



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231005
Nama Lengkap	Josephine Marcelia
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

Fungsi, Argument dan Parameter

Fungsi

Fungsi adalah kumpulan perintah-perintah yang dijadikan satu, memiliki suatu tujuan dan kegunaan khusus serta dapat digunakan ulang. Sedangkan modular programming pada Python merupakan konsep memecahkan program besar menjadi bagian-bagian yang lebih kecil, yang disebut modul-modul atau komponen. Setiap modul memuat fungsi, kelas, atau variabel terkait dan memiliki tugas atau fungsionalitas tertentu.

Berdasarkan asalnya, fungsi dibagi menjadi dua jenis yaitu:

- Fungsi bawaan (built-in function).

Daftar Fungsinya :

A <u>abs()</u> <u>aiter()</u> <u>all()</u> <u>anext()</u> <u>any()</u> <u>ascii()</u>	E <u>enumerate()</u> <u>eval()</u> <u>exec()</u>	L <u>len()</u> <u>list()</u> <u>locals()</u>	R <u>range()</u> <u>repr()</u> <u>reversed()</u> <u>round()</u>
B <u>bin()</u> <u>bool()</u> <u>breakpoint()</u> <u>bytearray()</u> <u>bytes()</u>	F <u>filter()</u> <u>float()</u> <u>format()</u> <u>frozenset()</u>	M <u>map()</u> <u>max()</u> <u>memoryview()</u> <u>min()</u>	S <u>set()</u> <u>setattr()</u> <u>slice()</u> <u>sorted()</u> <u>staticmethod()</u> <u>str()</u> <u>sum()</u> <u>super()</u>
C <u>callable()</u> <u>chr()</u> <u>classmethod()</u> <u>compile()</u> <u>complex()</u>	G <u>getattr()</u> <u>globals()</u>	N <u>next()</u>	T <u>tuple()</u> <u>type()</u>
D <u>delattr()</u> <u>dict()</u> <u>dir()</u> <u>divmod()</u>	H <u>hasattr()</u> <u>hash()</u> <u>help()</u> <u>hex()</u>	O <u>object()</u> <u>oct()</u> <u>open()</u> <u>ord()</u>	V <u>vars()</u>
	I <u>id()</u> <u>input()</u> <u>int()</u> <u>isinstance()</u> <u>issubclass()</u> <u>iter()</u>	P <u>pow()</u> <u>print()</u> <u>property()</u>	Z <u>zip()</u>
			- <u>__import__()</u>

- Fungsi yang dibuat sendiri oleh programmer.

Contoh :

Fungsi untuk menghitung luas lingkaran

```
pi = 22/7

def luas_lingkaran(jari_jari, pi=pi):
    luas = 0.5 * pi * jari_jari**2
    return luas

print(luas_lingkaran(5))
```

Output :

39.285714285714285

Beberapa hal yang perlu diperhatikan dalam fungsi luas lingkaran tersebut:

- Keyword def digunakan untuk mendefinisikan sebuah fungsi.
- Nama fungsi yang dibuat adalah luas_lingkaran().
- Isi dari fungsi harus anda tuliskan menjorok ke dalam 1 tab.
- Fungsi luas_lingkaran() membutuhkan dua argument, yang nantinya akan dikenali sebagai parameter jari_jari, dan pi.
- Fungsi tersebut akan menghasilkan hasil luas lingkaran yang dapat ditampung di sebuah variabel. Keyword return digunakan untuk mengembalikan/mengeluarkan nilai dari suatu fungsi.

Argumen

Argumen adalah nilai atau variabel yang diterima oleh fungsi saat fungsi tersebut dipanggil. Argumen dapat berupa nilai tunggal atau sekelompok nilai yang dikemas dalam sebuah tuple, list, atau dictionary. Argumen digunakan untuk menyediakan input yang dibutuhkan oleh fungsi saat dijalankan.

Contoh :

```
def jumlah_digit(bilangan):  
    jumlah = 0  
    while bilangan != 0:  
        bilangan //= 10  
        jumlah += 1  
    return jumlah  
  
bilangan = int(input("Masukkan bilangan: "))  
jumlah = jumlah_digit(bilangan)  
print("Jumlah digit dari bilangan tersebut adalah: ", jumlah)
```

- Dalam contoh di atas, kita memiliki sebuah fungsi bernama mencari_min_max yang menerima satu argumen, angka, yaitu sebuah list bilangan.
- Fungsi ini akan mencari nilai terbesar dan terkecil dari list tersebut dengan cara menginisialisasi min dan max dengan elemen pertama dari list.
- Kemudian, fungsi akan melakukan iterasi pada setiap elemen dari list dan membandingkannya dengan min dan max.
- Jika elemen tersebut lebih kecil dari min, maka min akan diganti dengan elemen tersebut.
- Jika elemen tersebut lebih besar dari max, maka max akan diganti dengan elemen tersebut.

Parameter

Parameter adalah variabel yang digunakan untuk menerima nilai yang akan digunakan dalam sebuah fungsi. Parameter digunakan sebagai input untuk fungsi dan dapat digunakan dalam tubuh fungsi untuk melakukan operasi yang diinginkan.

Contoh :

```
def tambah(a, b):  
    hasil = a + b  
    return hasil  
  
hasil = tambah(4, 5)  
print(hasil)
```

Output : 9

Dalam contoh di atas, kita memiliki sebuah fungsi bernama tambah yang menerima dua parameter, a dan b. Fungsi ini akan menambahkan dua nilai tersebut dan mengembalikan hasilnya. Kemudian, kita memanggil fungsi tambah dengan mengirimkan dua nilai, yaitu 4 dan 5. Hasil dari operasi tersebut kemudian disimpan dalam variable hasil dan dicetak ke layar.

Return Value

Secara umum ada dua jenis fungsi berdasarkan hasil yang dikeluarkan oleh fungsi, yaitu:

1. Fungsi yang tidak mengembalikan nilai (void function)

Contoh:

```
def print_nama():  
    print("Halo, namaku Jose")  
  
print_nama()
```

Output : **Halo, namaku Jose**

2. fungsi yang mengembalikan nilai.

Contoh : Fungsi untuk menghitung luas segitiga. Luas_segitiga

```
def luas_segitiga(alas, tinggi):  
    area = 0.5 * alas * tinggi  
    return area  
  
alas = 5  
tinggi = 7  
  
area = luas_segitiga(alas, tinggi)  
print("Luas segitiga:", area)
```

Output: **Luas segitiga: 17.5**

- Fungsi luas_segitiga diatas memiliki dua parameter, yaitu alas dan tinggi yang akan digunakan untuk menghitung luas segitiga dengan rumus luas segitiga = $0.5 \times \text{alas} \times \text{tinggi}$.
- Hasil dari perhitungan akan disimpan dalam variable area dan kemudian dikembalikan oleh fungsi.
- variabel alas dan tinggi diinisialisasi dengan nilai 5 dan 7. Kemudian, fungsi Luas_segitiga dipanggil dengan memasukkan nilai dari alas dan tinggi sebagai argumen. Hasil dari

perhitungan akan disimpan dalam variable area dan kemudian dicetak menggunakan fungsi print

Anonymous Function (Lambda)

Lambda adalah cara untuk membuat fungsi anonim atau fungsi tanpa nama. Lambda digunakan untuk membuat fungsi sederhana yang dapat dijadikan argumen untuk fungsi lain.

Contoh :

Program kalkulator menggunakan lambda

```
calculator = {
    "+": lambda x, y : x + y,
    "-": lambda x, y : x - y,
    "/": lambda x, y : x / y,
    "*": lambda x, y : x * y,
}

x = float(input("Masukkan x: "))
y = float(input("Masukkan y: "))
op = input("Masukkan Operator: ")

print(calculator[op](x,y))
```

Setelah melihat program kalkulator diatas, kita dapat mengetahui beberapa kelebihan lambda, yaitu :

- **Kode lebih ringkas:** Dengan menggunakan lambda, kita dapat menuliskan kode yang lebih ringkas dan mudah dibaca. Kita tidak perlu membuat fungsi terpisah untuk setiap operasi matematika, tetapi kita dapat menggunakan lambda untuk menuliskan fungsi secara singkat.
- **Mudah untuk dibuat dan dipanggil :** Dengan menggunakan lambda, kita dapat membuat fungsi secara mudah dan langsung dipanggil tanpa perlu mendefinisikannya terlebih dahulu. Hal ini sangat berguna ketika kita hanya perlu menggunakan fungsi sekali atau membuat fungsi yang sederhana.
- **Mudah untuk disimpan dalam variabel atau data struktur :** Kita dapat menyimpan fungsi lambda dalam variabel atau data struktur, seperti dictionary dalam program di atas. Hal ini membuat kode lebih mudah dipahami dan diorganisir.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 4.1

```
def cek_angka(x, y, z):  
    if x != y and x != z and y != z and (x + y == z or y + z == x or z + x == y):  
        return True  
    else:  
        return False  
  
x = int(input("Masukkan bilangan pertama: "))  
y = int(input("Masukkan bilangan kedua: "))  
z = int(input("Masukkan bilangan ketiga: "))  
hasil = cek_angka(x, y, z)  
print(hasil)
```

```
Masukkan bilangan pertama: 2  
Masukkan bilangan kedua: 4  
Masukkan bilangan ketiga: 6  
True
```

Penjelasan :

- Fungsi cek_angka saya buat dengan menggunakan 3 parameter, yaitu x, y, dan z
- Kondisi pertama ($x \neq y$ and $x \neq z$ and $y \neq z$) memeriksa apakah ketiga bilangan berbeda satu sama lain.
- Kondisi kedua ($(x + y == z$ or $y + z == x$ or $z + x == y)$) memeriksa apakah penjumlahan dua bilangan mana pun sama dengan bilangan ketiga.
- Lalu saya meminta inputan user untuk memasukkan angka ke dalam parameter, jika diambil dua parameter dan dijumlahkan hasilnya sama dengan parameter lainnya maka akan menghasilkan output true, tetapi jika tidak maka outputnya false.

SOAL 4.2

```
def cek_digit_belakang():
    x = int(input("Input angka pertama: "))
    y = int(input("input angka kedua: "))
    z = int(input("Input angka ketiga: "))

    ca1 = x % 10
    ca2 = y % 10
    ca3 = z % 10

    if (ca1 == ca2) or (ca1 == ca3) or (ca2 == ca3):
        return True
    else:
        return False

print(cek_digit_belakang())
```

```
Input angka pertama: 145
input angka kedua: 5
Input angka ketiga: 100
True
```

Penjelasan :

- Fungsi cek_digit_belakang diatas akan mengecek apakah minimal dua dari tiga digit paling kanan dari ketiga bilangan tersebut sama.
- Pertama, fungsi akan mendapatkan digit paling kanan dari setiap bilangan dengan menggunakan operasi modulo % 10
- Kedua, fungsi akan memeriksa apakah minimal dua dari tiga digit paling kanan dari ketiga bilangan tersebut sama. Jika ya, maka fungsi akan segera mengembalikan true
- Jika tidak, maka fungsi akan mengembalikan nilai false.

SOAL 4.3

```
celsius_to_fahrenheit = lambda c: (9/5) * c + 32
celsius_to_reamur = lambda c: 0.8 * c

celsius_input = float(input("Masukkan suhu dalam Celsius: "))

fahrenheit_output = celsius_to_fahrenheit(celsius_input)
print("Hasil konversi ke Fahrenheit:", fahrenheit_output)
reamur_output = celsius_to_reamur(celsius_input)
print("Hasil konversi ke Reamur:", reamur_output)
```

```
Masukkan suhu dalam Celsius: 100
Hasil konversi ke Fahrenheit: 212.0
Hasil konversi ke Reamur: 80.0
```


Penjelasan :

- Saya menggunakan fungsi lambda `celsius_to_fahrenheit` dan `celsius_to_reamur` untuk mengubah suhu dalam Celsius menjadi Fahrenheit dan Reamur.
- Lalu saya memasukkan rumus konversi suhu dari Celsius ke Fahrenheit adalah $(9/5) * \text{Celsius} + 32$. Sedangkan untuk konversi suhu dari Celsius ke Reamur, rumusnya adalah $0.8 * \text{Celsius}$.
- Program diatas kemudian akan meminta user memasukkan suhu dalam celcius lalu mengkonversinya ke dalam fahrenheit dan reamur.

Link Github : <https://github.com/Josephinemrc/Tugas4-PrakAlpro.git>