

2. Perceptron dengan Python

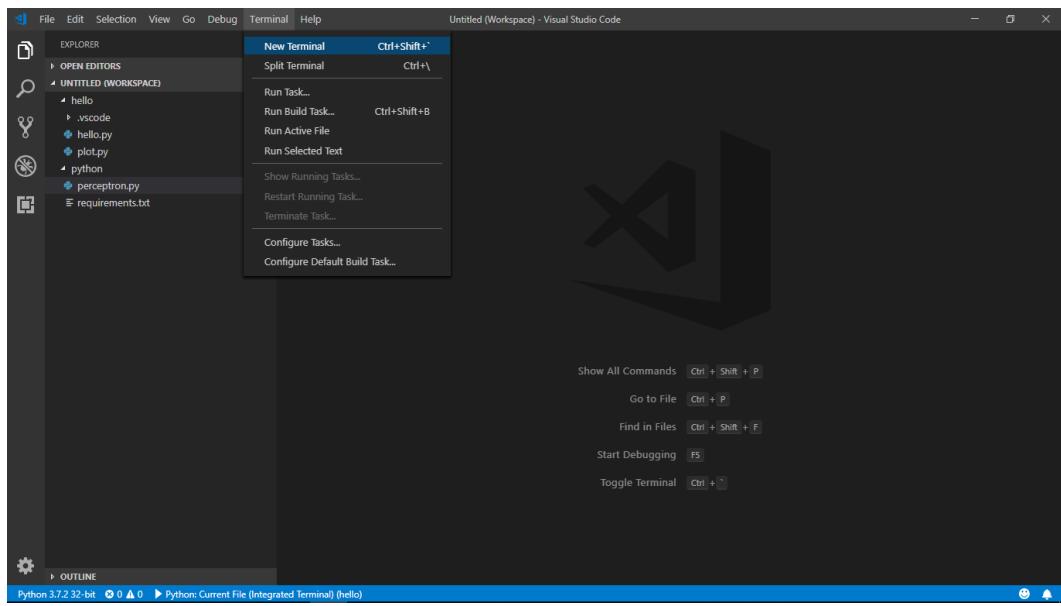
Perceptron juga termasuk salah satu bentuk jaringan saraf yang sederhana. Perceptron biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan pemisahan secara linear. Pada dasarnya, perceptron pada jaringan saraf dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang (threshold). Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Nilai threshold (θ) pada fungsi aktivasi adalah non negatif. Fungsiaktivasi ini dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negative.

Perceptron dilatih dengan menggunakan sekumpulan pola yang diberikan kepadanya secara berulang ulang selama latihan. Setiap pola yang diberikan merupakan pasangan pola masukan dan pola yang diinginkan sebagai target. Perceptron melakukan penjumlahan terhadap tiap tiap masukannya dan menggunakan fungsi ambang untuk menghitung keluarannya. keluaran ini kemudian dibandingkan dengan hasil yang diinginkan, perbedaan yang dihasilkan dari perbandingan ini digunakan untuk merubah bobot bobot dalam jaringan. Demikian dilakukan berulang ulang sampai dihasilkan keluaran yang sesuai dengan hasil yang diinginkan.

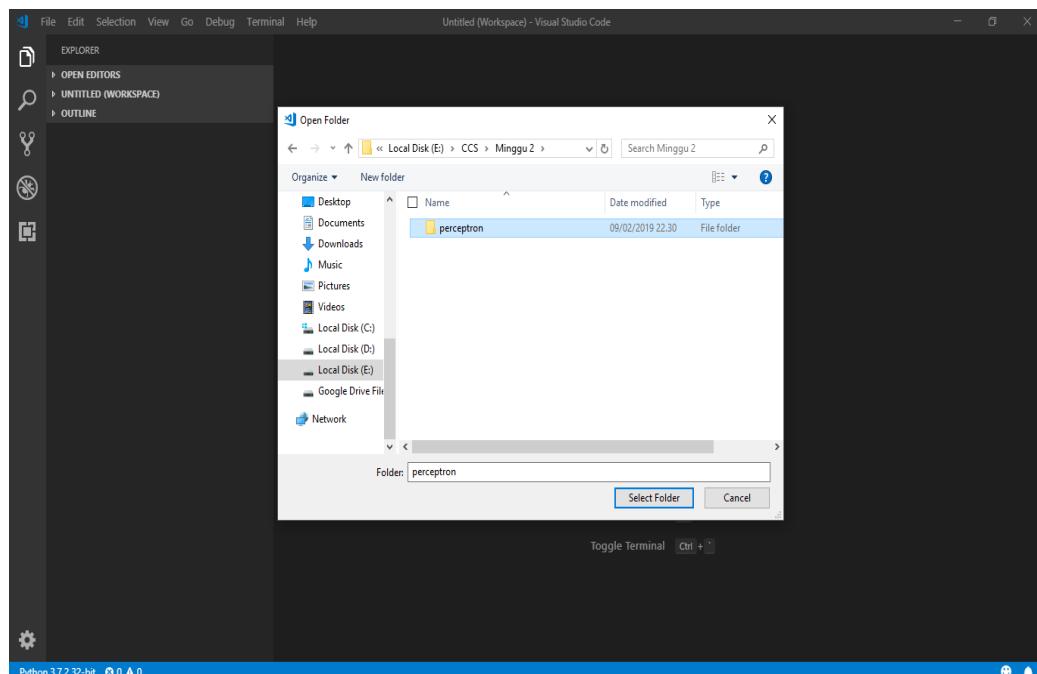
Pada materi kedua ini, akan dilakukan latihan penerapan konsep jaringan saraf tiruan secara sederhana pada fungsi logika OR dengan menggunakan algoritma perceptron pada python. Tahapan dalam dalam tersebut adalah sebagai berikut:

- a. Jalankan VSCode pada folder project (workspace)

Pertama elalui integrated terminal di VSCode atau melalui file explorer, buat folder kosong dengan nama “perceptron”, buka dan masuk ke folder tersebut, kemudian buka VSCode (code) dari folder tersebut.



Cara untuk Membuka Integrated Terminal di VSCode

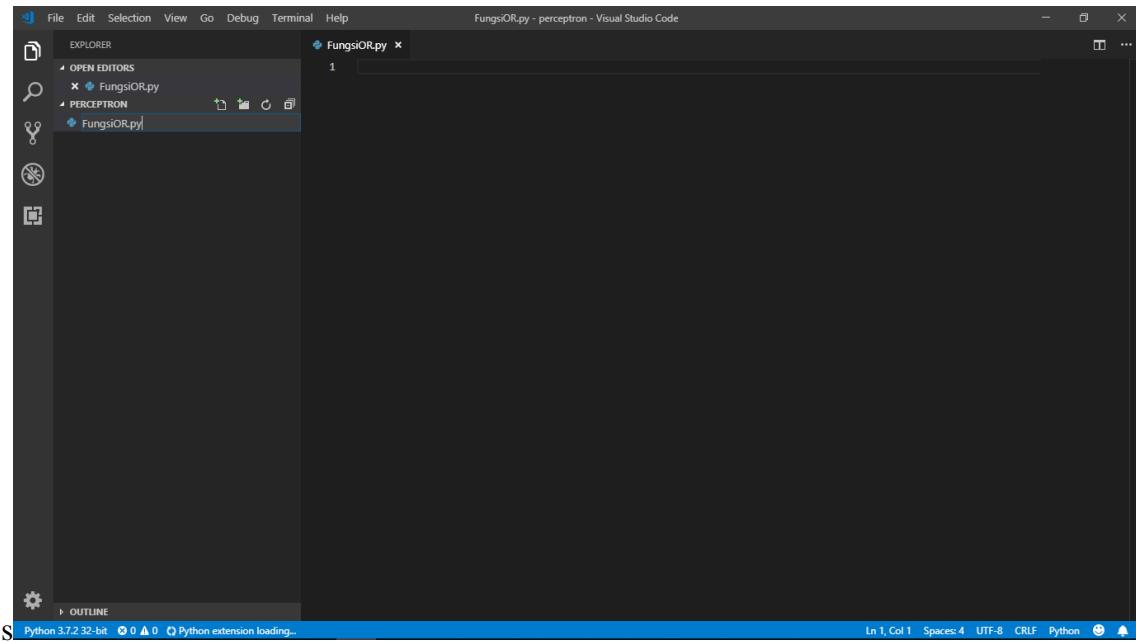


Cara Open Folder Dari VSCode dari Menu File>Open Folder

b. Buat File Source code perceptron Python

Dari toolbar File Explorer, klik tombol New File dari folder perceptron:

Beri nama fungsiOR.py, dan secara otomatis akan membuka editor baru.



Dengan menggunakan ekstensi .py, kita memberitahu VSCode untuk melakukan interpret file ini menggunakan program Python.

Langkah berikut yang dilakukan adalah memasukkan code berikut :

```
from random import choice
from numpy import array, dot, random
from matplotlib import pyplot as plt

# fungsi matematika
unit_step = lambda y: 0 if y < 0 else 1

# dua entri pertama pada array adalah nilai inputan, entri ketiga pada array
# adalah nilai bias/threshold
data_training = [ # data training adalah fungsi OR
    (array([0, 0, 1]), 0),
    (array([0, 1, 1]), 1),
    (array([1, 0, 1]), 1),
    (array([1, 1, 1]), 1),
]
# pemilihan 3 angka acak antara 0 dan 1 sebagai initial weight
```

```
w = random.rand(3)
errors = [] # menyimpan nilai error untuk dapat di visualisai
teta = 0.2 # variabel pengontrol learning rate
n = 100 # jumlah banyaknya learning iterations

# training process
for i in range(n):
    y, expected = choice(data_training)
    result = dot(w, y)
    error = expected - unit_step(result)
    errors.append(error)
    w += teta * error * y # learning rule

# Hasil pembelajaran (fungsi logika OR)
for y, _ in data_training:
    result = dot(y, w)
    print("{}: {} -> {}".format(y[:2], result, unit_step(result)))

# Visualisasi proses training (plot errors)
plt.ylim([-1, 1])
plt.plot(errors)
plt.show()
```

c. Menjalankan Program

Untuk menjalankan fungsiOR.py, klik kanan pada editor dan pilih Run Python File in the terminal (yang kemudian akan menyimpan file secara otomatis).

```
12     (array([1,0,1]), 1),
13     (array([1,1,1]), 1),
14 ]
15
16 w = random.rand(3) #pemilihan 3 angka acak antara 0 dan 1 sebagai initial weight
17 errors = [] #menyimpan nilai error untuk dapat di visualisasi
18 teta = 0.2 #variabel pengontrol learning rate
19 n = 100 #jumlah banyaknya learning iterations
20
21 #training process
22 for i in range(n):
23     y, expected = choice((0,1), 2, p=[0.5, 0.5])
24     result = dot(w, y)
25     error = expected - uni(result)
26     errors.append(error)
27     w += teta * error * y
28
29 #Hasil pembelajaran (fungsiOR)
30 for y, _ in data_training:
31     result = dot(y, w)
32     print("{}: {} -> {}".format(y, result, int(result)))
33
34 # Visualisasi proses training
35 plt.ylim([-1,1])
36 plt.plot(errors)
37 plt.show()
```

Cara running program

Perintah tersebut membuka panel Terminal dimana Python secara otomatis diaktifkan, kemudian menjalankan fungsiOR.py dan akan memberikan hasil output seperti pada gambar dibawah berikut ini

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS E:\CCS\Minggu 2\perceptron & C:/Users/Asus/AppData/Local/Programs/Python/Python37-32/python.exe "e:/CCS/Minggu 2/perceptron/FungsiOR.py"
[0 0]: -0.14861369491335896 -> 0
[0 1]: 0.589610885274479 -> 1
[1 0]: 0.74952004855314 -> 1
[1 1]: 1.4877955519939468 -> 1
```

Hasil Output fungsiOR.py

Dan secara bersamaan, jendela plot akan muncul dengan tampilan keluarannya seperti berikut.

