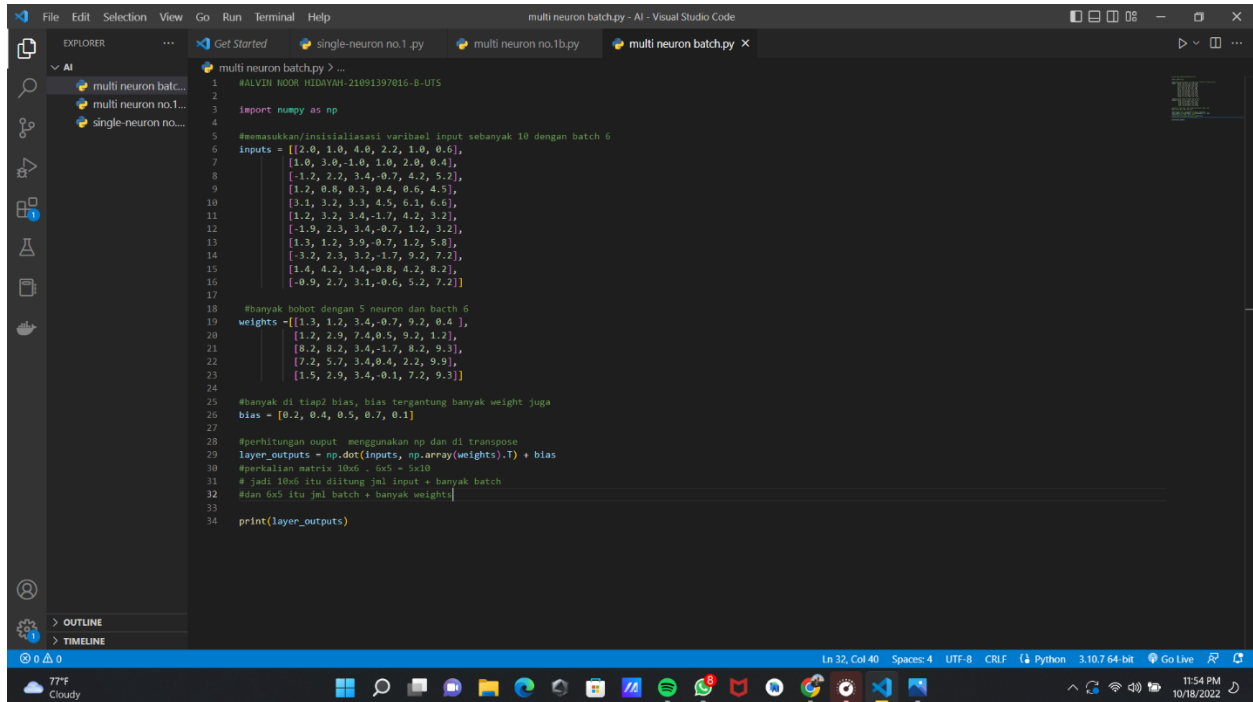


```
[Running] python -u "d:\file E\AI\multi neuron no.1b.py"
[22.65 23.65 24.65 25.65 22.15]

[Done] exited with code=0 in 0.209 seconds
```

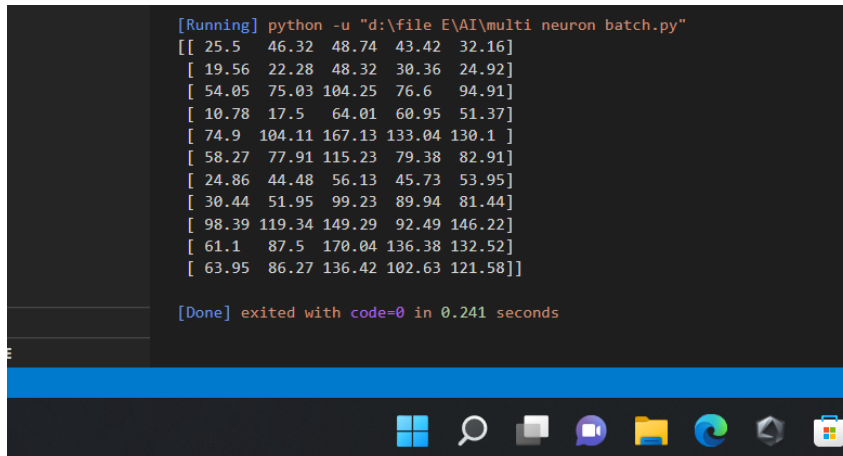
Analisa: Dalam kode diatas kami mempunyai 5 bobot dan 5 bias karena multiple lebih dari 1. Di setiap neuron/weights itu terhubung dengan input yg sama oleh karena itu saling terhubung. Didalam kode tersebut ada perhitungan np.dot berfungsi sebagai pergitungan sederhana agar tidak menghitung satu persatu weights dan inputnya.

## Multiple neuron batch codingan



```
1 #ALVIN MOOR HEDAYAH-21091397010-B-UTS
2
3 import numpy as np
4
5 #memasukkan/inisialisasi variabel input sebanyak 10 dengan batch 6
6 inputs = [[2.0, 1.0, 4.0, 2.2, 1.0, 0.6],
7           [1.0, 3.0,-1.0, 1.0, 2.0, 0.4],
8           [-1.2, 2.2, 3.4,-0.7, 4.2, 5.2],
9           [1.2, 0.8, 0.3, 0.4, 0.6, 4.5],
10          [3.1, 3.2, 3.3, 4.5, 6.1, 6.6],
11          [1.2, 3.2, 3.4,-1.7, 4.2, 3.2],
12          [-1.9, 2.3, 3.4,-0.7, 1.2, 3.2],
13          [1.3, 1.2, 3.9,-0.7, 1.2, 5.8],
14          [-3.2, 2.3, 3.2,-1.7, 9.2, 7.2],
15          [1.4, 4.2, 3.4,-0.8, 4.2, 8.2],
16          [-0.9, 2.7, 3.1,-0.6, 5.2, 7.2]]
17
18 #banyak bobot dengan 5 neuron dan batch 6
19 weights = [[1.3, 1.2, 3.4,-0.7, 9.2, 0.4],
20            [1.2, 2.9, 7.4,0.5, 9.2, 1.2],
21            [8.2, 8.2, 3.4,-1.7, 8.2, 9.3],
22            [7.2, 5.7, 3.4,0.4, 2.2, 9.9],
23            [1.5, 2.9, 3.4,-0.1, 7.2, 9.3]]
24
25 #banyak di tiap2 bias, bias tergantung banyak weight juga
26 bias = [0.2, 0.4, 0.5, 0.7, 0.1]
27
28 #perhitungan output menggunakan np dan di transpose
29 layer_outputs = np.dot(inputs, np.array(weights).T) + bias
30 #perhitungan matrix input = 6x5 = 5x10
31 #jadi jika itu diitung jel input + banyak batch
32 #dan 6x5 itu jel batch + banyak weights
33
34 print(layer_outputs)
```

Output :



```
[Running] python -u "d:\file E\AI\multi neuron batch.py"
[[ 25.5  46.32  48.74  43.42  32.16]
 [ 19.56  22.28  48.32  30.36  24.92]
 [ 54.05  75.03 104.25  76.6   94.91]
 [ 10.78  17.5   64.01  60.95  51.37]
 [ 74.9   104.11 167.13 133.04 130.1 ]
 [ 58.27  77.91 115.23  79.38  82.91]
 [ 24.86  44.48  56.13  45.73  53.95]
 [ 30.44  51.95  99.23  89.94  81.44]
 [ 98.39 119.34 149.29  92.49 146.22]
 [ 61.1   87.5   170.04 136.38 132.52]
 [ 63.95  86.27 136.42 102.63 121.58]]

[Done] exited with code=0 in 0.241 seconds
```

Analisa: masih sama di codingan diatas mempunyai 5 bobot dan 5 bias, dan input dengan neuron saling terhubung satu sama lain. Tapi disini ada yang unik dimana perhitungannya menggunakan np.dot dan matrix transpose dimana baris dan array dan kolom array dapat di seajarkan. Kemudian matrix yang dihasilkan terdiri dari keluaran semua neuron setelah operasi saat penginputan.

>> input batch itu yang horizontal

>> dan weights itu yang vertical

Yang sebelumnya matrix  $10 \times 6 \cdot 6 \times 10 = 5 \times 10$  karena ada perintah T yaitu transpose di metode tersebut ada array/ np. array