|  |  |
| --- | --- |
| A picture containing text, clipart  Description automatically generated | **Laporan Praktikum Algoritma dan Pemrograman**  Semester Genap 2023/2024 |

|  |  |
| --- | --- |
| **NIM** | **71230995** |
| **Nama Lengkap** | **Alven Tendrawan** |
| **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.**

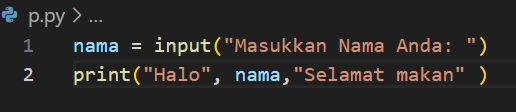
**SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.**

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

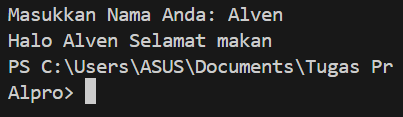
# BAGIAN 1: MATERI MINGGU INI (40%)

**Fungsi, Argument dan Parameter**

**Code:**

****

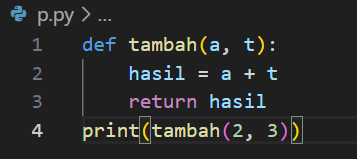
**Output:**

****

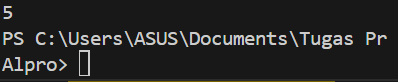
Program di atas merupakan program yang meminta input nama pengguna lalu menyapa nama pengguna tersebut. Seperti biasa kita menggunakan 2 fungsi bawaan python yakni fungsi “input” yang berfungsi untuk membaca input yang dimasukkan oleh pengguna sedangkan “print” berfungsi untuk menampilkan fungsi di layar.

Fungsi merupakan kumpulan perintah yang digabungkan menjadi satu, yang memiliki tujuan dan kegunaan khusus yang bisa digunakan ulang. Jika kita ingin membuat program yang membutuhkan langkah yang banyak, maka perlu untuk mengelompokkan beberapa kode program menjadi bagian-bagian (block). Karena itu disebut sebagai modular.

**Code:**

****

**Output:**

****

**Penjelasan:**

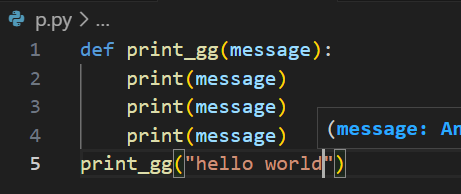
Pada fungsi di atas “def” didefnisikan sebuah fungsi. Nama fungsi yang dibuat adalah “bagi”(/). Setelah menuliskan “def” dan nama fungsi, isi dituliskan menjorok ke dalam dengan jarak 1 tab, ini agar tidak terjadi error. Pada fungsi ini membutuhkan minimal 2 parameter yang nantinya bisa disebut dengan parameter a dan t, kita bisa menggunakan huruf apapun untuk parameternya. Lalu ada keyword return yang berfungsi untuk mengembalikan/mengeluarkan nilai dari suatu fungsi.

Kita menggunakan angka untuk mendefinisikan fungsi , jika kita belum mendefinisikan fungsi maka akan terjadi kesalahan. Untuk memanggil fungsi di atas bisa ketik **print(nama fungsi(parameter1, parameter2))**. Selanjutnya kita dapat melihat output yang ditampilkan program. Parameter harus dituliskan secara berurutan. Parameter akan mengikuti parameter mana yang ditulis pertama. Seperti itulah ilustrasi pertama pada modular programming ini.

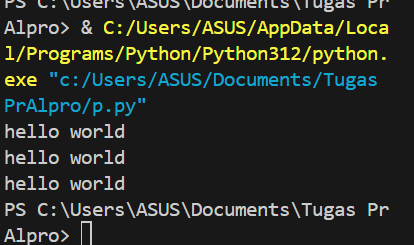
**Return Value**

Berdasarkan hasil yang dikeluarkan, secara umum fungsi ada 2 jenis yaitu fungsi yang mengembalikan nilai dan fungsi yang tidak mengembalikan nilai. Fungsi yang tidak mengembalikan nilai yang biasa disebut dengan void function. Sebagai contoh pada Source Code dibawah tersebut. Berikut ini adalah fungsi yang tidak mengembalikan nilai:

**Code:**

****

**Output:**

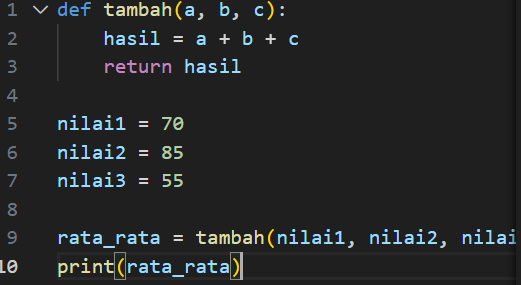
****

**Penjelaasan:**

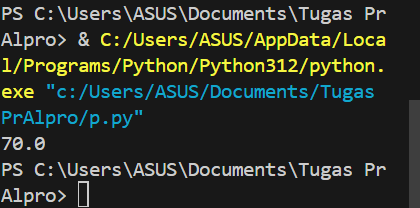
Pada fungsi print\_gg(), hanya menggunakan 1 parameter yaitu “massage”. Didalam fungsi “print\_gg()” tersebut akan menampilkan “hello world” sebanyak 3 kali. Kita cukup menulis nama fungsi untuk memanggil variable tersebut lalu memasukan parameter yang kita inginkan sesuai pada Code di atas(misal “hello world”.

Berikut adalah contoh kasus lain untuk materi return value:

**Code:**

****

**Output:**

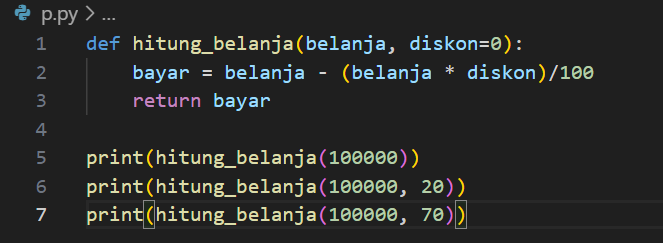
****

**Penjelasan:**

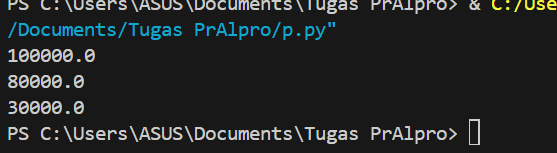
Pada Fungsi tambah() membutuhkan 3 parameter yaitu a, b dan c. Kemudian di dalam fungsi tambah() didefinisikan suatu variabel hasil yang diisi oleh a+b+c. Lalu menggunakan keyword “return” untuk mengeluarkan variabel tersebut dari fungsi sebagai hasil dari fungsi tambah(). Keyword return digunakan untuk **mengeluarkan nilai yang merupakan hasil dari fungsi dan mengakhiri fungsi**. Fungsi tambah() tersebut bisa digunakan untuk mencari hasil penjumlahan tiga bilangan jika kita ingin menjumlahkan lebih darti itu makan kita bisa menambahkan variabel lagi di sorce codenya.

**Optional Argument dan Named Argument**

Fungsi dapat memiliki optional parameter, yaitu parameter yang **bersifat opsional dan memiliki nilai bawaan (default)** yang sudah didefinisikan sebelumnya. Untuk mendefinisikan optional parameter, kita harus mendefinisikan nilai bawaannya terlebih dahulu. Berikut contohnya:

**Code:**

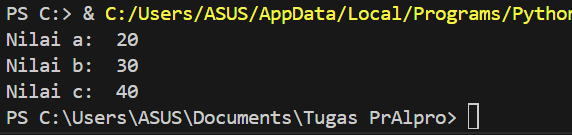
**Output:**

****

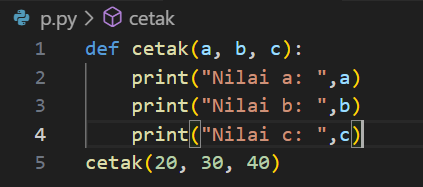
**Penjelasan:**

Fungsi hitung\_belanja() ini menggunakan 2 parameter yakni “belanja” dan “diskon”. Fungsi ini berguna untuk menghitung total belanja. Pada pemanggilan pertama mengunakan satu argumen dan pemanggilan kedua serta ketiga menggunakan dua argumen. Untuk menjalankan fungsi hitung\_belanja kita dapat memasukkan rumus **bayar = belanja – (belanja \* diskon)/100,** lalu di return.

Setiap parameter pada fungsi memiliki nama. Karena itu pada pemanggilan fungsi juga dapat disertakan nama parameternya dan tidak perlu sesuai dengan urutan yang diberikan. Berikut contohnya:

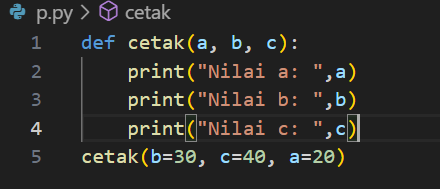
**Code:**

**Output:**

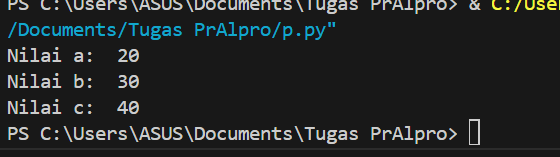
****

**Penjelasan:** Kita harus memastikan urutan argumen harus sama pada source di atas, nah agar urutan argument tidak harus sama dengan urutan parameter maka dapat menggunaka source code berikut:

**Code:**



**Output:**

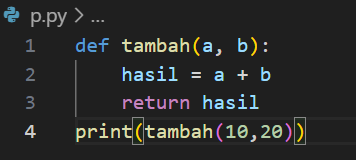
****

**Anonymous Function (Lambda)**

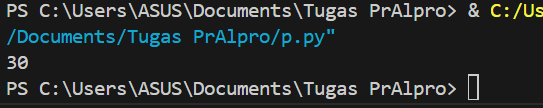
Sesuai dengan namanya, anonymous function adalah fungsi tanpa nama (anonymous). Anonymous function pada Python adalah fitur tambahan, bukan merupakan fitur utama. Berbeda dengan bahasa-bahasa pemrograman seperti Haskell, Lisp dan Erlang yang merupakan bahasa pemrograman fungsional. Pada Python, digunakan keyword lambda untuk mendefinisikan anonymous function. Sebagai contoh, perhatikan fungsi tambah() berikut ini:

**Tidak menggunakan Lambda**

**Code:**

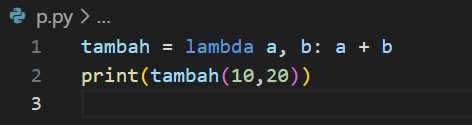
****

**Output:**

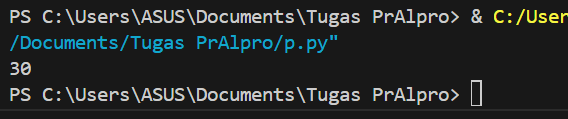
****

**Dengan Lambda**

**Code:**

****

**Output:**

****

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

• Keyword: lambda

• Bound variable: argument pada lambda function

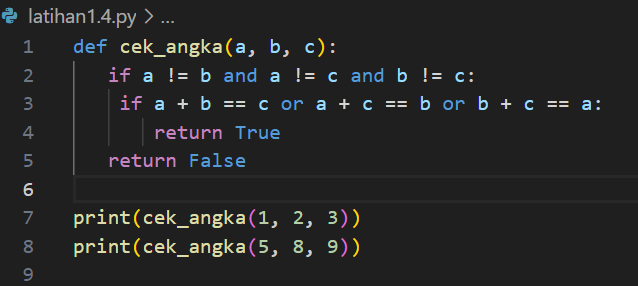
• Body: bagian utama lambda, berisi ekspresi atau statement yang menghasilkan suatu nilai

# BAGIAN 2: LATIHAN MANDIRI(60%)

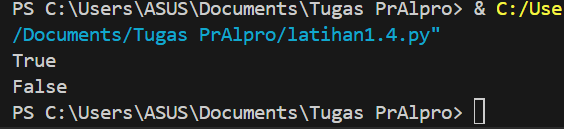
Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

## SOAL 1

**Code:**

****

**Output:**

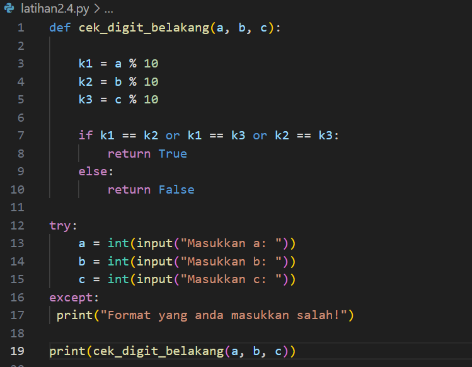
****

**Penjelasan:**

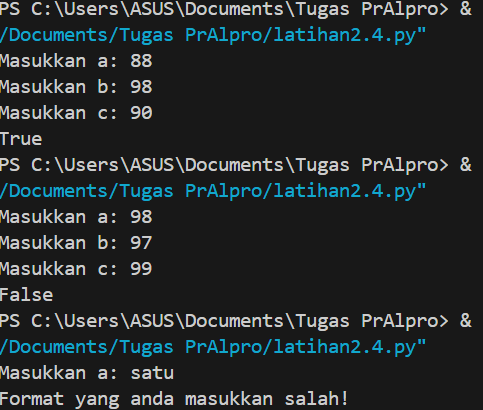
Nama fungsi yang digunakan yaitu cek\_angka, fungsi tersebut mempunyai tiga parameter yaitu a, b, dan c. Pada kondisi pertama, dinyatakan ketiga parameter tersebut berbeda semua, jika ada yang sama maka hasilnya **FALSE**. Lalu pada kondisi kedua jika dua angka dijumlahkan hasilnya adalah angka yang tersisa maka **TRUE**. Jika tidak ada yang sesuai hasil penjumlahannya maka akan menghasilkan **FALSE**. Untuk memanggil dapat menggunakan print(nama\_fungsi(parameter)

## SOAL 2

**Code:**

****

**Output:**

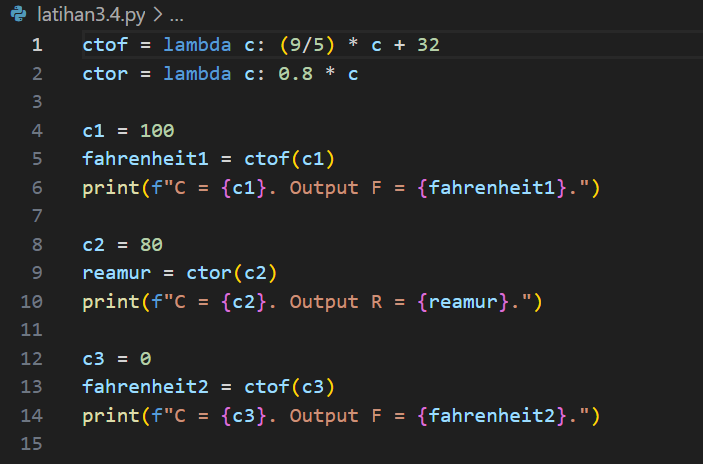
****

**Penjelasan:**

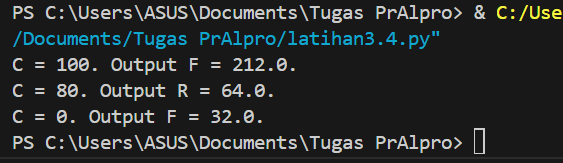
Nama fungsi yang digunakan yaitu cek\_digit\_belakang, fungsi tersebut memiliki tiga parameter yaitu a, b, c. Untuk menghasilkan angka paling kanan belakang menggunakan modulus 10(% 10). Maka dibuatlah 3 variabel yaitu: k1, k2, k3 untuk menghasilkan angka paling kanan belakang. Lalu membuat kondisi digit paling kanan belakang harus ada yang sama dengan digit kanan belakang parameter lainnya setidaknya 1 yang sama. Jika tidak ada digit paling kanan yang sama maka akan menghasilkan output **False.** Terakhir membuat fungsi untuk input dari pengguna.

## SOAL 3

**Code:**

****

**Output:**

****

**Penjelasan:**

Pada lambda function kita tuliskan nama fungsi nya masing- masing terlebih dahulu yaitu ctof (celcius\_to\_fahrenheit) lalu masukkan rumus sesuai soal dan ctor (celcius\_to\_reamur) lalu masukkan rumus sesuai soal. Untuk pemanggilan fungsi lambda tersebut kita membuat variabel baru yaitu: c1, c2, c3. Terakhir untuk melakukan pemanggilan kita lakukan print() dan pastikan output sama dengan pernyataan di soal.