LAPORAN KECERDASAN BUATAN

UJIAN TENGAH SEMESTER

Disusun Oleh :

Shabinna Rahmadilla Santoso

21091397004

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

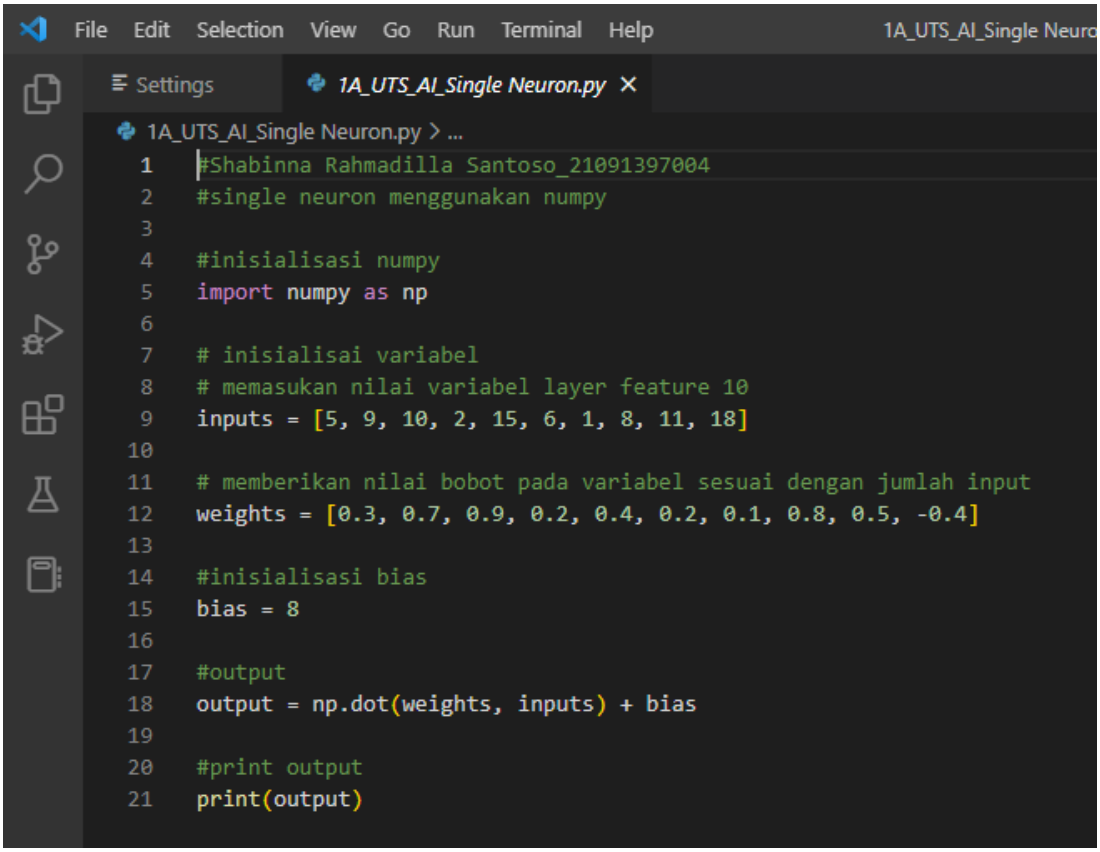
2022

UTS 1

A. Single Neuron

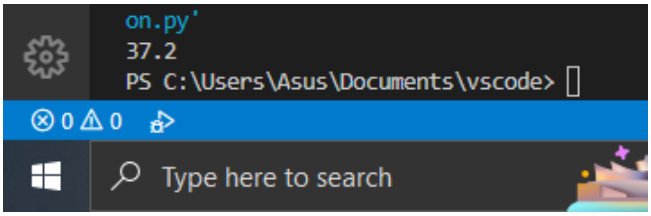
Input layer feature 10, Neuron 1

- Source Code



- Pada line 5 menginisialisasikan numpy ke np guna memudahkan dalam pengoperasionalkan source code ini, proses komputasi angka
- Di line 9 menginput variabel nilai sesuai dengan soal yaitu feature layer 10, jadi memasukkan angka sebanyak 10 baris angka
- Line 12 kita memasukkan weight atau bobot sesuai dengan input dan neuron yang telah ditentukan yaitu 1 neuron
- Bias yang terdapat pada line 15 berjumlah 1 karena sesuai dengan neuron yang ditetapkan yaitu 1
- Line 18 adalah output dari operasi perkalian antara weight dan input ditambah dengan bias, np.dot digunakan mengkalikan 2 matriks yaitu variabel weight dan input
- Di line 21 print output digunakan untuk menampilkan hasil operasi hitung dari source code tersebut

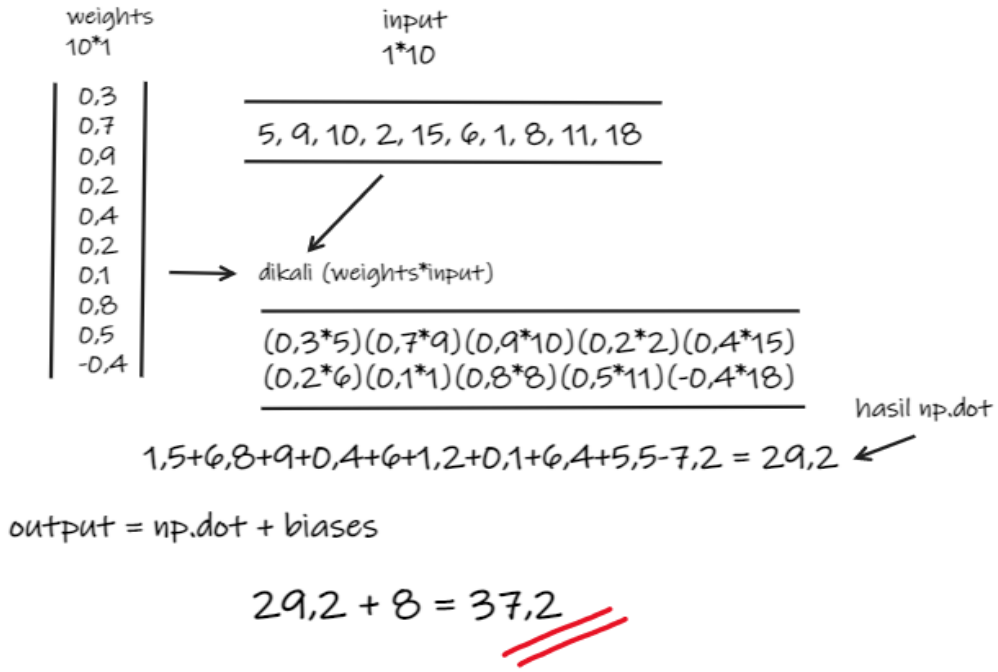
- Output



- Perhitungan output

```
#output
output = np.dot(weights, inputs) + bias
```

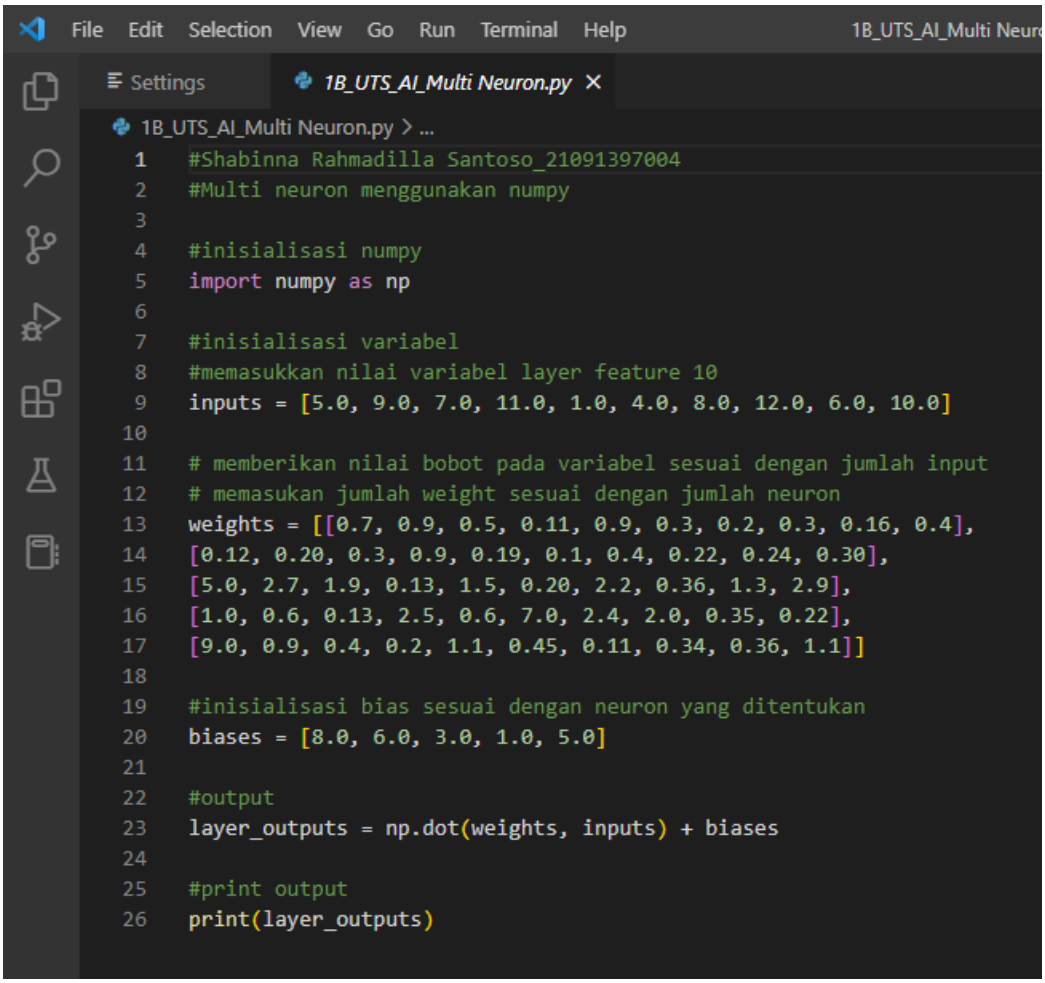
Dalam perhitungan np.dot weight dikali dengan input dan hasil np.dot ditambah dengan biases



B. Multi Neuron

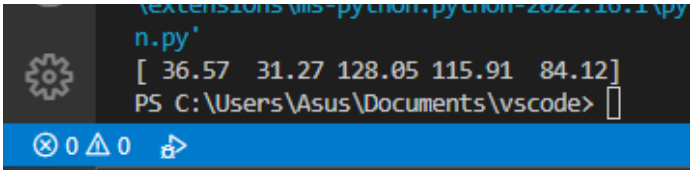
Input Layer Feature 10, Neuron 5

- Source Code



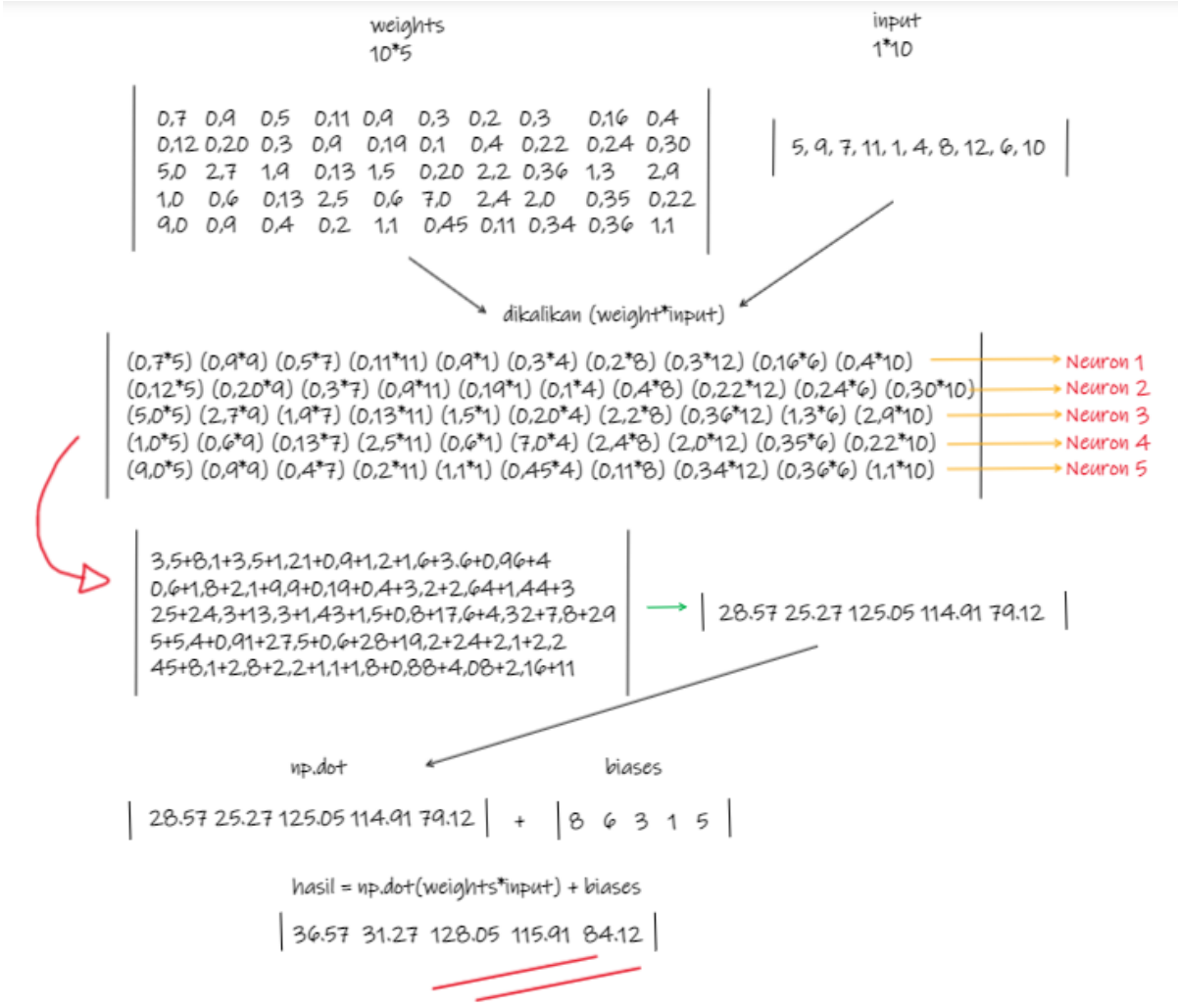
- Di line 5 menginisialisasikan numpy ke np guna memudahkan dalam pengoperasionalkan source code ini, proses komputasi angka
- Line 9 memasukkan nilai input sejumlah 10 baris angka sesuai dengan perintah yaitu feature layer 10
- Line 13 memasukkan nilai weight sejumlah 10 baris angka dan 5 kolom angka sesuai jumlah neuron karena neuron yang diminta adalah 5
- Line 20 memasukkan nilai bias sejumlah 5 baris angka karena neuron yang ditetapkan sebanyak 5
- Line 23 output dari operasi perkalian weight dan input ditambah bias, np.dot digunakan mengkalikan 2 matriks yaitu variabel weight dan input
- Line 26 menampilkan hasil perhitungan output dari source code tersebut

- Output



- Perhitungan output

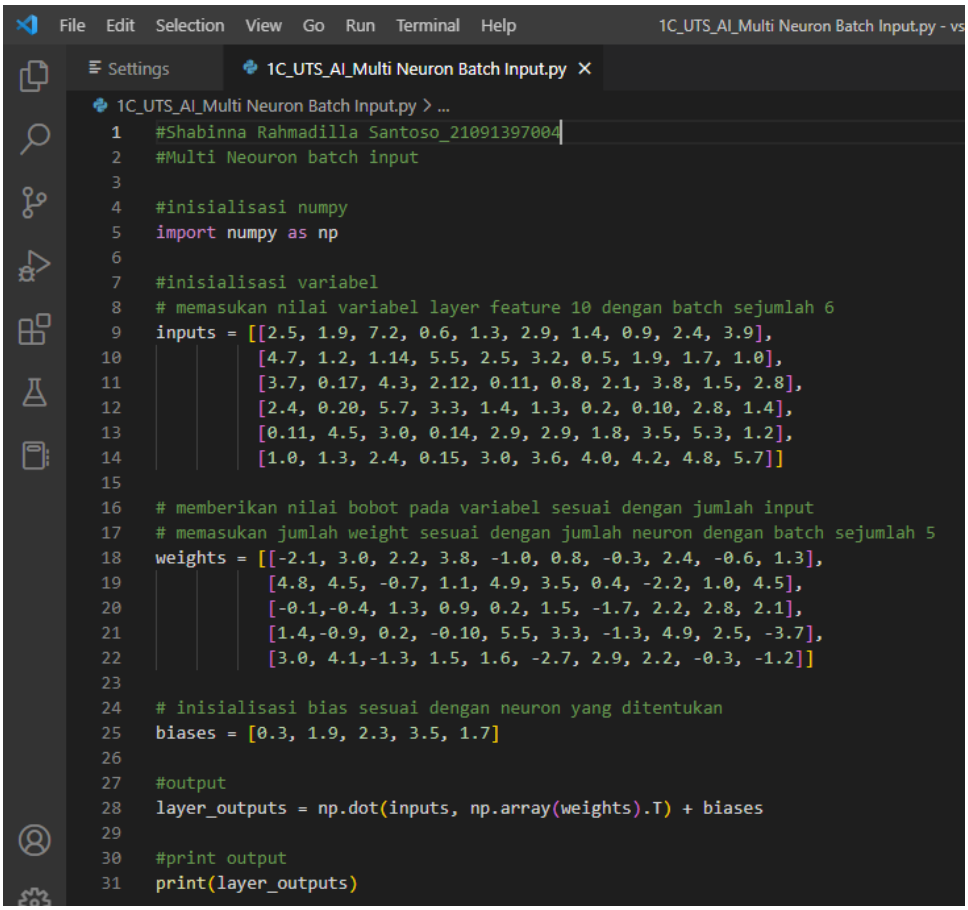
Dalam perhitungan np.dot weight dikali dengan input (sesuai dengan gambar) lalu hasil np.dot ditambah dengan biases



C. Multi Neuron Batch Input

Input layer feature 10, per batchnya 6 input, Neuron 5

- Source Code



- Line 5 menginisialisasikan numpy ke np guna memudahkan dalam pengoperasionalkan source code ini, proses komputasi angka
- Line 9 menginisialisasikan variabel, memasukkan nilai input dengan jumlah 10 baris angka sesuai dengan yang diminta yaitu feature layer 10 dan 6 kolom angka dengan perintah batch = 6
- Line 18 memasukan nilai weight dengan 5 kolom karena sesuai dengan neuron yang diminta yaitu 5 dan 10 baris angka
- Di line 25 memasukkan nilai bias sejumlah dengan banyaknya neuron yang ditetapkan yaitu 5
- Line 28 output operasi dari perkalian input dan weight yang kemudian di transpose dan ditambah oleh bias, np.dot digunakan mengkalikan 2 matriks yaitu variabel weight dan input
- Line 31 untuk menampilkan hasil output dari source code tersebut

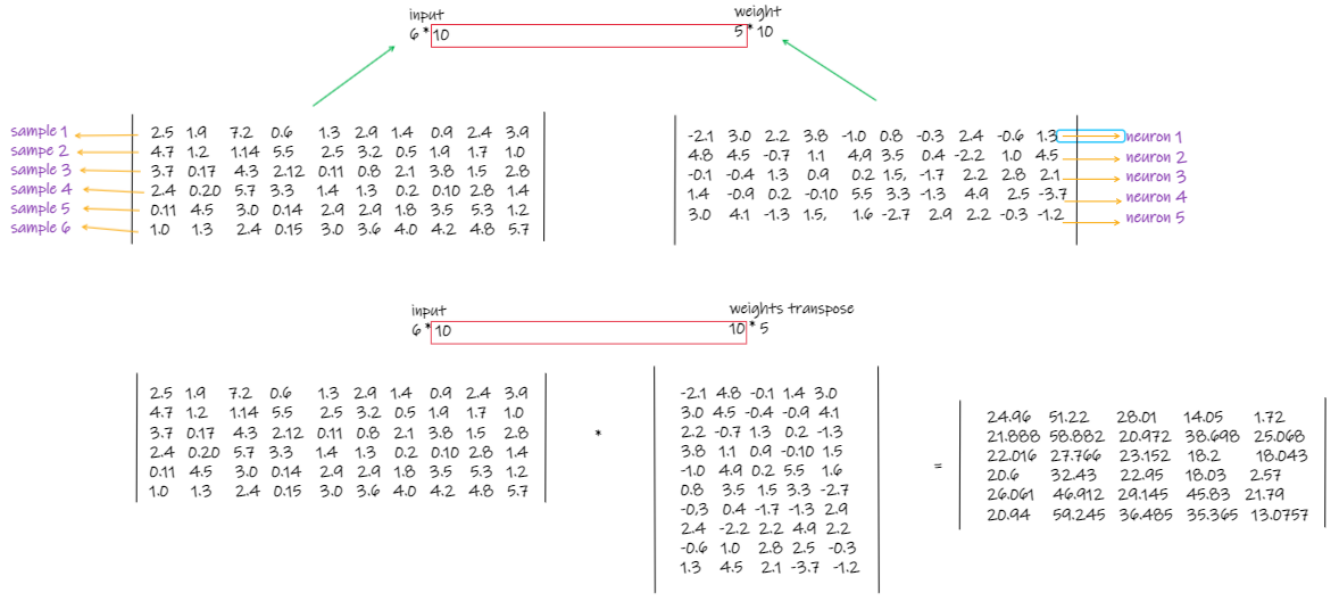
- Output

```
n Batch Input.py'
[[25.26  53.12  30.31  17.55   3.42 ]
 [22.188 60.782 23.272 42.198 26.768]
 [22.316 29.666 25.452 21.7   19.743]
 [20.9   34.33  25.25  21.53   4.27 ]
 [26.361 48.812 31.445 49.33  23.49 ]
 [21.24  61.145 38.785 38.865 14.775]]
PS C:\Users\Asus\Documents\vscode>
```

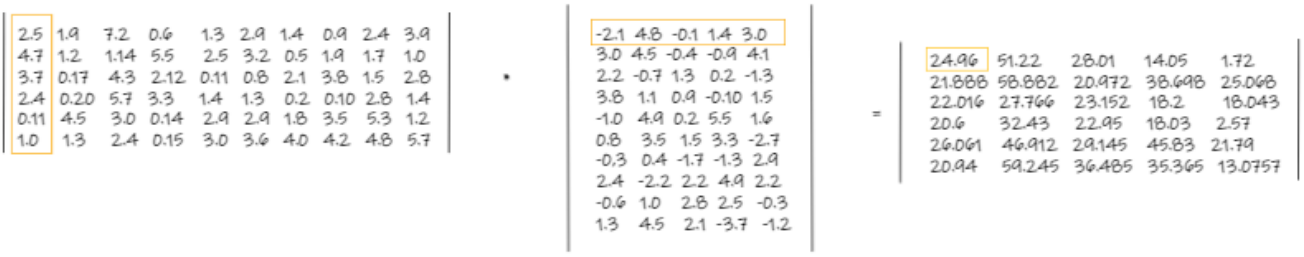
- Perhitungan output

```
#output
layer_outputs = np.dot(inputs, np.array(weights).T) + biases
```

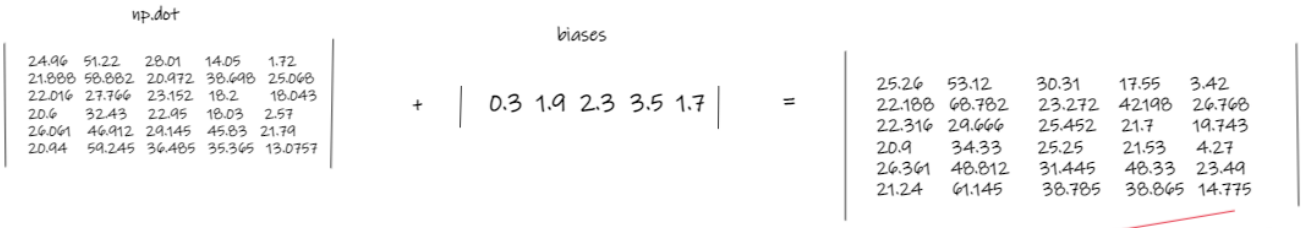
np.array (weight) T adalah mentranspose weight, dilakukan transpose pada weight karena untuk menyamakan ordo dengan input. Jika ordo tidak sama maka tidak akan dapat dihitung atau dioperasikan. Setelah weight di transpose maka operasi hitung dapat dilakukan dengan cara mengkali setiap baris input dengan kolom weight.



Cara menghitung np.dot adalah setiap baris input dikalikan dengan setiap baris kolom. Baris 1 input dikali kolom 1 weight, baris 1 input dikali kolom 2 weight dan seterusnya hingga selesai menghasilkan np.dot 6*5



Saat hasil np.dot dari perkalian input dan weight transpose keluar, np.dot ditambah dengan biases

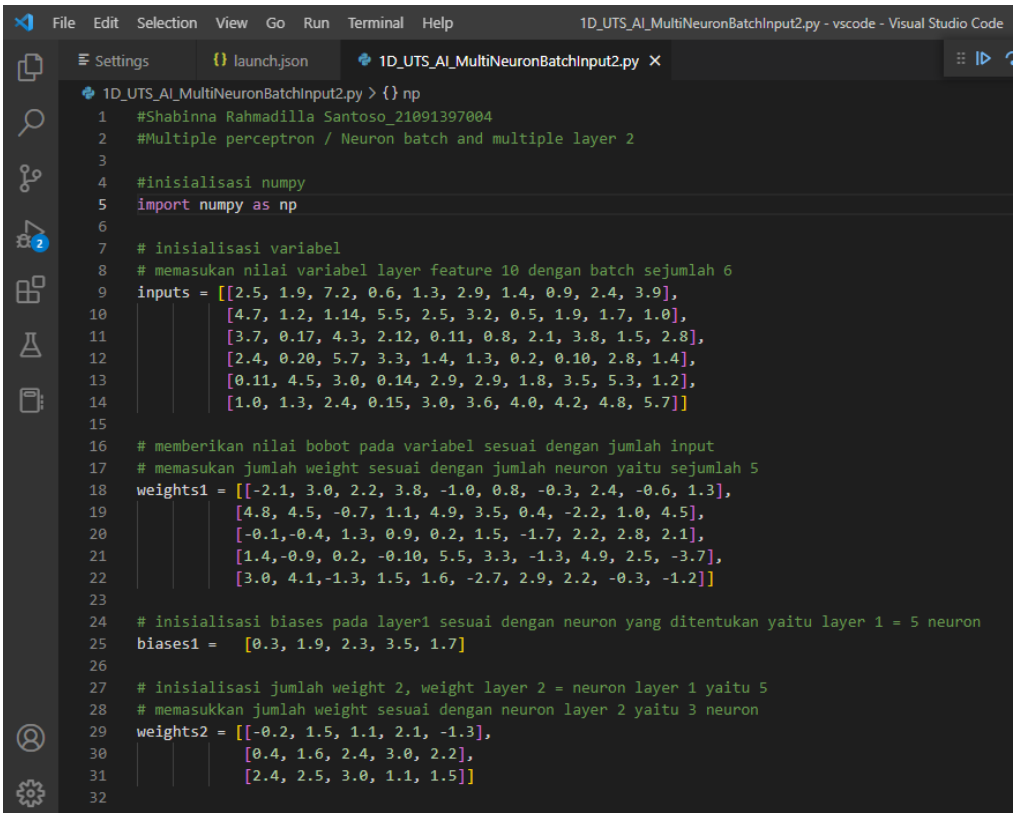


UTS 2

D. Multi Neuron Batch Input

Input layer feature 10, batchnya 6 input, hidden layer1 (5 neuron), hidden layer2 (3 neuron)

- Source Code




```
# inisialisasi biases pada layer2 dengan neuron yang ditentukan yaitu 3
biases2 =  [2.0, 1.0, 0.4]

# output
# menghitung layer1 dengan (inputs*weight1) dan biases1
layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1

# menghitung layer2 dengan hasil perhitungan pada layer1
layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2

#print output layer2
print(layer2_outputs)
```

- Line 4 menginisialisasikan numpy ke np guna memudahkan dalam pengoperasionalkan source code ini, proses komputasi angka
- Line 9 menginisialisasikan variabel, memasukkan nilai input dengan jumlah 10 baris angka sesuai dengan yang diminta yaitu feature layer 10 dan 6 kolom angka dengan perintah batch = 6

- Line 18 memasukan nilai weight dengan 5 kolom karena sesuai dengan neuron yang diminta yaitu 5 dan 10 baris angka
- Line 25 memasukkan nilai bias sejumlah dengan banyaknya neuron yang ditetapkan yaitu 5
- Line 29 memasukkan nilai weight2 dengan jumlah weight samadengan neuron atau ordo di layer1 yaitu 5 dengan 3 kolom sesuai dengan neuron layer2 yaitu 3
- Line 34 memasukkan nila biases sejumlah dengan banyaknya neuron yang ditetapkan di layer2 yaitu 3
- Line 38 melakukan perhitungan di layer1 dengan weight di transpose terlebih dahulu
- Line 41 melakukan perhitungan dari hasil layer1 dikali weight transpose ditambah biases2
- Line 44 untuk menampilkan hasil output dari source code tersebut

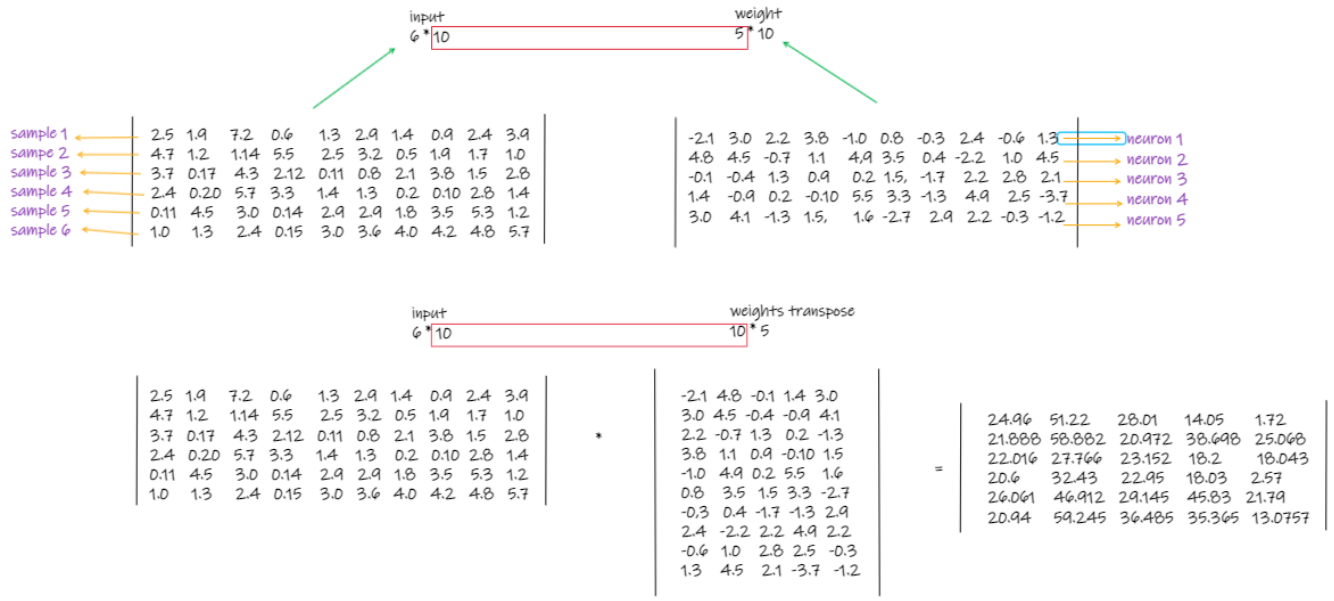
  - Output

```
[[142.378  229.014  309.189 ]
 [168.152  348.4628 361.992 ]
 [ 89.9371 227.0114 257.9639]
 [116.752  198.872  242.223 ]
 [177.5913 364.7796 369.5294]
 [194.542  349.512  385.5075]]
```

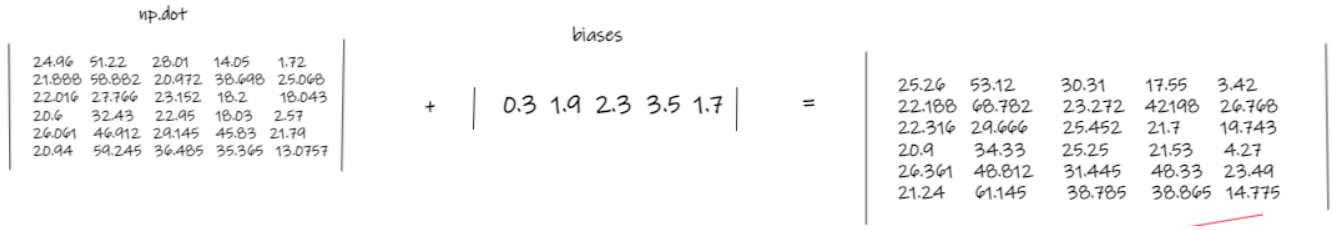
  - Perhitungan Output

```
# output
# menghitung layer1 dengan (inputs*weight1) dan biases1
layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
```

np.array (weight) T adalah mentranspose weight, dilakukan transpose pada weight karena untuk menyamakan ordo dengan input. Jika ordo tidak sama maka tidak akan dapat dihitung atau dioperasikan. Setelah weight di transpose maka operasi hitung dapat dilakukan dengan cara mengkali setiap baris input dengan kolom weight.



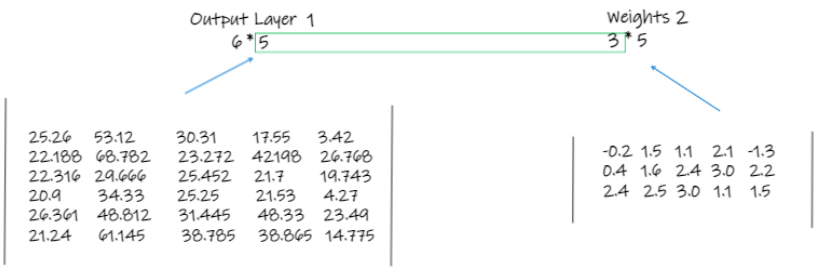
Saat hasil layer 1 sudah ditemukan dengan mengkali input dan transpose weight 1 hasil np.dot ditambah dengan biases 1

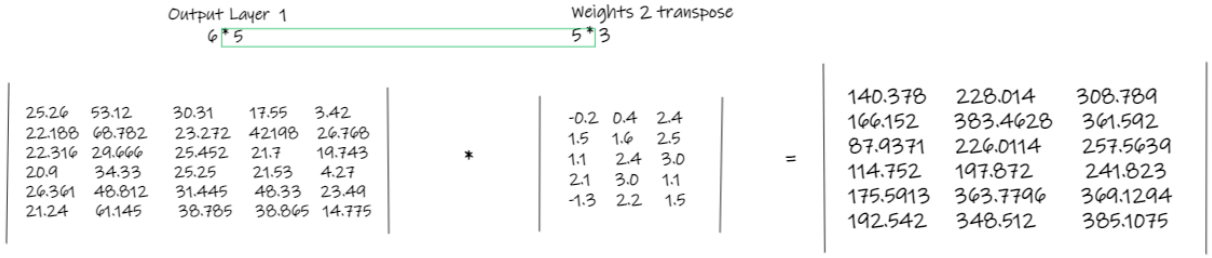


Menghitung hidden layer 2

```
# menghitung layer2 dengan hasil perhitungan pada layer1
layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
```

Saat hasil dari layer1 didapatkan lalu lanjut masuk ke penghitungan layer 2, mencari nilai np.dot layer 2 dengan cara hasil layer 1 dikali dengan weight transpose 2. Weights 2 di transpose untuk menyamakan ordo output hidden layer 1





Setelah di dapatkan hasil np.dot layer 2 lalu ditambah dengan biases 2 dan hasil layer 2 didapatkan