



BERITA ACARA PEMUSNAHAN REKAM MEDIS TAHUN 2023

Sehubungan dengan surat keputusan Rumah Sakit Tingkat III Baladhika Husada Jember No 12234 dengan ini menenrangkan terkebih dahulu:

1. Bahwa dalam rangka pemusanahan dokumen rekam medis Rumah Sakit Tingkat III Baladhika Husada Jember telah dibentuk tim pemusnahan yang mempunyai tugas untuk melaksanakan pemusnahan rekam medis sebagaimana petunjuk ddan ketentuan yang berlaku.
2. Bahwa pelaksanaan pemusnahan berdasarkan dan mengacu pada peraturan dan keputusanp>

Atas dasar tersebut tim pemusnah rekam medis Rumah Sakit Tingkat III Baladhika Husada Jember telah melakukan pemusnahan rekam medis inaktif tahun -

Nama Petugas : anna

Jabatan : Petugas

Cara Pemusnahan : Dijual ges

Tanggal Pemusnahan : 02, Februari, 2020

Waktu Pemusnahan : 20:30 WIB

Lokasi Pemusnahan : Loak an

Ketua Rekam Medis : Muhammad Rafi Kusdiarto, S.Kom, M.Kom

Lampiran : Berita Acara Pemusnahan 02, Februari, 2020.pdf

Mengetahui,

Kepala Rekam Medis

Nama _____
NIP.

Nama : Muhammad Rafi Kusdiarto

NIM : 202410103059

Kelas : IPPL B

UTS

soal :

Cari / buat program sederhana yang memiliki minimal 5 fungsi/method. Dari fungsi-fungsi yang ada, buatlah unit testnya.

- **Program**

```
• # Fungsi untuk menjumlahkan dua angka
• def tambah(a, b):
•     return a + b
•
• # Fungsi untuk mengurangi dua angka
• def kurang(a, b):
•     return a - b
•
• # Fungsi untuk mengalikan dua angka
• def kali(a, b):
•     return a * b
•
• # Fungsi untuk membagi dua angka
• def bagi(a, b):
•     if b != 0:
•         return a / b
•     else:
•         return "Error: Pembagian oleh nol tidak diizinkan"
•
• # Fungsi untuk menghitung BMI
• def hitung_bmi(berat, tinggi):
•     bmi = berat / (tinggi ** 2)
•     return bmi
•
• # Fungsi untuk menentukan kategori BMI
• def kategori_bmi(bmi):
•     if bmi < 18.5:
•         return "Berat badan kurang"
•     elif 18.5 <= bmi < 24.9:
•         return "Berat badan normal"
```

```

•     elif 25 <= bmi < 29.9:
•         return "Berat badan berlebih"
•     else:
•         return "Obesitas"
•
• # Fungsi utama program
• def main():
•     print("Pilih Operasi:")
•     print("1. Kalkulator")
•     print("2. Hitung BMI")
•     pilihan = input("Masukkan pilihan (1/2): ")
•
•     if pilihan == '1':
•         angka1 = float(input("Masukkan angka pertama: "))
•         angka2 = float(input("Masukkan angka kedua: "))
•         operasi = input("Masukkan operasi (+, -, *, /): ")
•
•         if operasi == '+':
•             print("Hasil: ", tambah(angka1, angka2))
•         elif operasi == '-':
•             print("Hasil: ", kurang(angka1, angka2))
•         elif operasi == '*':
•             print("Hasil: ", kali(angka1, angka2))
•         elif operasi == '/':
•             print("Hasil: ", bagi(angka1, angka2))
•         else:
•             print("Operasi tidak valid.")
•
•     elif pilihan == '2':
•         berat = float(input("Masukkan berat badan (kg): "))
•         tinggi = float(input("Masukkan tinggi badan (m): "))
•         bmi = hitung_bmi(berat, tinggi)
•         print("BMI Anda adalah {:.2f}".format(bmi))
•         print("Kategori BMI Anda: " + kategori_bmi(bmi))
•
•     else:
•         print("Pilihan tidak valid.")
•
• # Memanggil fungsi utama
• if __name__ == "__main__":
•     main()
•

```

```

Pilih Operasi:
1. Kalkulator
2. Hitung BMI
Masukkan pilihan (1/2): 1
Masukkan angka pertama: 20
Masukkan angka kedua: 4
Masukkan operasi (+, -, *, /): +
Hasil: 24.0
PS C:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7> python -u "c:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7\hitungBMI.py"
Pilih Operasi:
1. Kalkulator
2. Hitung BMI
Masukkan pilihan (1/2): 2
Masukkan berat badan (kg): 72
Masukkan tinggi badan (m): 1.68
BMI Anda adalah 25.51
Kategori BMI Anda: Berat badan berlebih
PS C:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7>

```

program yang saya buat di atas adalah gabungan dari program kalkulator operasi matematika (tambah, kurang, kali, bagi) dan penghitungan BMI. BMI adalah indikator pengukuran yang digunakan untuk menentukan kategori berat badan ideal atau tidak.

• Unit Test

```

• # test_bmi_calculator.py
• import unittest
• from hitungBMI import hitung_bmi, kategori_bmi, tambah, kurang, kali, bagi
•
• class TestBMICalculator(unittest.TestCase):
•
•     def test_hitung_bmi(self):
•         self.assertAlmostEqual(hitung_bmi(70, 1.75), 22.86, places=2)
•
•     def test_kategori_bmi(self):
•         self.assertEqual(kategori_bmi(18.0), "Berat badan kurang")
•         self.assertEqual(kategori_bmi(22.0), "Berat badan normal")
•         self.assertEqual(kategori_bmi(27.5), "Berat badan berlebih")
•         self.assertEqual(kategori_bmi(30.0), "Obesitas")
•
•     def test_tambah(self):
•         self.assertEqual(tambah(3, 5), 8)
•         self.assertEqual(tambah(-1, 1), 0)
•
•     def test_kurang(self):
•         self.assertEqual(kurang(8, 5), 3)
•         self.assertEqual(kurang(1, 1), 0)
•
•     def test_kali(self):
•         self.assertEqual(kali(3, 5), 15)
•         self.assertEqual(kali(2, 0), 0)
•
•     def test_bagi(self):
•         self.assertEqual(bagi(10, 2), 5)

```

```

•         self.assertEqual(bagi(5, 0), "Error: Pembagian oleh nol tidak
•         diizinkan")
•
•     if __name__ == '__main__':
•         unittest.main()

```

```

• PS C:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7> python -u "c:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7\testingBMI.py"
• .....
• Ran 6 tests in 0.001s
•
• OK

```

kode di atas adalah untuk unit testnya dari program kalkulator dan BMI. Untuk fungsi yang dilakukan testing ada sebanyak 6 fungsi dan juga ditampilkan hasil testingnya.

Nama : Muhammad Rafi Kusdiarto

NIM : 202410103059

Kelas : IPPL B

Unit Sistem Kalkulator

```
class thisKalkulator:
    def __init__(self, a,b) -> None:
        self.a = a
        self.b = b

    def penjumlahan(self):
        return self.a + self.b

    def pengurangan(self):
        return self.a - self.b

    def perkalian(self):
        return self.a * self.b

    def pembagian(self):
        if self.b == 0:
            raise ValueError(f'Hasil operasi dari {self.a} / {self.b} = Tak
Terhingga / Error')
        return self.a / self.b

def hasil(operasi, angka1, angka2):
    hasil = thisKalkulator(angka1,angka2)

    if operasi == "1":
        operasi = "+"
        hasil = hasil.penjumlahan()

    elif operasi == '2':
        operasi = "-"
        hasil = hasil.pengurangan()

    elif operasi == '3':
        operasi = "*"
        hasil = hasil.perkalian()

    elif operasi == '4':
```

```

        operasi = "/"
        hasil = hasil.pembagian()

    else:
        return 'Tidak valid'

    return f'Hasil operasi dari {angka1} {operasi} {angka2} = {hasil}'

def show():
    print('='*30)
    print('Kalkulator')
    print('1. Penjumlahan \t [+]')
    print('2. Pengurangan \t [-]')
    print('3. Perkalian \t [*]')
    print('4. Pembagian \t [/]')
    print('=' * 30)
    operasi = input('Pilih operasi (1/2/3/4): ')
    angka1 = int(input('Masukkan bilangan pertama: '))
    angka2 = int(input('Masukkan bilangan kedua: '))
    print('=' * 25)
    print(hasil(operasi, angka1, angka2))

if __name__ == '__main__':
    show()

```

Tampilan Ketika Program di Run

```

PS C:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7> python -u "c:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7\kalkulator.py"
=====
Kalkulator
1. Penjumlahan    [+]
2. Pengurangan   [-]
3. Perkalian      [*]
4. Pembagian      [/]
=====
Pilih operasi (1/2/3/4): 3
Masukkan bilangan pertama: 13
Masukkan bilangan kedua: 12
=====
Hasil operasi dari 13 * 12 = 156

```

Unit ini memiliki fungsionalitas untuk sebuah hasil dari operasi matematika yang kita inginkan. Inputan adalah operasi kemudian bilangan. Output berupa hasil dari input bilangan berdasarkan operasi yang dipilih.

Testing:

- ❖ Input: operasi matematika yang ingin dilakukan, bilangan yang akan dicari hasil operasi metematikanya
- ❖ Output: Hasil bilangan dari operasi yang tentukan, untuk menentukan operasi menggunakan percabangan if else
- ❖ Error: Error terjadi apabila ada ketidaksamaan hasil antara operasi yang ditentukan

Test Sistem Kalkulator

```
from kalkulator import thisKalkulator

import unittest

class testOperasi(unittest.TestCase):
    def __init__(self, methodName: str = "runTest") -> None:
        super().__init__(methodName)
        self.operasi = thisKalkulator(10,5)

    def test_penjumlahan(self):
        hasil = self.operasi.penjumlahan()
        self.assertEqual(hasil, 10+5)

    def test_pengurangan(self):
        hasil = self.operasi.pengurangan()
        self.assertEqual(hasil, 10-5)

    def test_perkalian(self):
        hasil = self.operasi.perkalian()
        self.assertEqual(hasil, 10*5)

    def test_pembagian(self):
        hasil = self.operasi.pembagian()
        self.assertEqual(hasil, 10/5)

    def test_pembagian_by_zero(self):
        # Uji kasus ketika b adalah nol
        with self.assertRaises(ValueError):
            hasil = thisKalkulator(10,0)
            hasil.pembagian()

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

Hasil Dari Testing Sistem Kalkulator

```
PS C:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7> python -u "c:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7\testKalkulator.py"
.....
Ran 5 tests in 0.001s

OK
PS C:\Users\RAFI\Downloads\tugas_pertemuan_7\tugas_pertemuan_7>
```