



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231042
Nama Lengkap	Revaldo Fransisco Hohary
Minggu ke / Materi	04 / Modular Programming

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada Modular Programming ini kita akan belajar def().

MATERI 1 Modular Programming

A. Fungsi, Argument dan Parameter

Tentunya kita sudah belajar dari sebelumnya dan sudah tahu apa yang sedang diminta oleh program ini:

```
nama = input("Masukkan nama: ")
print("Hallo ", nama, " selamat pagi!")
```

Di program ini di minta untuk memasukan nama dan juga akan di print dan akan menyapa dengan “selamat pagi”. Pada program ini ada dua fungsi yaitu input() dan print(), ini adalah bawaan dari python yang sering digunakan dan memang harus ada di setiap permasalahan.

Yang di maksud fungsi itu adalah blok kode atau perintah yang digunakan untuk mengelola, membaca serta memiliki satu tujuan untuk digunakan ulang. Keterkaitan antara fungsi dan modular programming sangat diperlukan dalam program yang memiliki langkah yang banyak untuk pengelompokan beberapa kode program menjadi bagian-bagian (block) dari program yang besar yang, maka itu disebut modular, yang mana dapat digunakan ulang dan juga terdiri dari banyak bagian yang memilki kegunaan khusus. Fungsi sendiri dibagi menjadi dua jenis yaitu :

- Fungsi bawaan (built-in function). Daftar fungsi bawaan Python 3 dapat dilihat di <https://docs.python.org/3/library/functions.html>
- Fungsi yang dibuat sendiri oleh programmer

Contohnya sebagai berikut :

```
def tambah(a, b):
    hasil = a + b
    return hasil
```

Dari kode program di atas itu terdiri dari :

- Kode **def** yang digunakan untuk mendefinisikan fungsi.
- Di sebelah def ada nama fungsi yang kita buat yaitu **tambah()**.
- Isi dari fungsi yang dituliskan menjorok ke dalam.
- Untuk di dalam fungsi ada dua argumen, yang nanti menjadi tolak ukur yaitu a dan b.
- Dalam fungsi ini akan menghasilkan jumlah pertambahan dari a dan b dan akan di tampung di sebuah variabel. Di sini juga ada return untuk mengembalikan atau mengeluarkan nilai dari sebuah fungsi.

Salah satu contoh penggunaan fungsi tersebut :

```

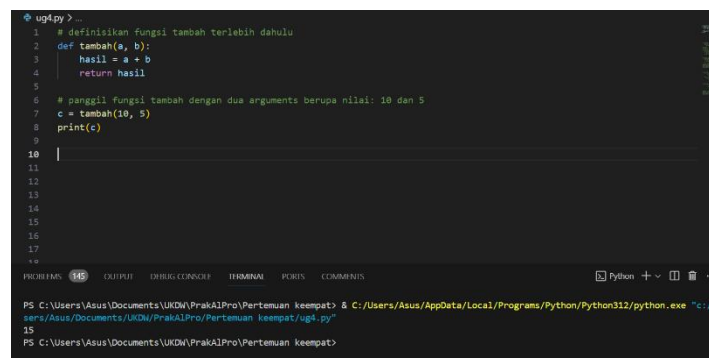
1. # definisikan fungsi tambah terlebih dahulu
2.     def tambah(a, b):
3.         hasil = a + b
4.         return hasil
5.
6. # panggil fungsi tambah dengan dua arguments berupa nilai: 10 dan 5
7.     c = tambah(10, 5)
8.     print(c)

```

Berikut urutan program itu berjalan :

- Baris 1 ada komentar, hanya sekedar komentar tidak akan di eksekusi oleh Python.
- Baris 2-3 ada fungsi dan pada fungsi ini belum bisa berjalan jika fungsi tambah() dipanggil.
- Baris 5-6 ini baris kosong dan juga sebuah komentar yang tidak akan di eksekusi oleh Python.
- Baris 7 variabel c , pada baris 7 ini untuk mendapatkan c disitu kita akan menginput yang akan di isi pada fungsi tambah(). Dan di program itu tambah() sekarang berisi 10 dan 5.
- Selanjutnya, program akan kembali ke baris 2 untuk mengisi fungsi tambah(), jadi sekarang fungsi tambah yang awalnya tambah(a,b) sekarang menjadi tambah(10,7).
- Habis diisi oleh 10 dan 5 akan lanjut ke baris ke 3 dan di situ akan dieksekusi untuk pertambahan $10 + 5 = 15$.
- Dan untuk selanjutnya, pada baris 4 terdapat return, yang mana menandakan akhir dari fungsi tersebut dan dihasilkan 15.
- Kemudian sesudah dari baris 4 dilanjutkan ke baris 7, dikarenakan fungsi tersebut kembali di mana dia di panggil. Dan sekarang pada baris 7 variabel c sudah diisi dengan 15 dari hasil penjumlahan tersebut.
- Sesudah itu dilanjutkan ke baris 8 untuk menampilkan variabel c ke layar, untuk menampilkannya kita menggunakan print(c) dan output yang keluar akan 15 dari hasil penjumlahan yang sudah diisi di variabel c.
- Maka urutan dari program tersebut adalah 1-2-3-4-5-6-7-2-3-4-7-8.

Berikut jika code program tersebut di masukan dan di munculkan menggunakan VS Code :



The image shows a VS Code editor window with a Python file named 'ug4.py'. The code in the file is as follows:

```

1 # definisikan fungsi tambah terlebih dahulu
2 def tambah(a, b):
3     hasil = a + b
4     return hasil
5
6 # panggil fungsi tambah dengan dua arguments berupa nilai: 10 dan 5
7 c = tambah(10, 5)
8 print(c)

```

The terminal at the bottom shows the command to run the script and the output:

```

PS C:\Users\Asus\Documents\UKOM\PrakAIPro\Pertemuan keempat> & C:\Users\Asus\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Asus/Documents/UKOM/PrakAIPro/Pertemuan keempat/ug4.py"
15
PS C:\Users\Asus\Documents\UKOM\PrakAIPro\Pertemuan keempat>

```

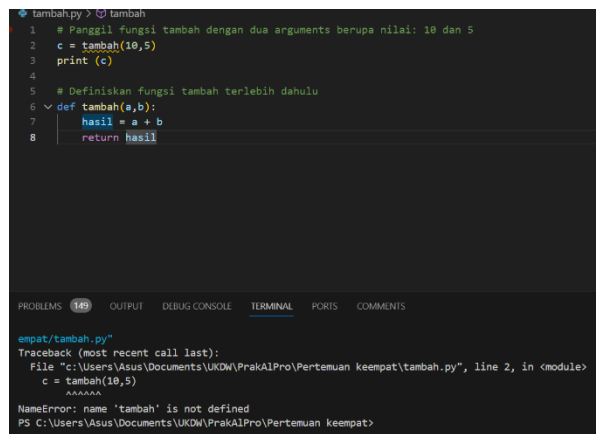
Gambar 4.1 : Hasil dari code program di atas.

B. Return Value

Secara umum, yang dikeluarkan oleh fungsi ada dua hal yaitu : fungsi yang tidak mengembalikan nilai atau disebut (void function) dan fungsi yang mengembalikan nilai. Contoh void function sebagai berikut :

```
def print_twice(message):  
    print(message)  
    print(message)  
  
print_twice("Hello World!")
```

Fungsi `print_twice()` membutuhkan satu isi atau parameter yang diatas saya isi dengan `message`. Dan untuk selanjutnya akan di tampilkan melalui `print` sebanyak dua kali. Tapi dalam codingan ini fungsi `print_twice()` tidak menghasilkan suatu nilai untuk proses selanjutnya. Tapi fungsi `print_twice` akan mengembalikan nilai `None`. Contohnya sebagai berikut:



```
tambah.py > tambah  
1 # Panggil fungsi tambah dengan dua arguments berupa nilai: 10 dan 5  
2 c = tambah(10,5)  
3 print(c)  
4  
5 # Definiskan fungsi tambah terlebih dahulu  
6 def tambah(a,b):  
7     hasil = a + b  
8     return hasil  
  
PROBLEMS 149 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
empat/tambah.py  
Traceback (most recent call last):  
  File "c:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat\tambah.py", line 2, in <module>  
    c = tambah(10,5)  
      ^^^^^  
NameError: name 'tambah' is not defined  
PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat>
```

Gambar 4.2 : Hasil dari void function

```
def print_twice(message):  
    print(message)  
    print(message)  
  
print(print_twice("Hello World!"))
```

Output yang dihasilkan adalah :

```
Hello World!  
Hello World!  
None
```

Berbeda dengan fungsi `tambah()` berikut ini:

```
def tambah(a, b, c):  
    hasil = a + b + c  
    return hasil
```

Dalam fungsi tambah() itu membutuhkan 3 isian yaitu a, b, dan c. Dan habis itu di dalam itu didefinisikan suatu hasil dalam pertambahan a+b+c. Kemudian habis mengetahui hasil itu di keluarkan dari fungsi dengan **return**. Return itu biasanya digunakan untuk

- Mengeluarkan nilai yang ada dari fungsi
- Mengakhiri fungsi

Fungsi tambah pada kode program di atas bisa digunakan mencari hasil dari penjumlahan. Dalam penjumlahan di atas bisa digunakan untuk proses/langkah berikutnya, seperti contoh berikut mencari rata-rata ketiga bilangan berikut :

```
1. def tambah(a, b, c):
2.     hasil = a + b + c
3.     return hasil
4.
5. nilai1 = 70
6. nilai2 = 85
7. nilai3 = 55
8. rata_rata = tambah(nilai1, nilai2, nilai3)/3
9. print(rata_rata)
```

Urutan jalannya program tersebut:

- Baris 1, 2, 3 adalah definisi fungsi tambah. Belum ada yang dijalankan.
- Baris 5, 6, 7 mendefinisikan tiga variabel dan langsung diisi nilainya.
- Baris 9 memanggil fungsi tambah, dengan mengirimkan tiga arguments yaitu nilai1 (70), nilai2 (85) dan nilai 3 (55). Program akan berjalan ke baris 1.
- Baris 1 ada 3 parameter yang diisi nilainya oleh tiga arguments yang dikirimkan, sehingga a = 70, b = 85 dan c = 55.
- Baris 2 mendefinisikan variabel hasil yang diisi oleh nilai a+b+c, sehingga hasil = 70 + 85 + 55 = 210.
- Baris 3 return nilai dari hasil, berarti fungsi tambah() mengeluarkan hasil dengan nilai 210. Dengan adanya return, artinya fungsi tambah() sudah berakhir dan program akan berjalan ke baris 9 (baris di mana fungsi tambah() dipanggil sebelumnya).
- Baris 9 sudah didapatkan hasil dari fungsi tambah, sehingga menjadi rata_rata = 210/3 = 70. Variabel rata_rata bernilai 70.
- Baris 10 menampilkan nilai dari variabel rata_rata, yaitu 70.

C. Optional Argument dan Named Argument

Fungsi dapat memiliki opsional parameter atau memiliki sifat nilai bawaan(default) yang sudah di isi sebelumnya. Untuk mendefinisikan parameter tersebut, harus didefinisikan nilai bawaannya terlebih dahulu seperti berikut :

```
def hitung_belanja(belanja, diskon=0):
    bayar = belanja - (belanja * diskon)/100
    return bayar
```

Fungsi `hitung_belanja` memiliki dua parameter yaitu `belanja` dan `diskon`. `Diskon` terisi dengan nilai 0 atau 0%. Untuk memanggil fungsi tersebut, sebagai berikut:

```
def hitung_belanja(belanja, diskon=0):  
    bayar = belanja - (belanja * diskon)/100  
    return bayar  
  
print(hitung_belanja(100000))  
print(hitung_belanja(100000, 10))  
print(hitung_belanja(100000, 50))
```

Outputnya sebagai berikut:

```
100000.0  
90000.0  
50000.0
```

D. Anonymous Function (Lambda)

Anonymous function adalah fungsi tanpa nama, ini adalah fitur tambahan dan bukan fitur utama Python. Pada Python, digunakan `lambda` untuk mendefinisikan anonymous function. Seperti berikut :

```
def tambah(a, b):  
    hasil = a + b  
    return hasil  
  
print(tambah(10,20))
```

Jika menggunakan `lambda`:

```
tambah = lambda a, b: a + b  
  
print(tambah(10,20))
```

Setiap anonymous function pada Python terdiri dari beberapa bagian berikut ini:

- Keyword: `lambda`.
- Bound variable: argument pada `lambda` function.
- Body: bagian utama `lambda`, berisi ekspresi atau statement yang menghasilkan suatu nilai.

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1

Pada soal ini kita akan menggunakan VS code. Berikut :

a) Source code

```
# Memasukan Fungsi a,b,c
def cek_angka(a, b, c)

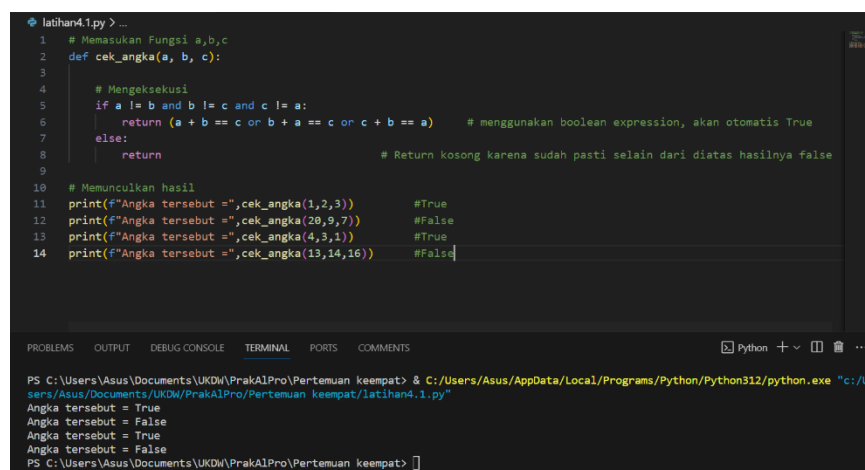
# Mengeksekusi
if a != b and b != c and c != a:
    return (a + b == c or b + a == c or c + b == a)
else:
    return

# Memunculkan hasil
print(f"Angka tersebut =", cek_angka(1,2,3))           #True
print(f"Angka tersebut =", cek_angka(20,9,7))         #False
print(f"Angka tersebut =", cek_angka(4,3,1))          #True
print(f"Angka tersebut =", cek_angka(13,14,16))       #False
```

b) Penjelasan singkat

Disini saya pertama kali memasukan fungsi, dan selanjutnya mengeksekusi, disitu saya tidak menambahkan return true/false karena saya menggunakan boolean expression yang dimana akan langsung memunculkan true dan falsenya secara langsung, dan untuk yang terakhir saya memunculkan hasil dan mengisi inputan a, b, dan c.

c) Inputan dan Output



```
latihan4.1.py > ...
1 # Memasukan Fungsi a,b,c
2 def cek_angka(a, b, c):
3
4     # Mengeksekusi
5     if a != b and b != c and c != a:
6         return (a + b == c or b + a == c or c + b == a) # menggunakan boolean expression, akan otomatis True
7     else:
8         return # Return kosong karena sudah pasti selain dari diatas hasilnya false
9
10 # Memunculkan hasil
11 print(f"Angka tersebut =", cek_angka(1,2,3))           #True
12 print(f"Angka tersebut =", cek_angka(20,9,7))         #False
13 print(f"Angka tersebut =", cek_angka(4,3,1))          #True
14 print(f"Angka tersebut =", cek_angka(13,14,16))       #False

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - [] ...

PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat> & C:\Users\Asus\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Asus/Documents/UKDW/PrakAIPro/Pertemuan keempat/latihan4.1.py"
Angka tersebut = True
Angka tersebut = False
Angka tersebut = True
Angka tersebut = False
PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat> |
```

Gambar 4.1 : Hasil inputan dan output

SOAL 2

Pada soal ini juga masih sama kita akan menggunakan VS Code :

a) Source code

```
# memasukan fungsi x,y,z
def cek_digit_belakang(x, y, z):

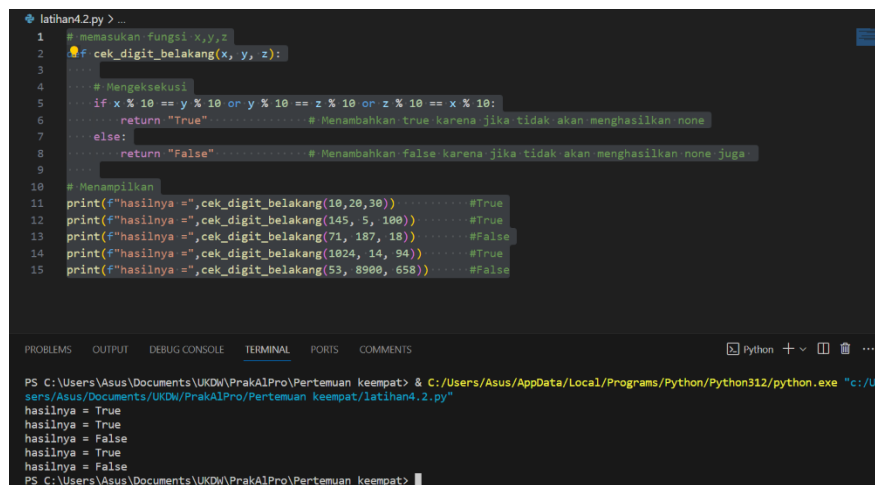
# Mengeksekusi
if x % 10 == y % 10 or y % 10 == z % 10 or z % 10 == x % 10:
    return "True"
else:
    return "False"

# Menampilkan
print(f"hasilnya =",cek_digit_belakang(10,20,30))           #True
print(f"hasilnya =",cek_digit_belakang(145, 5, 100))        #True
print(f"hasilnya =",cek_digit_belakang(71, 187, 18))         #False
print(f"hasilnya =",cek_digit_belakang(1024, 14, 94))        #True
print(f"hasilnya =",cek_digit_belakang(53, 8900, 658))        #False
```

b) Penjelasan singkat

Pada khusus kali ini saya pertama memasukan fungsi yaitu x,y dan z disini kita akan mencari digit belakang yang sama dengan menggunakan mod(%) seperti pada source code, jika sama maka true, jika beda maka false, dan yang terakhir akan menampilkan hasil tersebut dalam bentuk true atau false.

c) SS Inputan dan Output



The screenshot shows a VS Code editor with a Python file named 'latihan4.2.py'. The code defines a function 'cek_digit_belakang' that checks if the last digits of three numbers (x, y, z) are the same using modulo 10. It then prints the results for five different sets of numbers. The terminal output shows the execution of the script, displaying the results: 'hasilnya = True', 'hasilnya = True', 'hasilnya = False', 'hasilnya = True', and 'hasilnya = False'.

```
1 # memasukan fungsi x,y,z
2 def cek_digit_belakang(x, y, z):
3
4 # Mengeksekusi
5 if x % 10 == y % 10 or y % 10 == z % 10 or z % 10 == x % 10:
6     return "True"
7 else:
8     return "False"
9
10 # Menampilkan
11 print(f"hasilnya =",cek_digit_belakang(10,20,30))
12 print(f"hasilnya =",cek_digit_belakang(145, 5, 100))
13 print(f"hasilnya =",cek_digit_belakang(71, 187, 18))
14 print(f"hasilnya =",cek_digit_belakang(1024, 14, 94))
15 print(f"hasilnya =",cek_digit_belakang(53, 8900, 658))
```

PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat> & C:/Users/Asus/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Asus/Documents/UKDW/PrakAIPro/Pertemuan keempat/latihan4.2.py"

hasilnya = True
hasilnya = True
hasilnya = False
hasilnya = True
hasilnya = False

PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat>

Gambar 4.2 : Hasil Inputan dan Output

SOAL 3

Soal ketiga ini kita disuruh mengkonversikan Celcius :

a) Source code

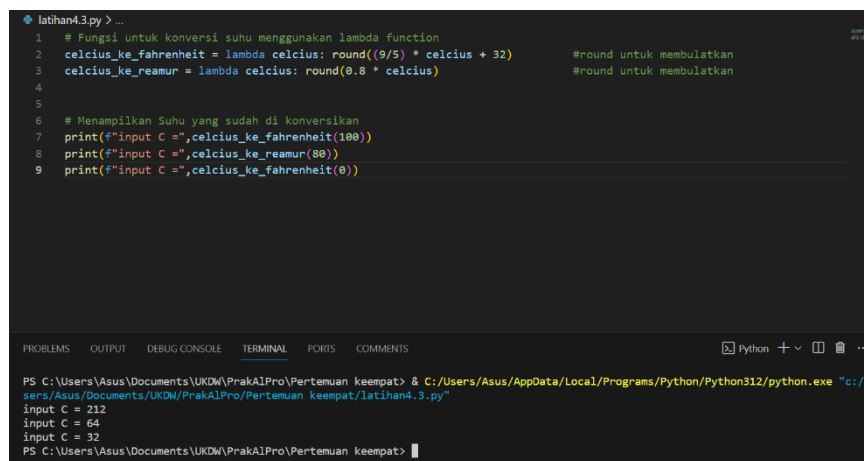
```
# Fungsi untuk konversi suhu menggunakan lambda function
celcius_ke_fahrenheit = lambda celcius: round((9/5) * celcius + 32)
celcius_ke_reamur = lambda celcius: round(0.8*celcius)

# Menampilkan Suhu yang sudah di konversikan
print(f"input C =",celcius_ke_fahrenheit(100))
print(f"input C =",celcius_ke_reamur(80))
print(f"input C =",celcius_ke_fahrenheit(0))
```

b) Penjelasan singkat

Dalam khusus ini kita akan melihat rumus dari celcius ke fahrenheit dan reamur terlebih dahulu dan jika sudah kita akan memasukkannya ke fseperti pada source code diatas dengan menggunakan lambda function, disitu saya menambahkan round untuk membulatkan hasil dari konversi terbut supaya tidak ada 0 di belakang koma, sesudah itu saya akan menampilkan dan menambahkan inputan di sorce code tersebut.

c) SS Inputan dan Output

The image shows a screenshot of a code editor with a Python script and its terminal output. The script defines two lambda functions for temperature conversion and prints the results for specific inputs. The terminal shows the execution of the script, displaying the converted values for 100, 80, and 0 degrees Celsius.

```
latihan4.3.py >...
1 # Fungsi untuk konversi suhu menggunakan lambda function
2 celcius_ke_fahrenheit = lambda celcius: round((9/5) * celcius + 32)      #round untuk membulatkan
3 celcius_ke_reamur = lambda celcius: round(0.8 * celcius)                #round untuk membulatkan
4
5
6 # Menampilkan Suhu yang sudah di konversikan
7 print(f"input C =",celcius_ke_fahrenheit(100))
8 print(f"input C =",celcius_ke_reamur(80))
9 print(f"input C =",celcius_ke_fahrenheit(0))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
Python + - [ ] [ ] ...

PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat> & C:/Users/Asus/AppData/Local/Programs/Python/Python312/python.exe "c:/U
sers/Asus/Documents/UKDW/PrakAIPro/Pertemuan keempat/latihan4.3.py"
input C = 212
input C = 64
input C = 32
PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Pertemuan keempat>
```

Gambar 4.3 : Hasil Input dan Output

E. GitHub

<https://github.com/Frealy0901/PraAIPro4.git>