

# **IMPLEMENTASI PENGUJIAN PERANGKAT LUNAK**

**KELAS A**



**TUGAS 7**

**Penyusun :**

Suhermanto

212410103022

**PRODI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2023**

## Code : Kalkulator

```
def main():  
    print('=' * 25)  
    print('Operasi Matematika')  
    print('1. Jumlah \t [+]')  
    print('2. Kurang \t [-]')  
    print('3. Kali \t [*]')  
    print('4. Bagi \t [/]')  
    print('=' * 25)  
    hitung()
```

Dalam Fungsi Menu ini digunakan untuk membuat menu “Operasi Matematika” seperti ada Jumlah/penjumlahan, kurang/pengurangan, kali/perkalian, dan bagi/pembagian.

```
def hitung(operasi, a, b):  
    if operasi == '1':  
        hasil = a + b  
        return f'Hasil operasi dari {a} + {b} = {hasil}'  
    elif operasi == '2':  
        hasil = a - b  
        return f'Hasil operasi dari {a} - {b} = {hasil}'  
    elif operasi == '3':  
        hasil = a * b  
        return f'Hasil operasi dari {a} * {b} = {hasil}'  
    elif operasi == '4':  
        hasil = a / b  
        return f'Hasil operasi dari {a} / {b} = {hasil}'  
    else:  
        return 'Tidak Valid'
```

Dalam fungsi hitung ini merupakan 5 perulangan yang digunakan untuk melakukan logika perhitungan dari menu sebelumnya, dalam fungsi ini menggunakan 3 parameter yaitu “operasi” yang digunakan untuk pilihan user, “a” ini digunakan untuk input angka user yang pertama, dan “b” untuk input angka yang kedua.

```
if __name__ == '__main__':  
    main()
```

Digunakan untuk menjalankan program yang diatas dimulai dari fungsi/method main()

### UnitTest :

```
import unittest  
from kalkulator import hitung  
  
You, 2 days ago | 1 author (You)  
class TestKalkulator(unittest.TestCase):  
    def test_jumlah(self):  
        result = hitung('1', 5, 3)  
        self.assertEqual(result, 'Hasil operasi dari 5 + 3 = 8')  
  
    def test_kurang(self):  
        result = hitung('2', 10, 3)  
        self.assertEqual(result, 'Hasil operasi dari 10 - 3 = 7')  
  
    def test_kali(self):  
        result = hitung('3', 4, 7)  
        self.assertEqual(result, 'Hasil operasi dari 4 * 7 = 28')  
  
    def test_bagi(self):  
        result = hitung('4', 10, 5)  
        self.assertEqual(result, 'Hasil operasi dari 10 / 5 = 2.0')  
  
    def test_invalid_input(self):  
        result = hitung('5', 10, 2)  
        self.assertEqual(result, 'Tidak Valid')  
  
if __name__ == '__main__':  
    unittest.main()
```

Import Unittest ini digunakan untuk menjalankan modul unittest, from kalkulator ini digunakan untuk mengimport fungsi hitung yang ada pada modul kalkulator yang akan diuji, lalu clas TestKalkulator ini digunakan untuk metode pengujian seperti test\_jumlah, test\_kurang, test\_kali, test\_bagi, dan test\_invalid\_input dengan menggunakan fungsi hitung, setiap metode ini menggunakan self.assertEqual ini untuk memeriksa apakah hasil akan sesuai yang diharapkan atau tidak. Dari code yang sudah dijelaskan itu nantinya akan menguji apakah program atau code kalkulator tersebut akan berfungsi dengan benar dan dapat menghasilkan output sesuai dengan apa yang diharapkan.

```
.....
-----
Ran 5 tests in 0.001s

OK
```

Output yang akan dihasilkan jika semua test benar.

```
def test_bagi(self):
    result = hitung('4', 20, 5)
    self.assertEqual(result, 'Hasil operasi dari 20 / 5 = 2.0')

=====
FAIL: test_bagi (__main__.TestKalkulator.test_bagi)
-----
Traceback (most recent call last):
  File "d:\Kuliah\Semester 5\IPPL\IPPL_Tugas-7\unit-test.py", line 23, in test_bagi
    self.assertEqual(result, 'Hasil operasi dari 20 / 5 = 2.0')
AssertionError: 'Hasil operasi dari 20 / 5 = 4.0' != 'Hasil operasi dari 20 / 5 = 2.0'
- Hasil operasi dari 20 / 5 = 4.0
?                               ^
+ Hasil operasi dari 20 / 5 = 2.0
?                               ^

-----
Ran 5 tests in 0.002s

FAILED (failures=1)
```

jika kita membuat code baru untuk mengetahui error atau tidak nya seperti yang sudah dibuat diatas, seperti kita memilih pilihan operasi 4 dan memasukan angka 20 dan 5 harapan kita nilai yang akan keluar yaitu 4.0 tapi kita coba masukan pada code selanjutnya dengan hasil pembagian yang salah, maka output yang dihasilkan akan ada 1 test yang mengalami failed dan disitu sudah ada apa yang salah dari jawaban dan jawaban apa yang benar.