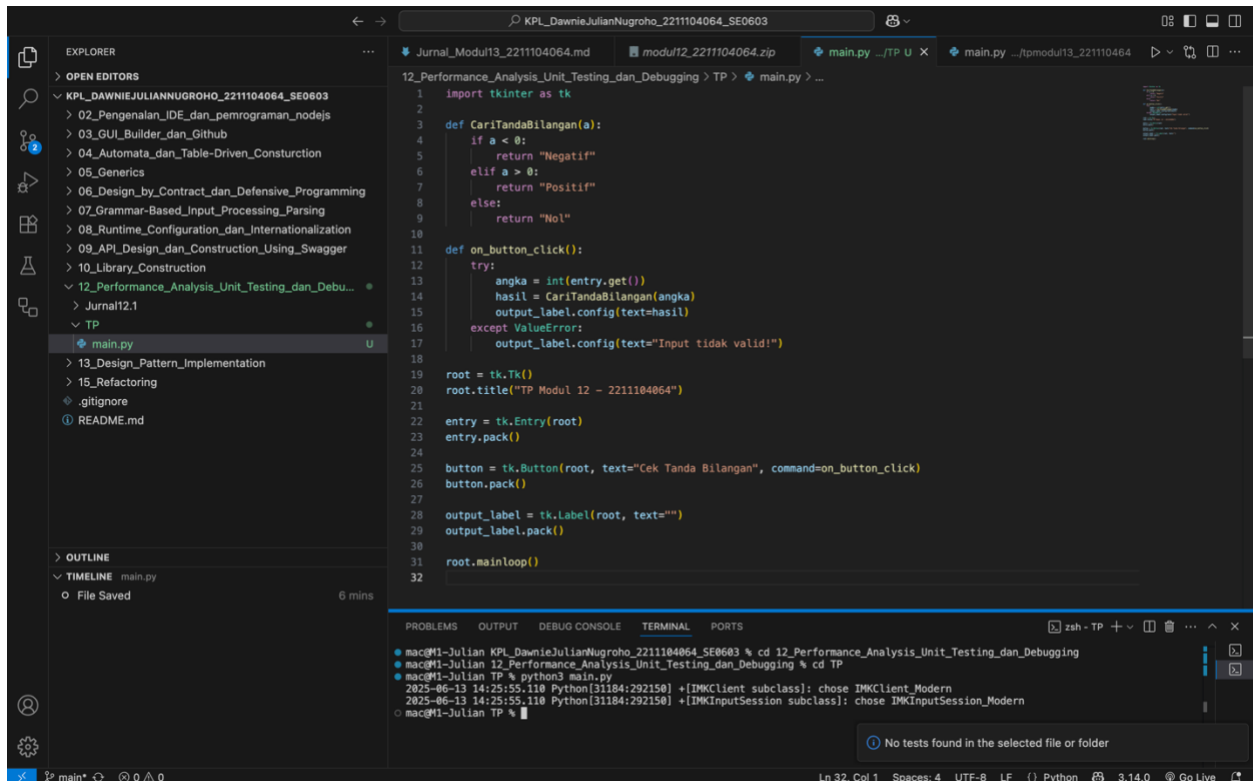


## TP Modul 12

### 1. MEMBUAT PROJECT GUI BARU

Buka IDE misalnya dengan Visual Studio

A. Misalnya menggunakan Visual Studio, buatlah project baru dengan nama tpm modul12\_NIM



B. Pastikan project yang dibuat dapat menggunakan GUI (misalnya tipe Windows Form pada

Visual Studio).

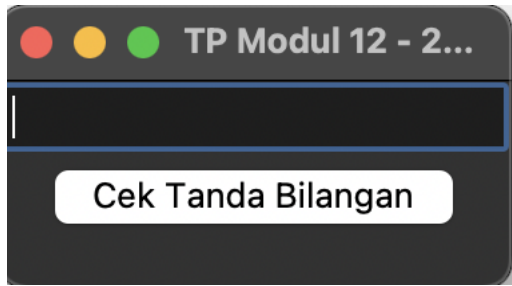
### 2. MEMBUAT GUI SEDERHANA DAN

Pada project yang telah dibuat sebelumnya:

A. Buatlah suatu Form atau tampilan GUI sederhana dengan satu buah textbox, satu button

dan satu label untuk menampilkan output.

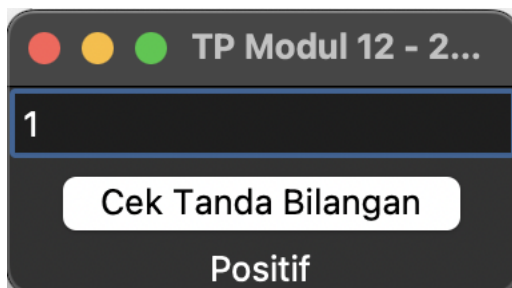
B. Tambahkan satu method dengan nama “CariTandaBilangan(int a)” yang menerima satu input dan mengembalikan nilai string dengan aturan sebagai berikut:



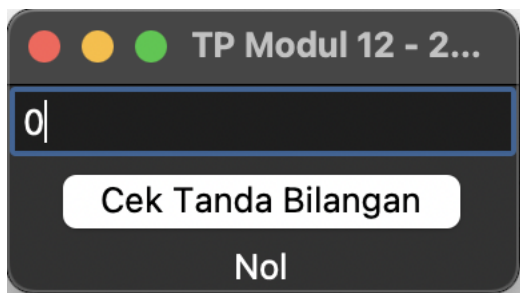
i. Apabila input a kurang dari 0 maka output adalah “Negatif”



ii. Apabila input a lebih dari 0 maka output adalah “Positif”

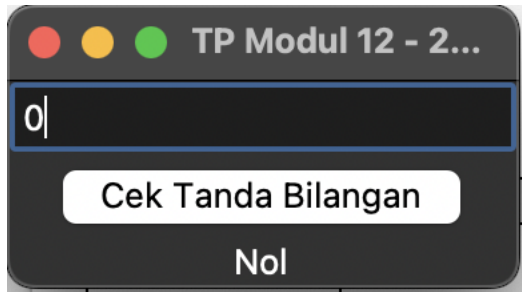


iii. Apabila input sama dengan 0 maka output adalah “Nol”



C. Pada tampilan GUI, pada saat tombol ditekan, maka label output akan menampilkan hasil

pangkat dari pemanggilan fungsi “CariTandaBilangan” dari input textbox yang diberikan.



### 3. MELAKUKAN SOFTWARE PROFILING

Jalankan project yang dibuat sebelumnya dan jalankan profiling tools (misal dari visual studio,

task manager atau sejenisnya):

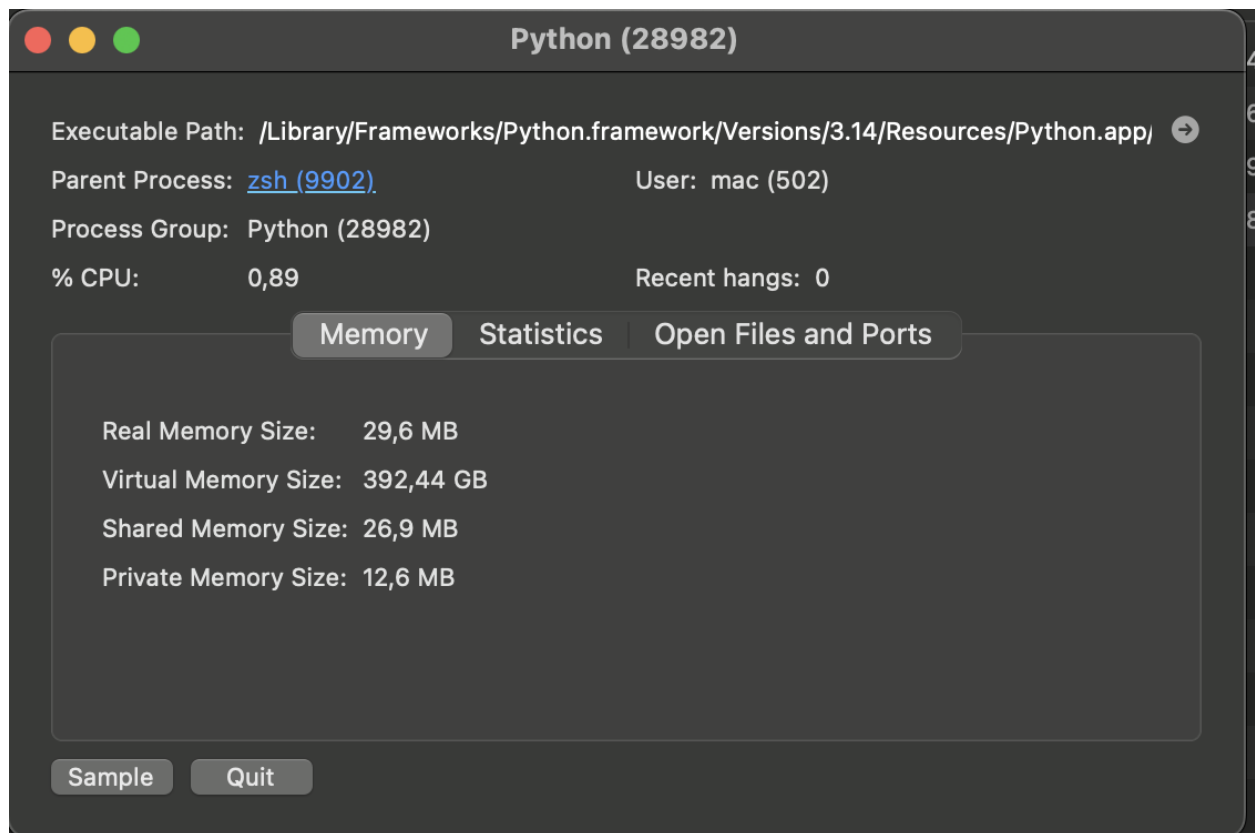
A. Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan

tanpa melakukan input apapun.

B. Pada saat program berjalan, catat dan amati memory usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.

C. Coba masukkan beberapa angka pada textbox dan tekan tombol button.

D. Laporkan apakah terdapat perubahan pada CPU usage dan memory (apabila tidak ada perubahan juga perlu dilaporkan di file docx).

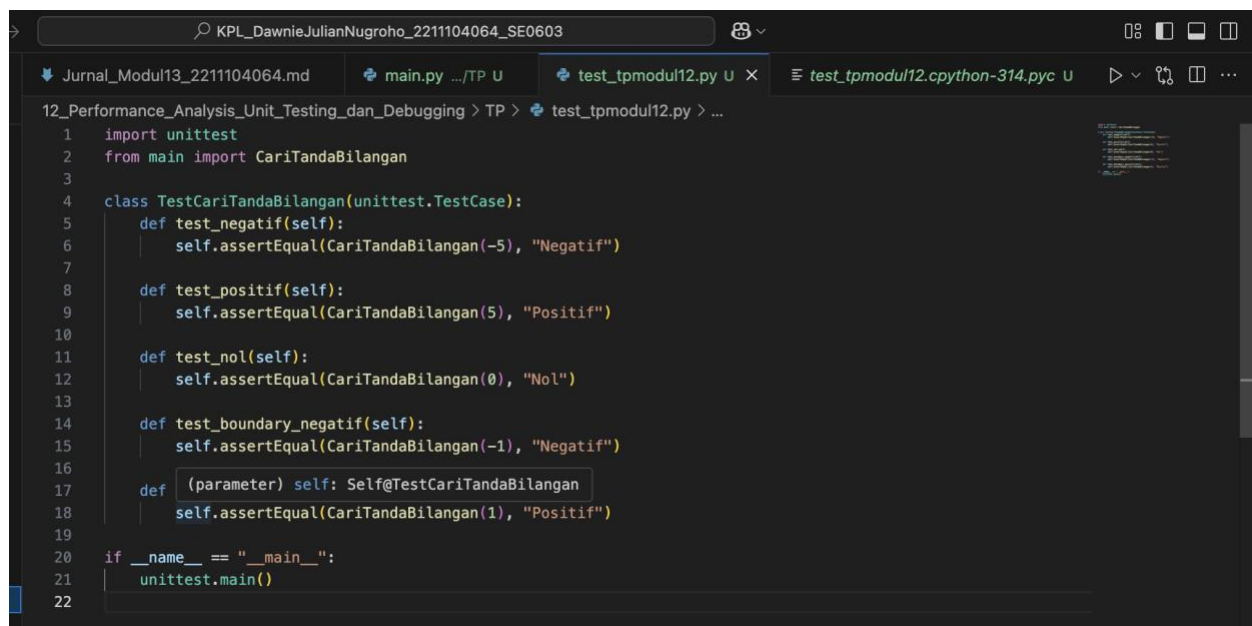


#### 4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

A. Buatlah kode unit test untuk menguji method "CariTandaBilangan" yang dibuat

Sebelumnya

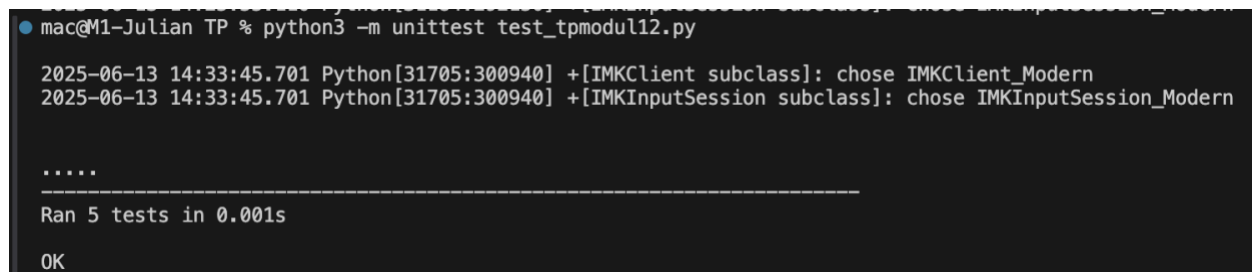


The screenshot shows a code editor with a dark theme. The top bar displays the file path 'KPL\_DawnieJulianNugroho\_2211104064\_SE0603'. Below the top bar, there are tabs for 'Jurnal\_Modul13\_2211104064.md', 'main.py .../TP U', 'test\_tpmodul12.py U', and 'test\_tpmodul12.cpython-314.pyc U'. The main editor area shows the following Python code:

```
12_Performance_Analysis_Unit_Testing_dan_Debugging > TP > test_tpmodul12.py > ...
1  import unittest
2  from main import CariTandaBilangan
3
4  class TestCariTandaBilangan(unittest.TestCase):
5      def test_negatif(self):
6          self.assertEqual(CariTandaBilangan(-5), "Negatif")
7
8      def test_positif(self):
9          self.assertEqual(CariTandaBilangan(5), "Positif")
10
11     def test_nol(self):
12         self.assertEqual(CariTandaBilangan(0), "Nol")
13
14     def test_boundary_negatif(self):
15         self.assertEqual(CariTandaBilangan(-1), "Negatif")
16
17     def (parameter) self: Self@TestCariTandaBilangan
18         self.assertEqual(CariTandaBilangan(1), "Positif")
19
20 if __name__ == "__main__":
21     unittest.main()
22
```

B. Pastikan kode unit test tersebut memiliki branch coverage yang baik untuk method “CariTandaBilangan”.

C. Jalankan kode unit test yang dibuat dan lampirkan hasil unit testing yang dilakukan.



The screenshot shows a terminal window with a dark background. The command executed is 'python3 -m unittest test\_tpmodul12.py'. The output shows the test results for the 'TestCariTandaBilangan' class. The output is as follows:

```
mac@M1-Julian TP % python3 -m unittest test_tpmodul12.py
2025-06-13 14:33:45.701 Python[31705:300940] +[IMKClient subclass]: chose IMKClient_Modern
2025-06-13 14:33:45.701 Python[31705:300940] +[IMKInputSession subclass]: chose IMKInputSession_Modern

.....
Ran 5 tests in 0.001s

OK
```