

Praktikum Sistem Cerdas



NRP	: 3223600019
Nama	: Muhammad Bimo Fachrizky
Materi	: Membuat Program dengan Visual C++ dan Python
Tanggal	: Senin, 17 Februari 2025

Praktikum 1

Membuat Program dengan Visual C++ dan Python

I. Tujuan Pembelajaran

- Mahasiswa dapat melakukan instalasi MS Visual C++
- Mahasiswa dapat melakukan instalasi PyCharm
- Mahasiswa dapat membuat aplikasi dengan MS Visual C++ dan PyCharm

Software yang di perlukan

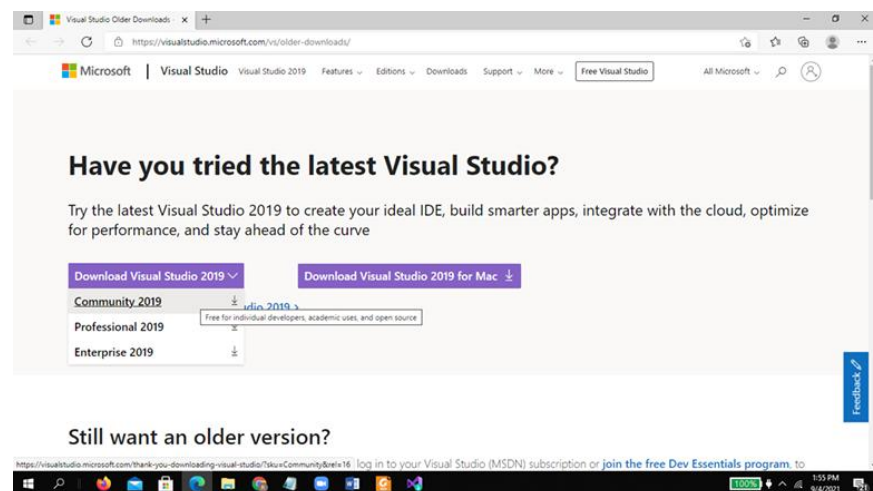
- Microsoft Visual C++
- PyCharm

II. Langkah percobaan

1. Instalasi MS Visual C++

Masukkan CD MS Visual C++ (pada buku ini digunakan MS Visual C++ 2010) untuk instalasi, atau kunjungi website berikut dan ikuti Microsoft Guide untuk menginstal Visual C++.

<https://visualstudio.microsoft.com/vs/older-downloads/>

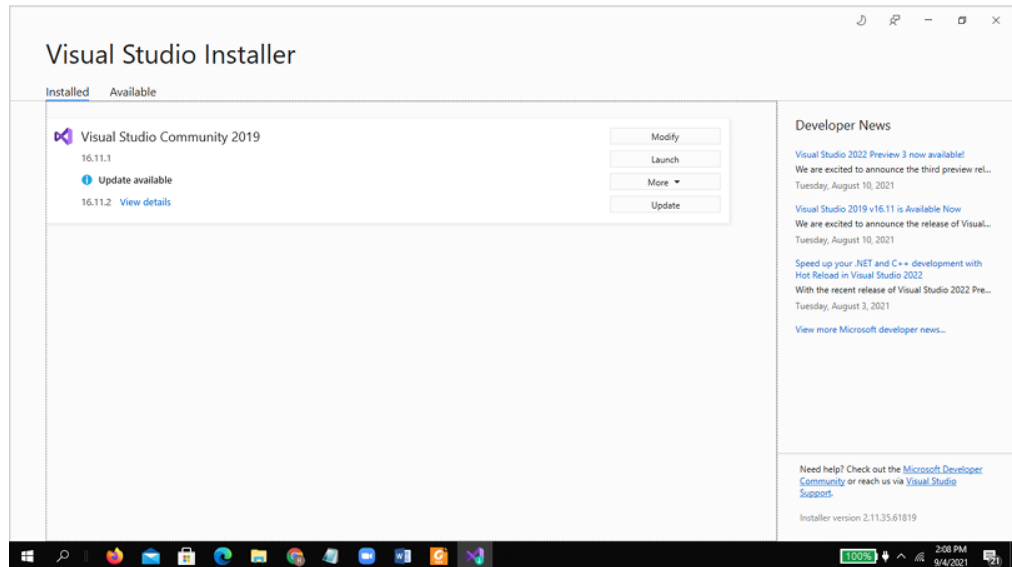


Gambar 1. Download Visual C++

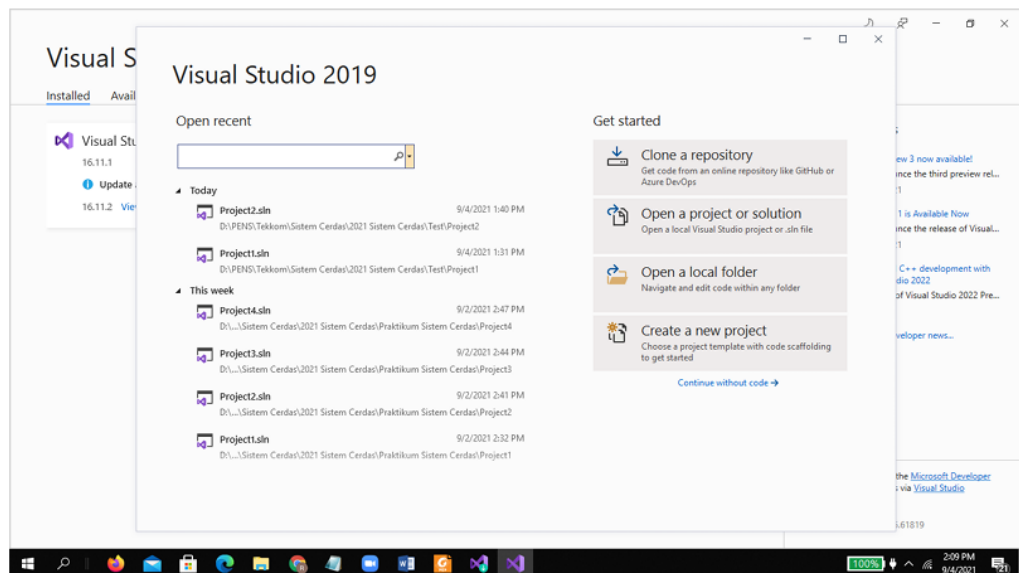
Pilih Download Visual Studi 2019 (untuk window)

Pilih Community 2019

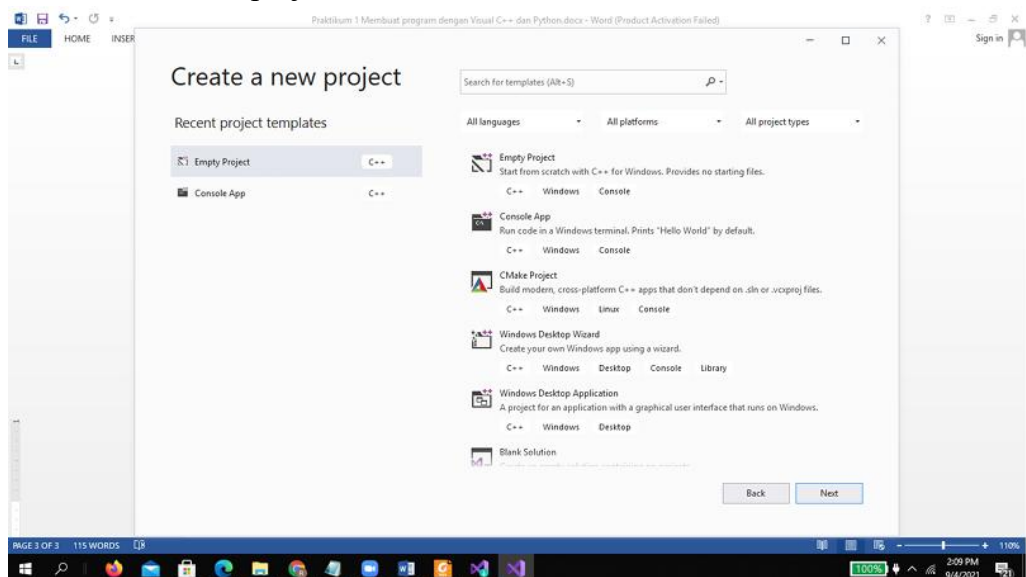
2. Run MS Visual C++



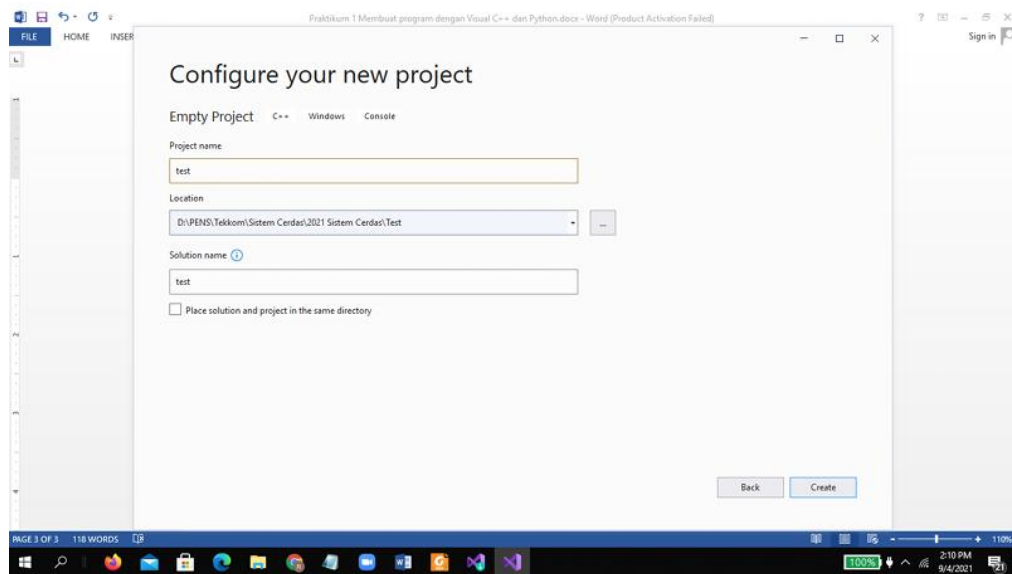
Pilih Launch



Pilih Create a new project

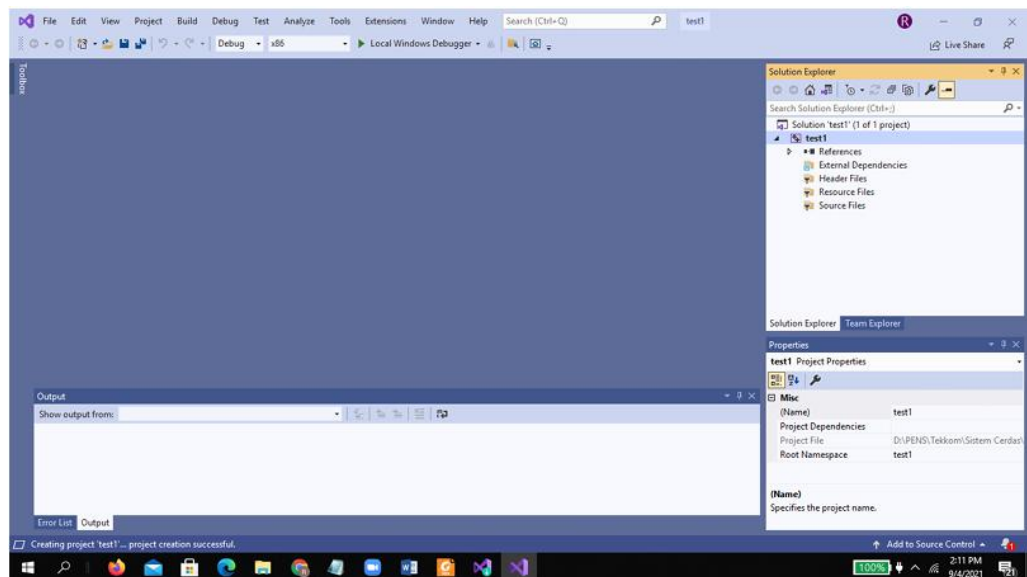


Pilih Empty Project

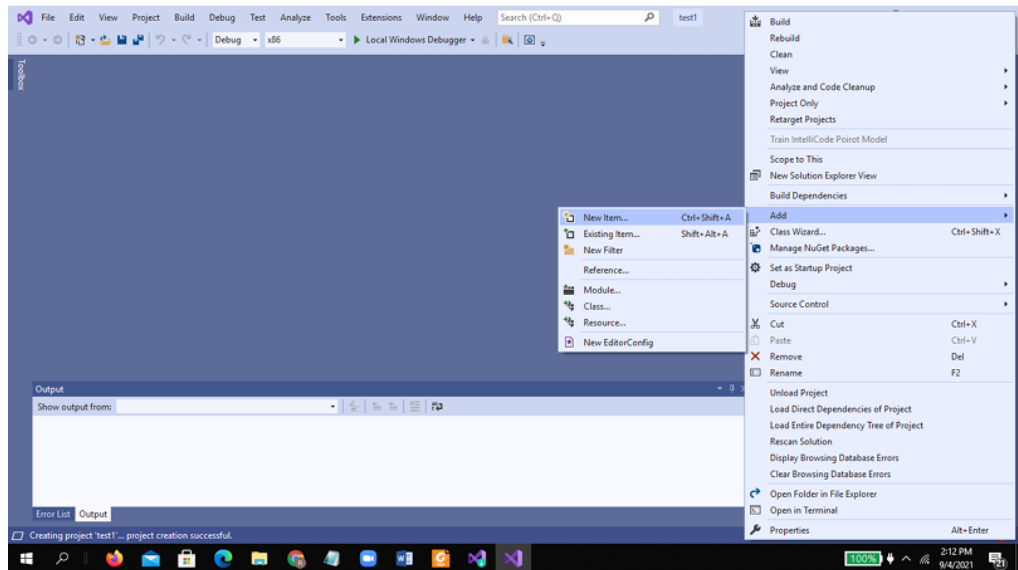


Isi project name dengan misalnya: test1

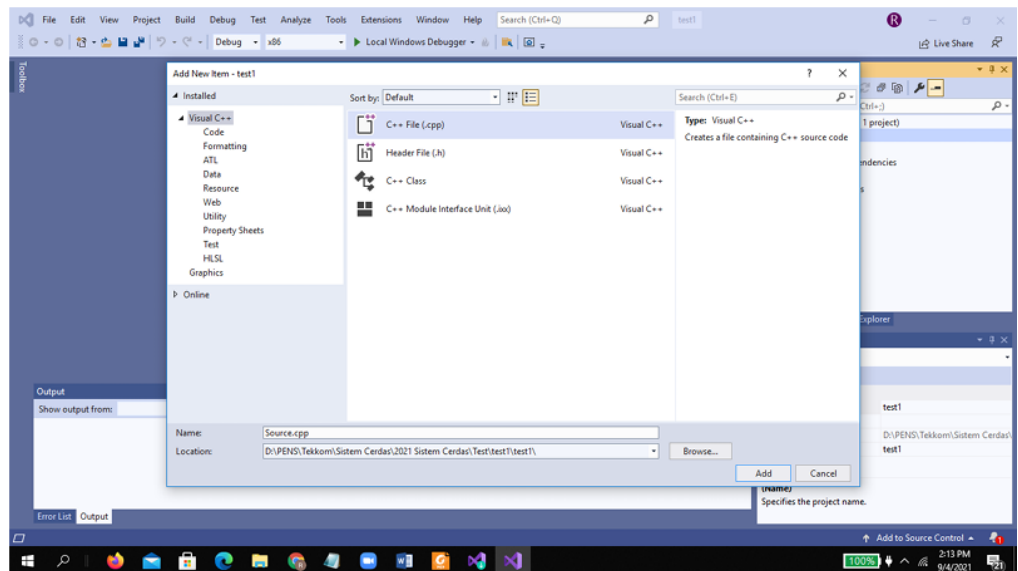
Pilih Create



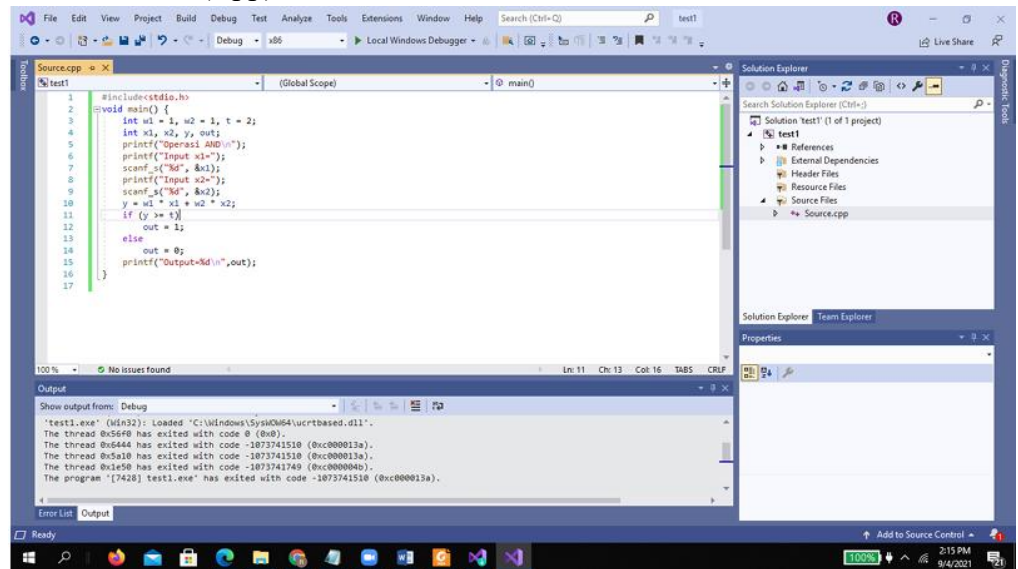
Klik kanan pada test 1



Pilih Add->New Item



Pilih C++ File (.cpp)



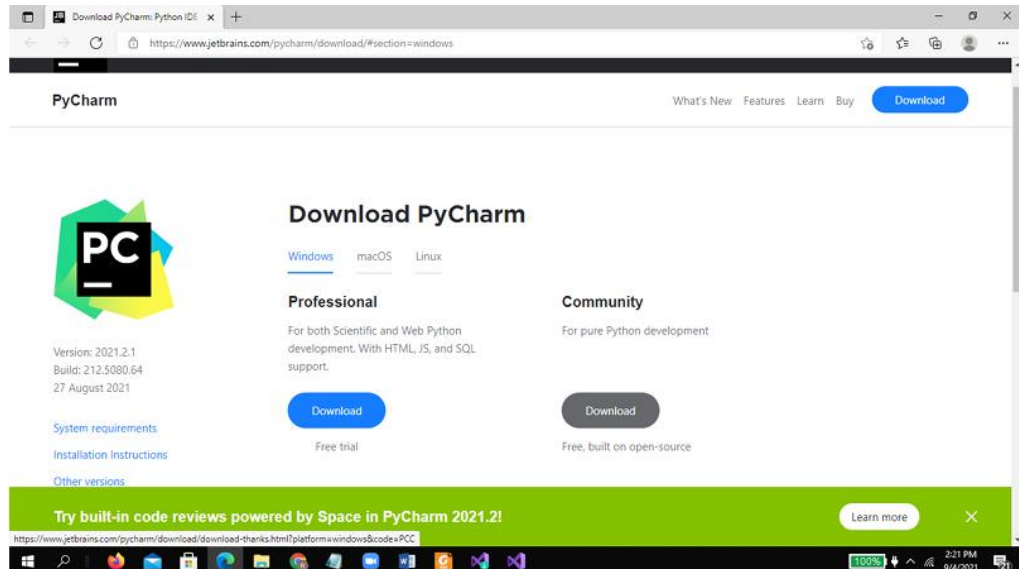
Buat program seperti diatas

Untuk Run -- Pilih Local Windows Debugger

3. Instalasi PyCharm di windows

Download installer .exe in

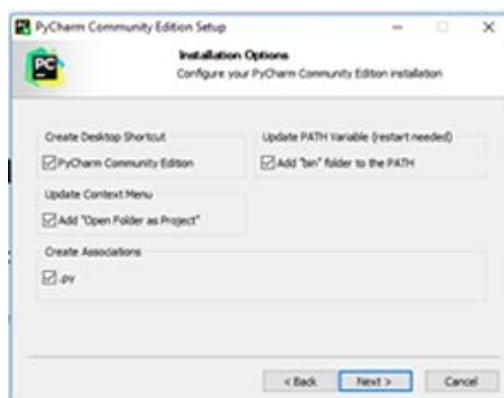
<https://www.jetbrains.com/pycharm/download/#section=windows>. Pilih versi Community. Ukuran file sekitar 366 MB.

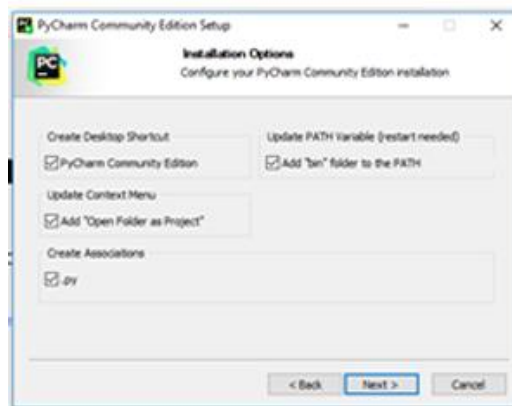


Pilih Community (tombol Download)

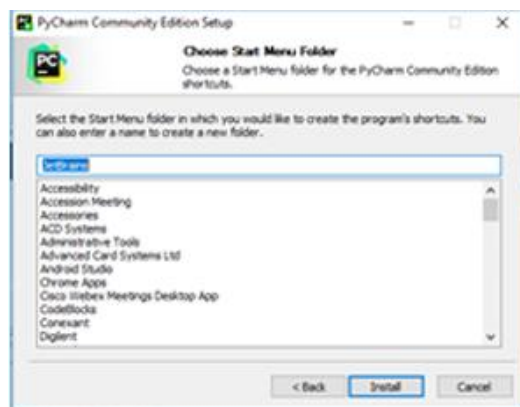


Pilih Next





Pilih semua dan Next

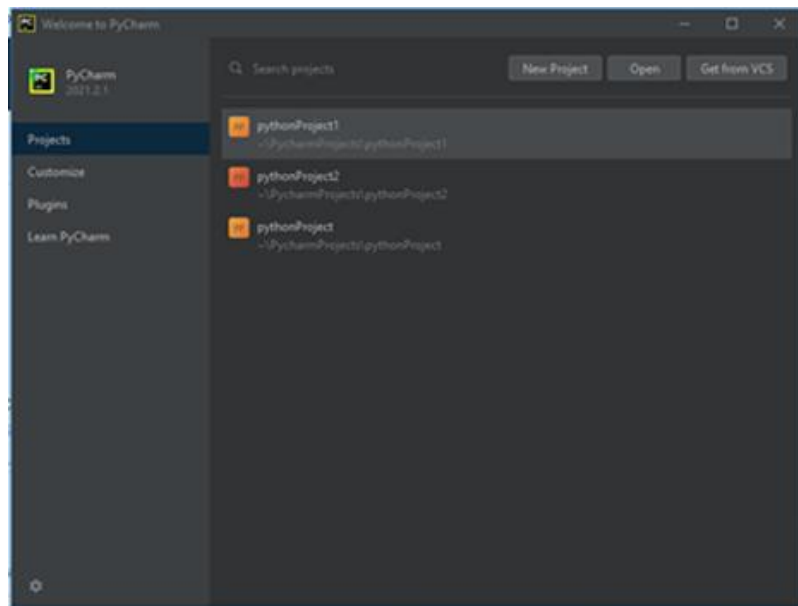


Pilih Install

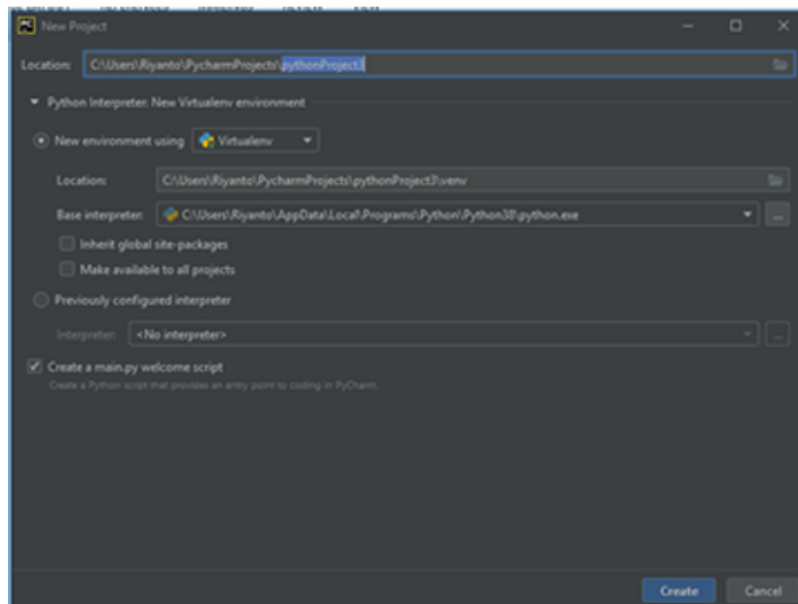


Pilih Finish

Run PyCharm

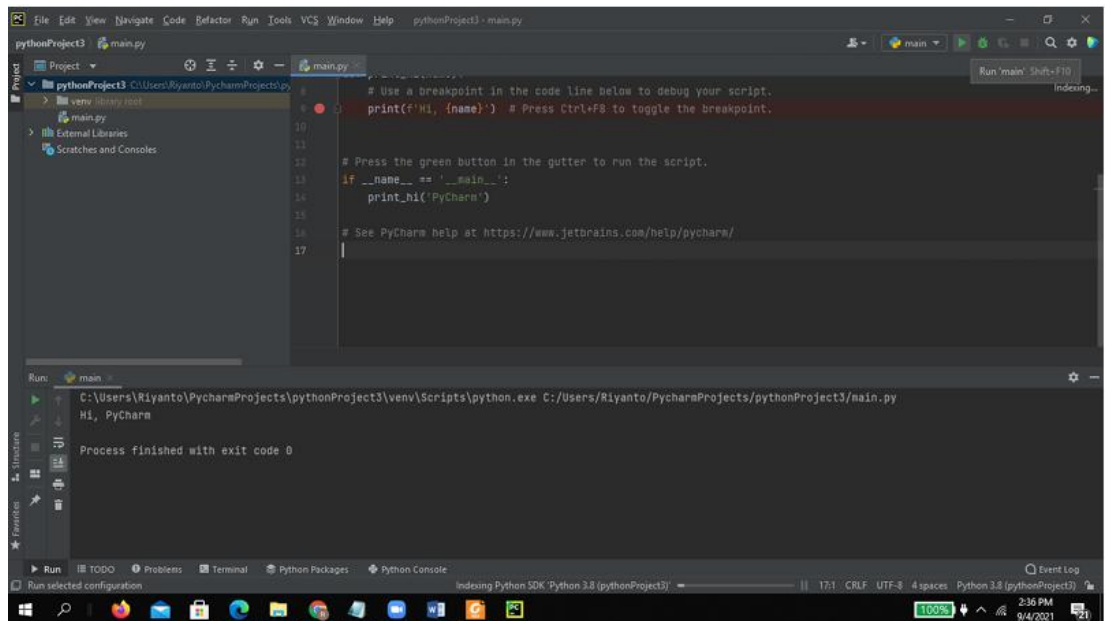


Pilih New Project

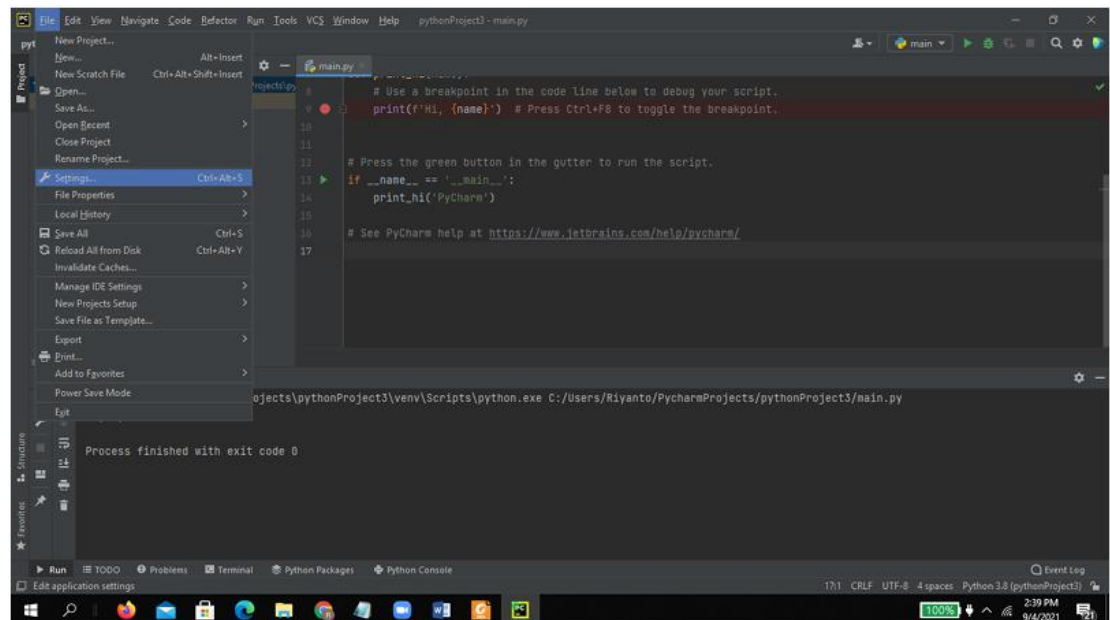


Pilih Create

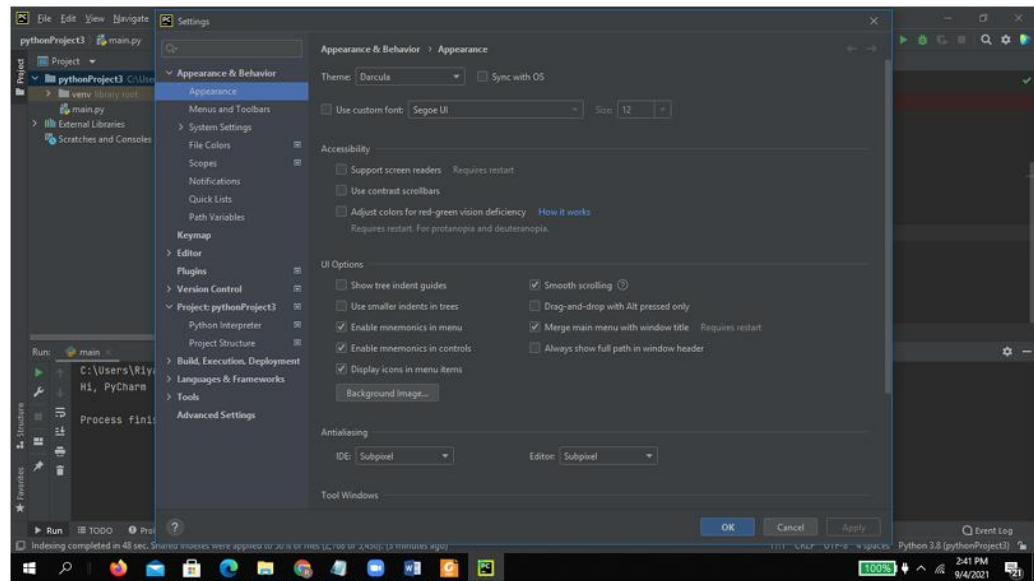
Pilih Run 'main Shift + F10



Setting



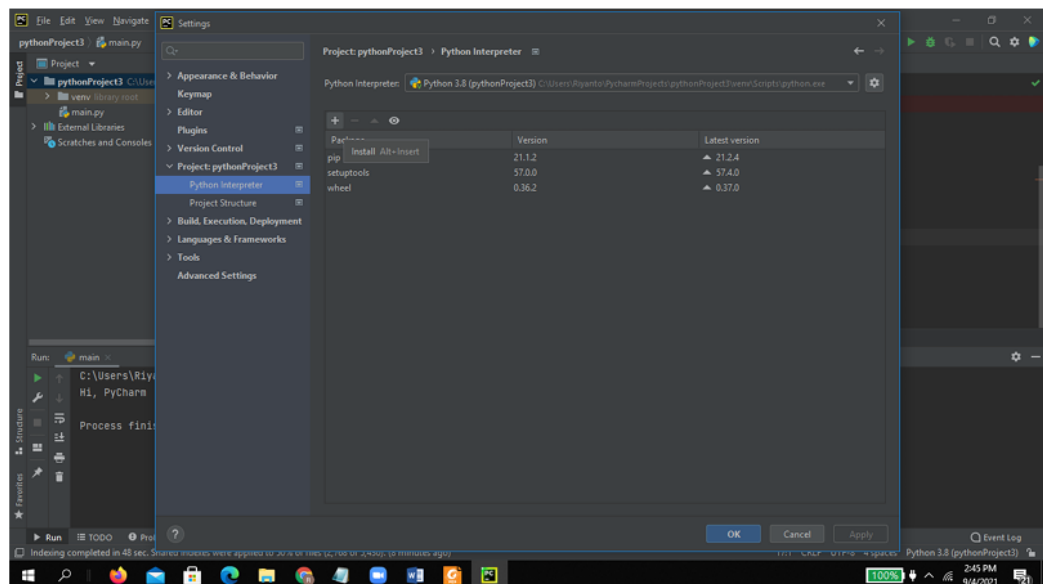
Pilih File Setting



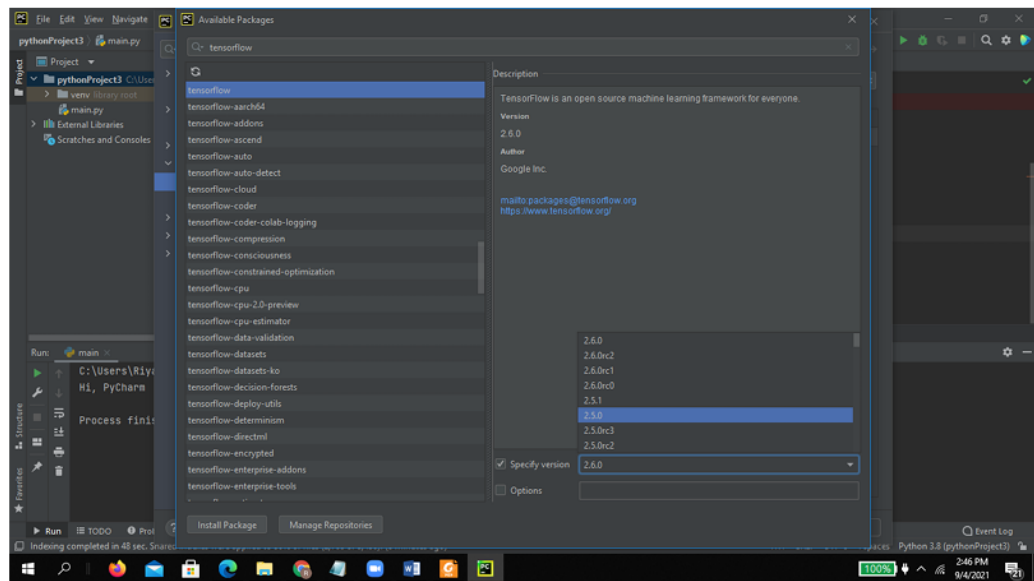
Pilih Project -> Python interpreter

Install library dibawah ini

- tensorflow==2.5.0
- tensorflow-datasets==4.3.0
- Pillow==8.2.0
- pandas==1.2.4
- numpy==1.19.5
- scipy=1.7.0

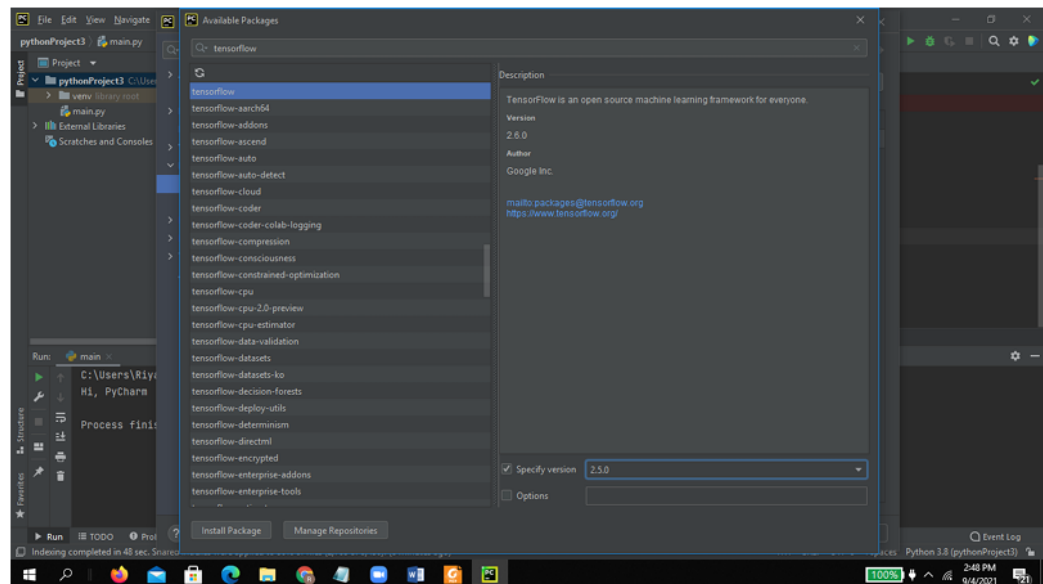


Pilih + (Install)

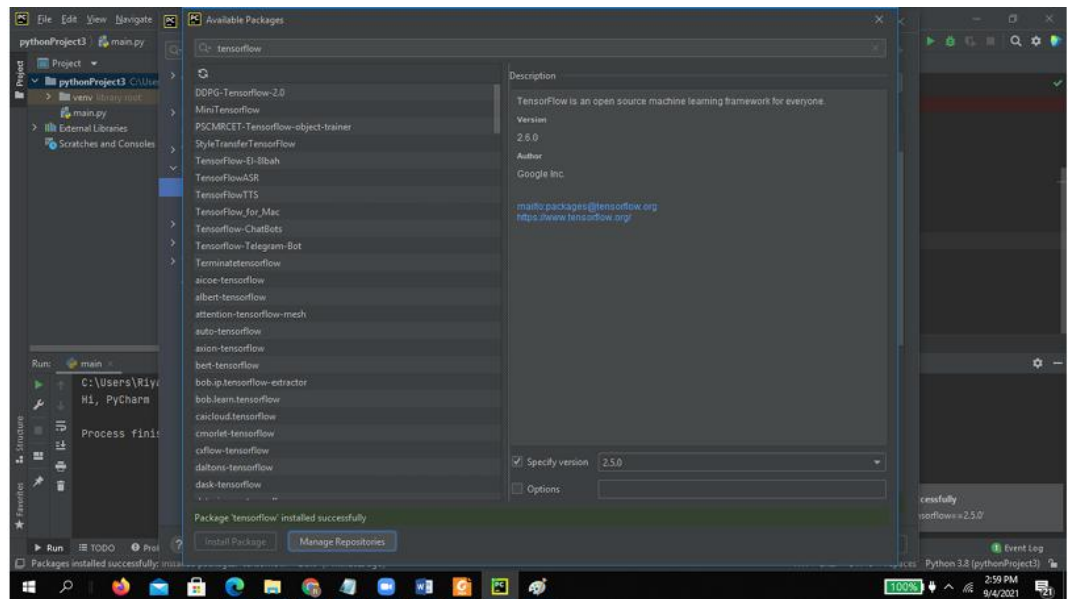


Cari tensorflow 2.5.0

Pilih Specify version dan pilih 2.5.0

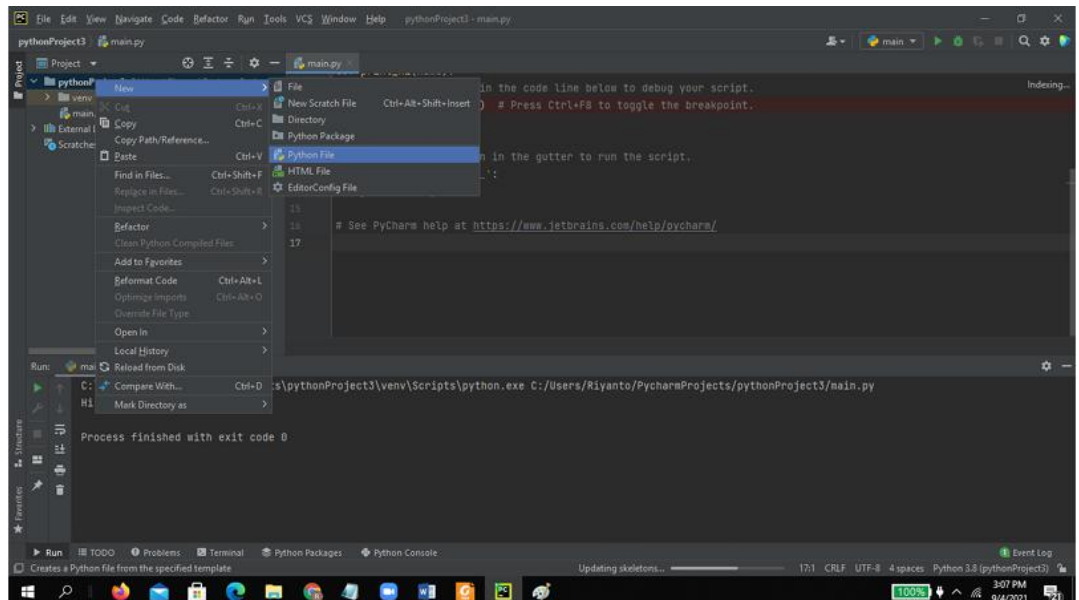


Install Package

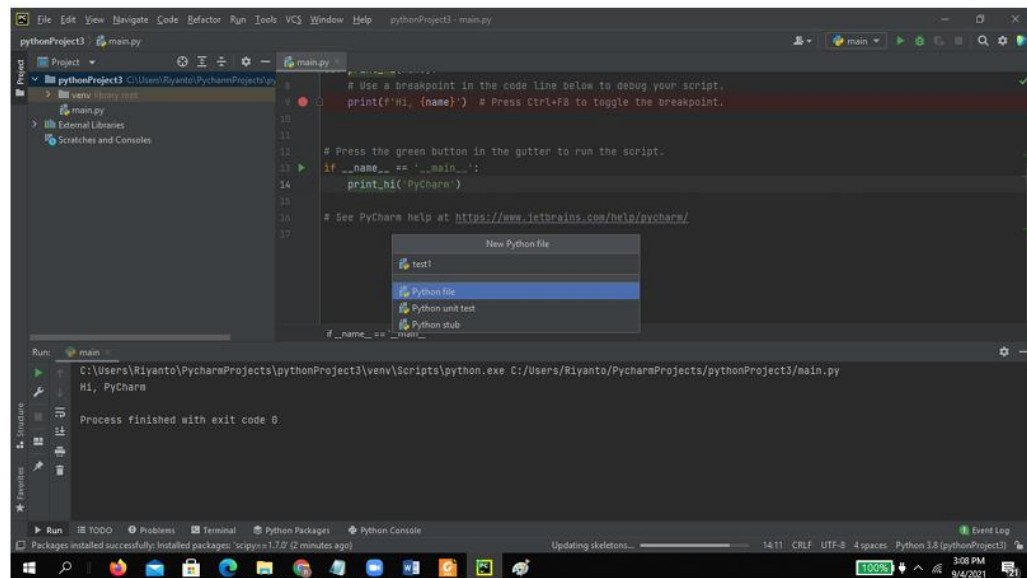


Install Sukses

Run program Python



Pilih New – Python File



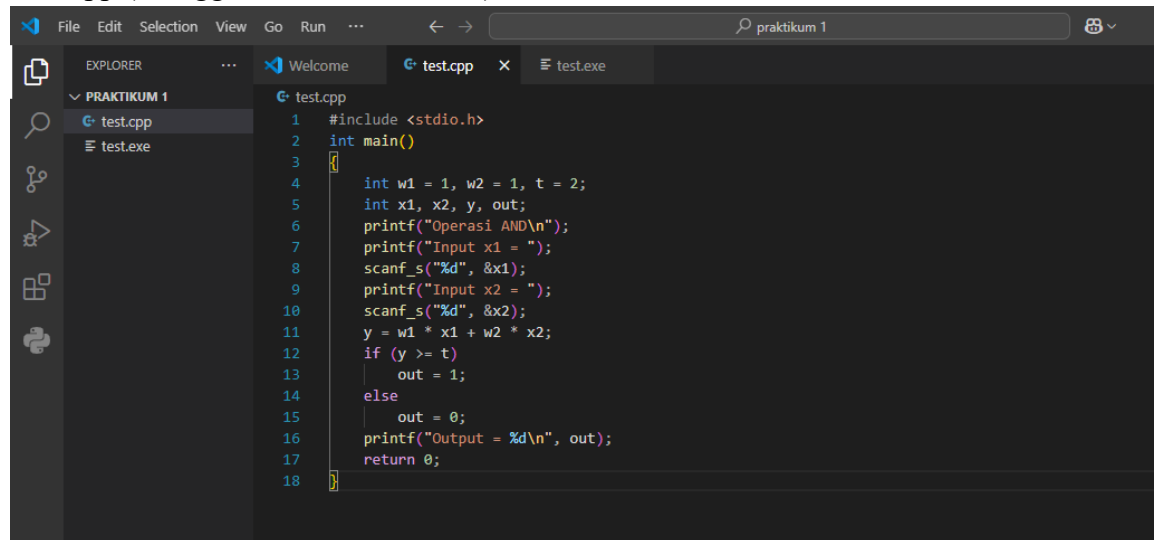
Isi test1

Contoh program : simple_regression.py

```
1 import tensorflow as tf
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 from tensorflow import keras
6
7
8 def plot_data(x_data, y_data):
9     fig, ax = plt.subplots()
10
11     ax.plot(x_data, y_data, 'ro')
12
13     plt.pause(1)
14
15     return None
16
17
18 if __name__ == "__main__":
19     # Define and compile the neural networks
20     model = tf.keras.Sequential(
21         [keras.layers.Dense(units=1, input_shape=[1])])
22     model.compile(optimizer="sgd", loss="mean_squared_error")
23
24     # Providing the data
25     xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0])
26     ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0]) # y = 2x - 1
27
28     # Plotting data will help us to understand faster
29     # visually about the data
30     plot_data(xs, ys)
31
32     # Train the neural network
33     model.fit(xs, ys, epochs=500)
34
35     # Predict using the model
36     print(model.predict([10.0])) # this result should be close to 19.
```

III. Hasil Percobaan

- Test.cpp (Menggunakan Visual Code)



```
1 #include <stdio.h>
2 int main()
3 {
4     int w1 = 1, w2 = 1, t = 2;
5     int x1, x2, y, out;
6     printf("Operasi AND\n");
7     printf("Input x1 = ");
8     scanf_s("%d", &x1);
9     printf("Input x2 = ");
10    scanf_s("%d", &x2);
11    y = w1 * x1 + w2 * x2;
12    if (y >= t)
13        out = 1;
14    else
15        out = 0;
16    printf("Output = %d\n", out);
17    return 0;
18 }
```

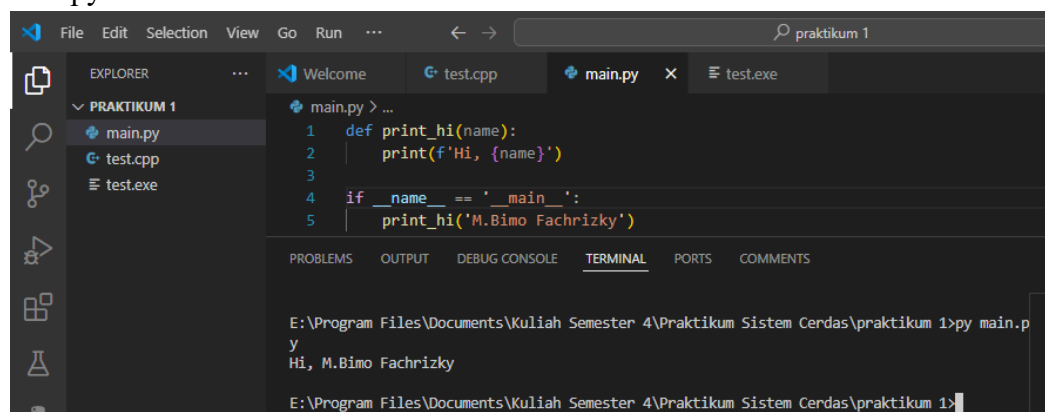
```
E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 1>test
Operasi AND
Input x1 = 1
Input x2 = 1
Output = 1

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 1>test
Operasi AND
Input x1 = 1
Input x2 = 0
Output = 0

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 1>test
Operasi AND
Input x1 = 0
Input x2 = 1
Output = 0

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 1>test
Operasi AND
Input x1 = 0
Input x2 = 0
Output = 0
```

- Test.py



```
1 def print_hi(name):
2     print(f'Hi, {name}')
3
4 if __name__ == '__main__':
5     print_hi('M.Bimo Fachrizky')
```

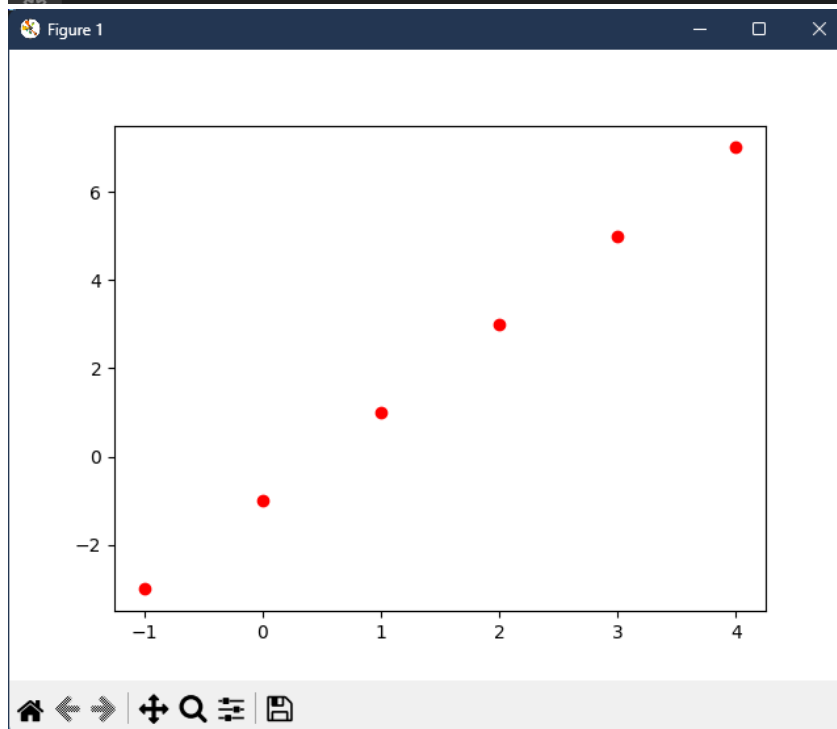
PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS

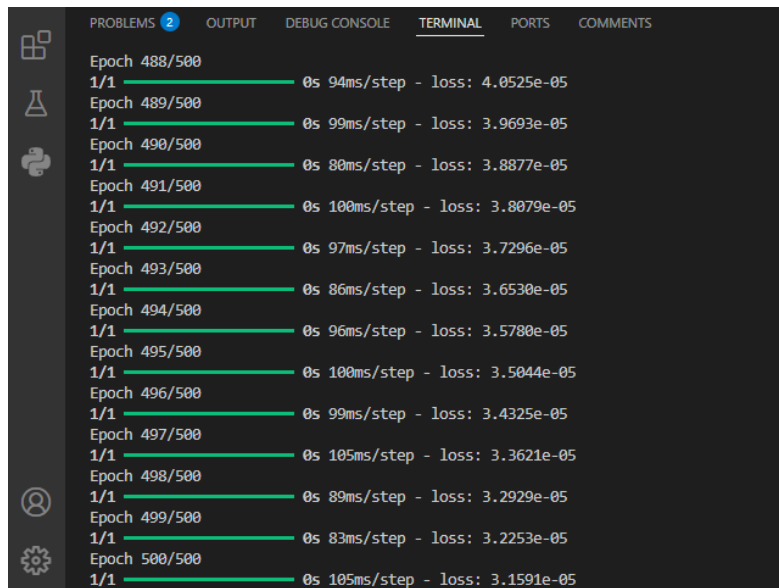
```
E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 1>py main.p
y
Hi, M.Bimo Fachrizky

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 1>
```

- Simple_regression.py

```
File Edit Selection View Go Run ... < > praktikum 1
Welcome test.cpp main.py simple_regression.py 2 test.exe
simple_regression.py > ...
1 import tensorflow as tf
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from tensorflow import keras
5
6 def plot_data(x_data, y_data):
7     """Fungsi untuk menampilkan plot data input."""
8     fig, ax = plt.subplots()
9     ax.plot(x_data, y_data, 'ro')
10    plt.pause(1)
11
12 if __name__ == "__main__":
13     # Definisikan dan kompilasi model neural network
14     model = tf.keras.Sequential([
15         keras.layers.Dense(units=1, input_shape=[1])
16     ])
17     model.compile(optimizer="sgd", loss="mean_squared_error")
18
19     # Data input
20     xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0])
21     ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0]) # y = 2x - 1
22
23     # Tampilkan plot data
24     plot_data(xs, ys)
25
26     # Melatih neural network
27     model.fit(xs, ys, epochs=500)
28
29     # Prediksi dengan model
30     print(model.predict([10.0])) # Hasil seharusnya mendekati 19.0
```





IV. Analisa

Praktikum di atas melakukan sebuah percobaan yaitu menginstall software yang akan digunakan pada praktikum sistem cerdas, diantaranya ada Visual Studio, dan Pycharm. Namun pada percobaan ini menggunakan visual code untuk hasil dikarenakan visual code lebih fleksibel. Pada Program pertama ini merupakan implementasi sederhana dari gerbang logika AND menggunakan pendekatan perceptron.

```
#include <stdio.h>
int main()
{
    int w1 = 1, w2 = 1, t = 2;
    int x1, x2, y, out;
    printf("Operasi AND\n");
    printf("Input x1 = ");
    scanf_s("%d", &x1);
    printf("Input x2 = ");
    scanf_s("%d", &x2);
    y = w1 * x1 + w2 * x2;
    if (y >= t)
        out = 1;
    else
        out = 0;
    printf("Output = %d\n", out);
    return 0;
}
```


Program meminta dua input biner dari pengguna, yaitu x_1 dan x_2 , yang kemudian dikalikan dengan $w_1 = 1$ dan $w_2 = 1$. Hasil perkalian ini dijumlahkan menjadi nilai y , yang kemudian dibandingkan dengan threshold $t = 2$. Jika nilai y lebih besar atau sama dengan t , maka output yang dihasilkan adalah 1, sedangkan jika lebih kecil, outputnya adalah 0.

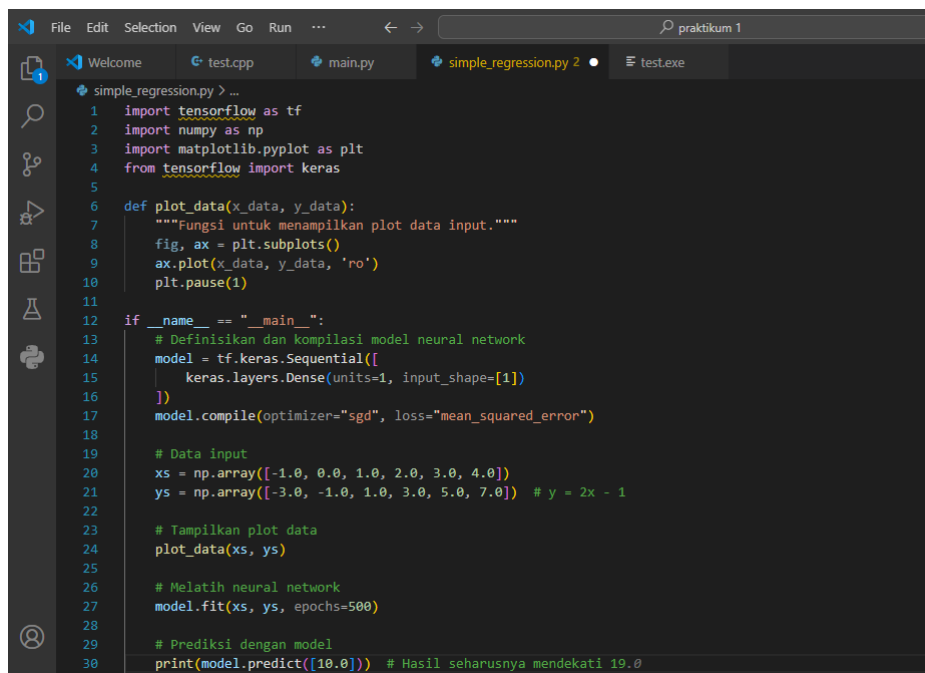
Pada program yang kedua adalah Kode Python

```
def print_hi(name):
    print(f'Hi, {name}')

if __name__ == '__main__':
    print_hi('M.Bimo Fachrizky')
```

Program ini mendefinisikan sebuah fungsi bernama `print_hi(name)`, yang menerima satu parameter `name` dan mencetak pesan "Hi, {name}" menggunakan f-string untuk formatting string secara langsung. Kemudian, pada bagian `if __name__ == '__main__':`, program memanggil fungsi `print_hi('M.Bimo Fachrizky')`, yang berarti ketika program dijalankan langsung (bukan diimpor sebagai modul), maka akan mencetak "Hi, M.Bimo Fachrizky" ke terminal. Program ini sebenarnya digunakan untuk mencoba Python setelah menginstallnya.

Program ketiga merupakan implementasi sederhana dari neural network menggunakan TensorFlow untuk menyelesaikan masalah regresi linear.



```
File Edit Selection View Go Run ... praktikum 1
Welcome test.cpp main.py simple_regression.py 2 test.exe
simple_regression.py > ...
1 import tensorflow as tf
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from tensorflow import keras
5
6 def plot_data(x_data, y_data):
7     """Fungsi untuk menampilkan plot data input."""
8     fig, ax = plt.subplots()
9     ax.plot(x_data, y_data, 'ro')
10    plt.pause(1)
11
12 if __name__ == "__main__":
13     # Definisikan dan kompilasi model neural network
14     model = tf.keras.Sequential([
15         keras.layers.Dense(units=1, input_shape=[1])
16     ])
17     model.compile(optimizer="sgd", loss="mean_squared_error")
18
19     # Data input
20     xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0])
21     ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0]) # y = 2x - 1
22
23     # Tampilkan plot data
24     plot_data(xs, ys)
25
26     # Melatih neural network
27     model.fit(xs, ys, epochs=500)
28
29     # Prediksi dengan model
30     print(model.predict([10.0])) # Hasil seharusnya mendekati 19.0
```

Program dimulai dengan mengimpor pustaka yang diperlukan, termasuk TensorFlow untuk membangun model, NumPy untuk manipulasi array, dan Matplotlib untuk menampilkan data input. Fungsi `plot_data()` digunakan untuk memvisualisasikan pasangan data (x, y) , yang mengikuti pola $y = 2x - 1$, sebelum proses pelatihan model dimulai. Setelah itu, model dilatih dengan data x_s dan y_s selama 500 epoch, di mana proses ini bertujuan untuk menemukan bobot dan bias optimal agar output model mendekati fungsi linear yang diharapkan. Setelah pelatihan selesai, model digunakan untuk memprediksi nilai y ketika $x = 10$, di mana hasil yang diharapkan adalah sekitar 19, sesuai dengan persamaan $y = 2(10) - 1$.

V. Kesimpulan

Ketiga program menunjukkan pendekatan berbeda dalam pemrosesan data dan pembelajaran mesin. Program pertama mengimplementasikan operasi logika AND, dimana dua input biner dikalikan dengan bobot tetap dan dibandingkan dengan nilai threshold untuk menentukan output. Program kedua merupakan contoh dasar dalam pemrograman Python yang mencetak teks dinamis berdasarkan input nama menggunakan f-string. Program ketiga menggunakan neural network berbasis TensorFlow untuk menyelesaikan regresi linear dengan mempelajari pola hubungan antara input dan output melalui pelatihan menggunakan Stochastic Gradient Descent (SGD). Ketiga program ini menunjukkan perkembangan dari konsep pemrograman dasar hingga penerapan machine learning.