

Bahasa Python

Part #2

- Seluruh materi di dalam PPT ini bersumber dari:

Modul PTI-B Python

Tim Materi PTI-B 2018/2019 Institut Teknologi Bandung
2019-03-19

Bahasa Python itu mudah...

- Tidak sulit mempelajari Bahasa Python, karena strukturnya mirip dengan bahasa pemrograman lainnya (C, C++, Pascal, Java, dll).
- Asalkan sudah mengerti algoritma penyelesaian sebuah persoalan, maka menuliskan kodenya ke dalam Bahasa Python tidaklah sukar.
- Yang penting banyak melakukan praktek, latihan mandiri, membaca berbagai tutorial di Internet, dll

Karakteristik Bahasa Python

- Python case-sensitive, artinya perbedaan huruf besar dan huruf kecil menyebabkan perbedaan makna.
- Python sangat memerhatikan indentasi dan pergantian baris. Kesalahan indentasi dapat menyebabkan gagal compile hingga kesalahan program.
- Variabel di python bersifat implisit dan dinamis. Artinya, sebuah variabel tidak perlu dideskripsikan tipe datanya. Namun di modul ini kita tetap mempelajari tipe data yang ada.

Input dan Output

Dalam python, program untuk menulis "Hello, World!" ke layar adalah seperti berikut:

```
print("Hello, World!")           # (1)

print("Ini program", end="")     # (2)
print(" pertama saya")          # (3)

# Ini adalah sebuah komentar.

# Semua yg ada setelah tanda pagar (#)
# akan diabaikan oleh interpreter.

'''
Selain itu, kita juga bisa membuat komentar
multi baris dengan tiga petik '''
```

- Contoh program yang menerima input dan menuliskan ulang yang dimasukkan:

```
S = input("Masukkan kalimat: ")           # 1
print("Anda memasukkan kalimat " + S)      # 2

N = int(input("Masukkan sebuah angka: "))   # 3
print("Jika angka Anda ditambah 5, hasilnya " + str(N + 5)) # 4
```

Tipe Data

bool	Boolean	True atau False
int	Bilangan bulat	seluruh bilangan bulat
float	Bilangan real	seluruh bilangan real
string	Teks	kumpulan karakter

Contoh penggunaan:

```
B = True          # Boolean
I = 123456789012  # Bilangan bulat
R = 2.331973      # Bilangan real
S = "abc"         # Teks
S = 'def'
```

Operator

1.4.1 Operator Aritmatika

Operator	Deskripsi	Contoh
+	Penjumlahan	2 + 3 bernilai 5
-	Pengurangan	1 - 8 bernilai -7
*	Perkalian	5 * 6 bernilai 30
/	Pembagian	13 / 5 bernilai 2.6
//	Pembagian (dibulatkan ke bawah)	13 // 5 bernilai 2
%	Sisa Bagi / Modulo	13 % 5 bernilai 3

1.4.2 Operator Assignment

Operator	Deskripsi	Contoh
=	Assignment	N = 5
+=	Penjumlahan	N += 5, N akan ditambah 5.
-=	Pengurangan	N -= 5, N akan dikurang 5.
*=	Perkalian	N *= 5, N akan dikali 5.
/=	Pembagian	N /= 5, N akan dibagi 5.
//=	Pembagian (dibulatkan ke bawah)	N //= 5, N akan dibagi 5 (dibulatkan ke bawah).
%=	Sisa Bagi / Modulo	N %= 5, N akan dimodulo 5.

1.4.3 Operator Relasional

Operator	Deskripsi	Contoh True	Contoh False
==	Sama dengan	2 == 2	2 == 3
!=	Tidak sama dengan	3 != 2	3 != 3
<	Kurang dari	2 < 3	2 < 2
>	Lebih dari	3 > 2	3 > 2
<=	Kurang dari sama dengan	2 <= 2	1 <= 3
>=	Lebih dari sama dengan	6 >= 5	2 >= 4

1.4.4 Operator Logika

Operator	Deskripsi	Contoh True	Contoh False
and	Dan	(1 < 2)and (3 == 3)	(1 == 2)and (3 == 3)
or	Atau	(1 < 2)or (4 == 3)	(3 < 2)or (2 == 3)
not	Negasi	not (3 < 2)	not (1 > 2)

Pencabangan

- Misal akan membuat program untuk memeriksa apakah sebuah bilangan bernilai positif

```
print("Masukkan nilai N: ", end="")
N = int(input())

if (N > 0):
    print(str(N) + " bilangan positif")
```

Lalu, jika kita ingin menuliskan kebalikannya:

```
...
    if (N > 0):
        print(str(N) + " bilangan positif")
    else: # N <= 0
        print(str(N) + " bilangan bukan positif")
...
```

Namun, kita tahu kadang bilangan bisa nol atau negatif, jadi perlu kita tambahkan:

```
...  
    if (N > 0):  
        print(str(N) + " bilangan positif")  
    elif (N < 0):  
        print(str(N) + " bilangan negatif")  
    else: # N == 0  
        print(str(N) + " bilangan nol")  
...
```

Perhatikan juga kalau kita bisa membuat else ini berulang sampai yang kita mau. Selain itu, kita juga bisa meletakkan `if` di dalam `if`.

```
...
    if (N >= 0):
        if (N > 0):
            print(str(N) + " bilangan positif")
        else: # N == 0
            print(str(N) + " bilangan nol")
    else: # N < 0
        print(str(N) + " bilangan negatif")
...
```

Latihan

1. Tulis program yang menerima input bilangan bulat lalu menuliskan apakah bilangan tersebut genap atau ganjil.
2. Tulis program yang menerima input suhu zat cair (T) dalam Celcius, lalu menentukan apakah wujudnya cair, padat, atau gas. (Ket: wujud gas jika $T \geq 100$, wujud padat jika $T \leq 0$, dan cair jika $0 < T < 100$)

Pengulangan

- While loop

Berikut adalah program yang menerima a dan b dan menuliskan $a, a + 1, a + 2, \dots, b - 1, b$.

```
a = int(input())  
b = int(input())  
i = a  
while (i <= b):  
    print(i)  
    i += 1
```

- For loop

Berikut program untuk mencetak 'Hello world' sebanyak 1000 kali

```
for i in range(1000):  
    print("Hello world")
```

Berikut adalah program yang menerima a dan b dan menuliskan $a, a + 1, a + 2, \dots, b - 1, b$.

```
a = int(input())  
b = int(input())  
for i in range(a, b+1):  
    print(i)
```


`range` dapat digunakan dengan satu, dua, atau tiga parameter. Untuk lebih jelasnya, perhatikan potongan kode berikut:

```
range(10)           # 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
range(2, 5)         # 2, 3, 4
range(7, 2, -1)     # 7, 6, 5, 4, 3
range(2, 8, 2)      # 2, 4, 6
```

- Perulangan bersarang (*nested loop*)

Perhatikan pula, perulangan dapat dilakukan di dalam perulangan. Sebagai contoh, berikut adalah program untuk membuat pola persegi.

```
n = int(input())
for i in range(n):
    for j in range(n):
        print('*', end=' ')
    print()
```

Latihan

- Buatlah program yang menerima input N buah bilangan bulat lalu menghitung nilai rata-ratanya.
- Buatlah program yang menerima input sejumlah karakter. Hitunglah berapa banyak huruf 'a'. Pembacaan data berakhir jika input yang dimasukkan adalah titik ('.').

Array

Array adalah variabel dengan satu buah nama, tetapi mengandung banyak nilai. Akses nilai-nilainya dilakukan dengan indeks.

Perhatikanlah contoh berikut!

Indeks	0	1	2	3	4	5	6	7	8	9
A	3	10	5	7	11	19	23	35	37	12

- $A[0] = 3$
- $A[1] = 10$
- $A[7] = 35$

Perhatikan pula bahwa indeks dimulai dari 0.

- Deklarasi array

Contoh deklarasi array:

- `x = [0 for i in range(n)]` membuat array berukuran n dengan isi 0.
- `x = ['*' for i in range(100)]` membuat array berukuran 100 dengan isi karakter '*'.

Untuk contoh tersebut array A yang terdefinisi adalah `A[0], A[1], A[2], ..., A[9]` Mengakses nilai indeks di luar batasan tersebut akan menyebabkan runtime error. Oleh karena itu, tentukanlah rentang indeks yang akan digunakan saat deklarasi dengan tepat (sesuai dengan kebutuhan).

```
tabel = [0 for i in range(10)]
```

Maka array tabel tersebut akan terdefinisi untuk indeks 0 sampai dengan indeks 9. Maka kita bisa melakukan instruksi

```
tabel[2] = int(input())
```

Jika diberikan 5 buah bilangan , dan kita perlu menyimpan bilangan tersebut pada tabel, kita bisa melakukan

```
tabel[0] = int(input('Masukkan nilai ke-0: '))  
tabel[1] = int(input('Masukkan nilai ke-1: '))  
tabel[2] = int(input('Masukkan nilai ke-2: '))  
tabel[3] = int(input('Masukkan nilai ke-3: '))  
tabel[4] = int(input('Masukkan nilai ke-4: '))
```

Namun, cara menginput tersebut kurang efisien. Akan lebih efisien jika kita menginput menggunakan perulangan (looping).

```
for i in range(5):  
    tabel[i] = int(input('Masukkan nilai ke-' + str(i) + ': '))
```

Latihan

- Buatlah program yang menerima input N buah bilangan bulat, simpan ke dalam array, lalu tentukan nilai rata-ratanya.
- Buatlah program yang menerima input N buah bilangan riil, simpan di dalam array, lalu tentukan nilai yang terbesar.

Fungsi

Apa itu fungsi? Fungsi adalah suatu bagian dari program yang mampu mengerjakan tugas atau operasi tertentu di luar program utama. Fungsi akan mengembalikan nilai sesuai algoritma yang diberikan.

Sebagai contoh, fungsi untuk menghitung kuadrat adalah sebagai berikut:

```
def kuadrat(x):  
    x2 = x * x  
    return x
```

Sebagai contoh, berikut program lengkap yang menerima input dan mengeluarkan kuadrat dari bilangan input.

```
def kuadrat(x):  
    x2 = x * x  
    return x2  
  
n = int(input())  
n2 = kuadrat(n)  
print(n2)
```

Sebuah fungsi juga dapat menerima lebih dari satu parameter. Selain itu, fungsi juga dapat melakukan hal-hal layaknya program biasa, namun tidak dapat mengubah variabel di program utama. Sebagai contoh, berikut adalah fungsi yang menghitung nilai a^b .

```
def pangkat(a, b):  
    # asumsi b >= 0  
    c = 1  
    for i in range(b):  
        c *= a  
    return c
```

Prosedur

Prosedur sebenarnya sama seperti fungsi, namun tidak ada kembalian.

```
def tulis_menu(pil1, pil2, pil3):  
    print("Menu:")  
    print("1. " + str(pil1))  
    print("2. " + str(pil2))  
    print("3. " + str(pil3))  
    print("Masukkan pilihan: ")  
  
tulis_menu("Burger", "Ayam Geprek", "Mie Instan")  
pilihan_makanan = int(input())  
  
tulis_menu("Jus Alpukat", "Thai Tea", "Teh Tarik")  
pilihan_minuman = int(input())  
  
tulis_menu("Kentang", "Krupuk", "Abon")  
pilihan_tambahan = int(input())
```

Latihan

- Buatlah fungsi untuk menentukan nilai minimum dari dua buah bilangan bulat. Misalnya $a = 3$, $b = 5$, maka $\min(a,b) = 3$.
- Buatlah fungsi untuk menghitung luas sebuah lingkaran. Parameter fungsi adalah jari-jari lingkaran.
- Buatlah fungsi untuk menghitung perpangkatan a^n , a dan n adalah bilangan bulat.
- Buatlah prosedur untuk menerima input bilangan 1 sampai 10, lalu menampilkan huruf romawinya. Misalkan $x = 8$, maka luaran: VIII