

Python Fundamental

Ch.7 : Function

Instruktur : Ahmad Rio Adriansyah
Nurul Fikri Komputer

Pemrograman Tak Terstruktur

- Setelah mempelajari dasar dasar dari pemrograman (tipe data, variabel, input-output, control, looping), kita sudah bisa membuat program python
- Tahapan ini disebut **unstructured programming** karena semua kode program dituliskan dalam satu tempat

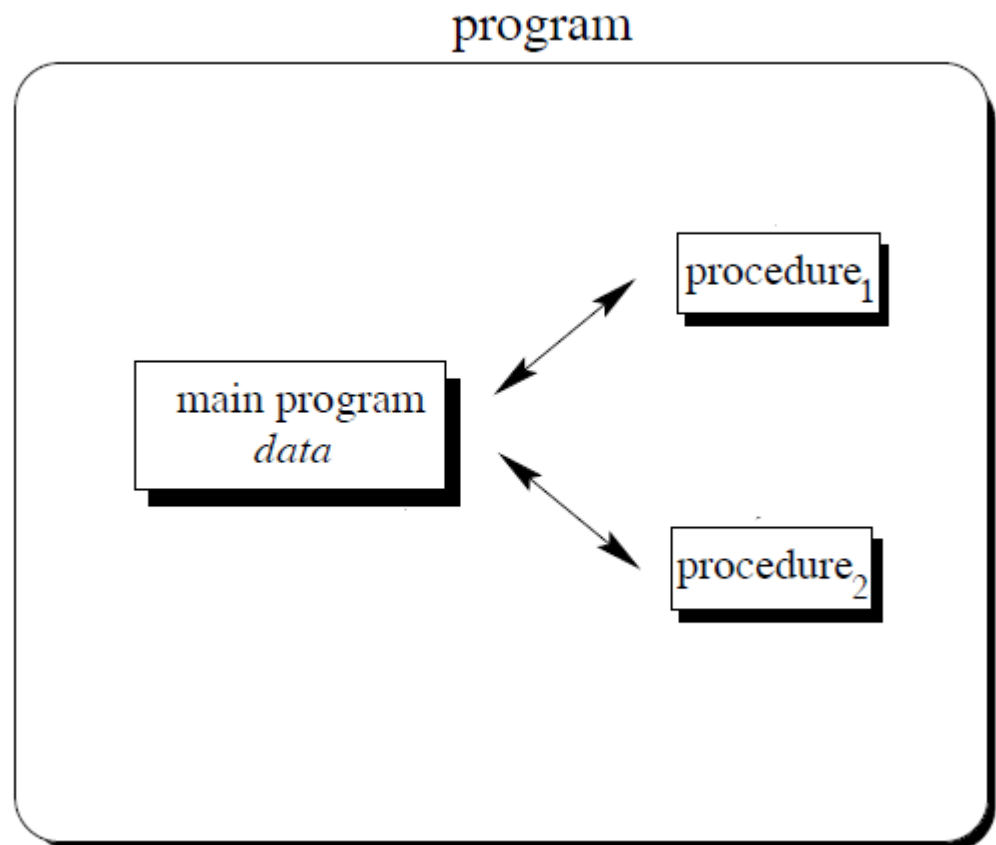
- Buat program yang bisa untuk
 - Menginput nama
 - Menginput tahun lahir
 - Menghitung umur
 - Mencetak informasi nama dan umur

```
>>> nama = input('Masukkan nama !')  
>>> tahun = input('Masukkan tahun lahir !')  
>>> umur = 2019-tahun  
>>> print(nama, 'berumur', umur, 'tahun')
```

- Buat program yang bisa untuk
 - Menginput nama
 - Menginput tahun lahir
 - Menghitung umur
 - Mencetak informasi nama dan umur
 - Menginput nama ayah
 - Menginput tahun lahir ayah
 - Menghitung umur ayah
 - Mencetak informasi nama dan umur ayah
 - ... ibu ... kakak ... adik ... suami ... istri

Pemrograman Prosedural

- Beberapa tahapan proses yang sering dilakukan bersama bisa dibuat dalam sebuah fungsi / prosedur
- Mengurangi kerumitan main program



Fungsi

- Prosedur dan fungsi pada python dikenal sebagai satu nama yaitu **function**
- Yang membedakan adalah ada atau tidaknya parameter input dan pengembalian nilai
 - Fungsi : mengandung parameter input (argumen) / pengembalian nilai
 - Prosedur : tidak mengandung input ataupun pengembalian nilai
- Python mengenal 2 jenis fungsi yaitu fungsi yang terpasang pada python dan fungsi yang didefinisikan oleh user

Fungsi Bawaan Python

Diantaranya :

- **len()**
- **list()**
- **abs()**
- **ascii()**
- **bin()**
- **str()**
- **float()**
- **range()**
- **max()**
- **sum()**
- **sorted()**
- **set()**
- **zip()**
- **eval()**

Fungsi Buatan User

- Format

```
>>> def nama(argumen) :  
...     expression  
...     return x #if needed
```

- Call

```
>>> x = nama(argumen)
```

Contoh

```
>>> def masukkan_nama():  
...     nama = input('Masukkan nama !')  
...     tahun = input('Masukkan tahun lahir !')  
...     umur = 2018-tahun  
...     print(nama, 'berumur', umur, 'tahun')  
>>> print('Data diri anda')  
>>> masukkan_nama()
```

Contoh

```
>>> def masukkan_nama(tahun_ini):  
...     nama = input('Masukkan nama !')  
...     tahun = input('Masukkan tahun lahir !')  
...     umur = tahun_ini - tahun  
...     print(nama, 'berumur', umur, 'tahun  
pada tahun',tahun_ini)  
>>> print('Data diri anda')  
>>> masukkan_nama(2010)
```

Contoh

```
>>> def kuadrat(x):  
...     return x*x  
>>>  
>>> y = kuadrat(3)  
>>> print(y)
```

Contoh

```
>>> def kali(x,y):  
...     return x*y
```

```
>>>
```

```
>>> kali(3,5)
```

```
>>> kali(2.71, 3.14)
```

```
>>> kali('betul ',3)
```

Latihan

- Setelah mempelajari materi tentang percabangan (if-then-else) anda bisa mengkategorikan suatu kondisi dikatakan sejuk atau panas jika diberikan sebuah nilai suhu
- Misalnya kondisi **sejuk** jika di bawah 20 derajat celsius, jika diantara 20 dan 35 dikatakan **biasa**, dan di atas itu dikatakan **panas**
- Buat sebuah fungsi **status()** yang inputannya berupa suhu (dalam celsius) dan mengeluarkan informasi apakah suhu saat ini sejuk, biasa, atau panas

Latihan

- 1 derajat (lintang/bujur) setara dengan 111,322 km
- 1 derajat terdiri dari 60 menit atau 3600 detik
- Buat sebuah fungsi yang mengkonversi derajat, menit, detik ke km dan sebaliknya.

Latihan

- Coba dipikirkan sebuah pekerjaan yang rutin dilakukan yang bisa dibuat fungsinya

Quiz

- Misalnya sebuah fungsi memerlukan parameter input sebanyak 2 buah
- Apa yang terjadi jika fungsinya dipanggil dengan hanya memberikan 1 argumen?
- Apa yang terjadi jika fungsinya dipanggil dengan 3 buah argumen?

Argumen Default dan Optional

- Argumen default

argumen dari suatu fungsi yang sudah diberikan nilai default (nilai standard yang diberikan jika fungsinya dipanggil tanpa diberikan nilai untuk argumen tersebut)

- Argumen optional

argumen dari fungsi yang diberikan berupa sekumpulan tuple (himpunan berurut)

Argumen Default

- Format

```
>>> def nama (argumen = value):  
...     expression
```

Argumen Default : Contoh

```
>>> def profesi(nama,pekerjaan = 'Freelancer'):
...     print(nama, 'adalah seorang',pekerjaan)
...
>>> profesi('Andi' , 'Doktor')
>>> profesi('Budi' , 'Peneliti')
>>> profesi(nama = 'Candra', pekerjaan =
'Karyawan BUMN')
>>> profesi('Deni')
```

Argumen Optional

- Format

```
>>> def nama(*args):  
...     expression
```

Argumen Optional : Contoh

```
>>> def kelompok(*anggota):  
...     n = len(anggota)  
...     print('Kelompok ini ada %d anggota, yaitu:' % n)  
...     for nama in anggota:  
...         print(nama)  
>>>  
>>> kelompok('Andi', 'Budi')  
>>> kelompok('Andi', 'Budi', 'Chandra')
```

Quiz

Apa yang terjadi jika diinputkan 0,1,2,3,4, atau 5 buah argumen pada fungsi f berikut?

```
>>> def f(a,b,c=1,*d):
```

```
...     print(a,b,c,d)
```

```
>>>
```

```
>>> f()
```

```
>>> f(1)
```

```
>>> f(1,2)
```

```
>>> f(1,2,3)
```

```
>>> f(1,2,3,4)
```

Keyword Argumen

- Kita bisa membuat dan memanggil fungsi dengan argumen berupa keyword=value
- Contohnya sudah ada pada contoh-contoh sebelumnya

```
>>> profesi(nama = 'Candra', pekerjaan = 'Karyawan BUMN')
```
- Jika urutannya dibalik, tidak masalah

```
>>> profesi(pekerjaan = 'Karyawan BUMN', nama = 'Candra')
```


Keyword Argumen : Contoh

```
>>> def keyword(nama= 'arasy', pekerjaan=
'CKO', hobi= 'main'):
...     print('Ini adalah %s' %nama)
...     print('yang pekerjaannya %s',
%pekerjaan)
...     print('dan kesukaannya %s', %hobi)
>>>
```

Keyword Argumen : Contoh

- Fungsi keyword tadi dapat dipanggil dengan

```
>>> keyword('a', 'b', 'c')
```

```
>>> keyword(pekerjaan= 'supir')
```

```
>>> keyword()
```

```
>>> keyword('rio', hobi= 'baca')
```

```
>>> keyword(hobi= 'ngoprek', nama= 'ario',  
pekerjaan= 'opreker')
```

Keyword Argumen : Contoh

- Fungsi keyword tadi **tidak dapat dipanggil** dengan

```
>>> keyword(nama= 'rio', 'baca')
```

```
#non kwarg setelah kwarg
```

```
>>> keyword('ario', nama= 'arasy')
```

```
#duplikat argumen
```

```
>>> keyword(alamat = 'depok')
```

```
#tidak ada kwarg alamat pada fungsinya
```

Argumen

- Argumen dicocokkan oleh python berdasarkan posisi dan nama
- Tidak bisa ada 2 nilai yang dimasukkan ke parameter yang sama
- Jika ada argumen default pada fungsinya, maka argumen tersebut dapat dipanggil ataupun tidak
 - Jika dipanggil nilainya akan sesuai dengan inputan barunya
 - Jika tidak dipanggil nilainya akan menggunakan nilai default

Keyword Argumen

- Lebih jauh lagi, bisa dibuat sebuah argumen opsional yang inputannya berupa keyword=value, seperti dictionary python
- Format

```
>>> def nama(**kwargs):  
...     expression
```

Keyword Argumen : Contoh

```
>>> def keyargs (a, b=0, **rest):
```

```
...     print(a,b,rest)
```

```
>>>
```

```
>>> keyargs(1,c=3)
```

```
>>> keyargs(1,2,3)
```

```
>>> keyargs(1, c=3, d=4, e=5, b=2)
```

Fungsi Rekursif

- Sebuah fungsi yang memanggil dirinya sendiri
- Digunakan jika fungsinya dapat dibelah menjadi fungsi yang mirip
- Contoh :
 - Fungsi faktorial
 - Fungsi deret fibonacci
 - Fungsi menghitung determinan matriks
 - Binary search

Fungsi Rekursif

- Fungsi rekursif yang baik memiliki **titik henti** saat suatu kondisi terpenuhi (*recursion termination*), yaitu bagian fungsi yang saat dipanggil mulai mengembalikan nilai, tidak lagi memanggil dirinya sendiri

Intermezzo

- Rekursif sering menjadi jokes diantara orang IT
- Banyak nama populer di dunia pemrograman yang rekursif (atau direkursifkan)
 - GNU = GNU Not Unix
 - PHP = PHP Hypertext Processor
 - WINE = WINE Is Not Emulator
 - dll

Faktorial

- Faktorial dari suatu bilangan bulat positif n adalah sebuah perkalian dari 1,2,3 hingga n
- Disimbolkan dengan tanda seru (!) setelah bilangan yang difaktorialkan

$$n! = 1 \times 2 \times 3 \times \dots \times (n-2) \times (n-1) \times n$$

- Contoh :
 - $3! = 6$
 - $5! = 120$

Latihan

- Buat fungsi yang menghitung nilai faktorial sebuah bilangan (tanpa rekursif)
- Hint : gunakan for loop atau while loop

Fungsi Faktorial Rekursif

$$n! = 1 \times 2 \times 3 \times \dots \times (n-2) \times (n-1) \times n$$

$$(n-1)! = 1 \times 2 \times 3 \times \dots \times (n-3) \times (n-2) \times (n-1)$$

- Berarti dapat dituliskan bahwa

$$n! = (n-1)! \times n$$

- Dengan titik henti

$$1! = 1$$

Latihan

- Buat fungsi yang menghitung nilai faktorial sebuah bilangan secara rekursif!

Quiz

- Selesaikan perhitungan berikut ini

$$25 - 55 + (85 + 65)$$

Percaya atau tidak, tapi jawabannya 5!

Fungsi Rekursif

- + Fungsi rekursif menghemat penulisan kode
- + Program lebih kecil dan lebih mudah dibaca
- - Program berjalan lebih lambat dibandingkan fungsi yang sama yang non rekursif
- - Menghabiskan memori lebih banyak