

# STRUKTUR PEMROGRAMAN PYTHON

## 1. Aturan Penulisan

Program-program yang ditulis dalam Python secara khas jauh lebih pendek dibandingkan dengan program-program C atau C++, karena beberapa pertimbangan:

- tipe data tingkat tinggi digunakan untuk menyatakan operasi kompleks dalam suatu statemen tunggal;
- pengelompokan statemen telah selesai dengan indentasi sebagai pengganti dari pengurangan mulai dan akhiran;
- tidak ada deklarasi-deklarasi argumentasi atau variabel yang diperlukan.

### 1.1. Indentasi

Bahasa pemrograman Python adalah bahasa pemrograman yang mudah dibaca dan terstruktur, hal ini karena digunakannya sistem indentasi. Yaitu memisahkan blok - blok program dengan susunan indentasi. Jadi untuk memasukan sub - sub program dalam suatu blok, sub - sub program tersebut diletakkan satu atau lebih spasi dari kolom suatu blok program.

Python memiliki sedikit perbedaan pada cara penulisan program dengan bahasa pemrograman yang lain seperti C/Java. Kalau pada C/Java menggunakan tanda kurung sebagai pemisah blok program, di Python kita hanya menggunakan spasi sebagai pemisah blok program yang biasa disebut sebagai Indentasi. Karena Python menjalankan perintah secara berurutan, maka kita harus pintar menyusun perintah agar mendapatkan hasil seperti yang diinginkan.

```
if a = b :  
    |print a, 'sama dengan', b  
else :  
    |print a, 'tidak sama dengan', b
```

Pada contoh diatas kita dapat melihat jika suatu kondisi  $a = b$  dipenuhi maka program akan menjalankan baris perintah yang ada di dalam suatu blok kondisi tersebut, yang ditandai dengan penggunaan satu spasi atau lebih dari blok kondisi sebelumnya, dalam contoh diatas perintah yang akan dilaksanakan jika suatu kondisi diatasnya terpenuhi menggunakan dua (2) spasi, sedangkan pada

pernyataan `else`, menggunakan satu spasi. Perbedaan penggunaan spasi ini tidak dianjurkan meskipun dalam program Python dibenarkan, karena struktur program akan lebih sulit dibaca. Seharusnya blok-blok program diatas adalah sebagai berikut :

```
if a = b :
    |print a, 'sama dengan', b #Menggunakan 2 spasi
else :
    |print a, 'tidak sama dengan', b #Menggunakan 2 spasi
```

## 1.2. Baris Perintah

Dalam Python, program tersebut dapat langsung dijalankan sebagai berikut:

```
print "Hello Python!"
```

Hasilnya akan menampilkan string "Hello Python!" sebagai berikut : Hello Python!

Bandingkan dengan Kedua bahasa pemograman di bawah ini :

Java :

```
class helloPython {
    public static void main(String argsp[]) {
        System.out.println("Hello Python"); }
}
```

C/C++:

```
# Untuk C
#include <stdio.h>
int main() {
    printf("Hello Python!"); return 0;
}

#Untuk C++
#include <iostream.h>
int main() {
    cout << "Hello Python!";
}
```

tanda kutip dua (""), yang berarti tempat nilai string diletakkan pada program ini tidak akan ditampilkan pada layar.

Sebuah kalimat perintah adalah sebuah instruksi yang dapat dieksekusi oleh interpreter Python. Kita telah melihat dua jenis kalimat perintah, yakni print dan pendeklarasian nilai.

Pada saat Anda mengetikkan kalimat perintah pada prompt perintah, maka python mengeksekusinya dan langsung menampilkan hasilnya. Jika ada, hasil dari perintah print adalah sebuah nilai. Pendeklarasian nilai tidak menampilkan hasil.

Pada sebuah script biasanya berisikan beberapa kalimat perintah. Jika lebih dari satu kalimat perintah, hasilnya akan tampil sesuai dengan kalimat perintah yang dieksekusi.

Contohnya:

```
>>> print 1
>>> x = 2
>>> print x
```

Menghasilkan hasil;

```
1
2
```

Sekali lagi, sebuah pendeklarasian nilai tidak menghasilkan output.

### 1.3. Keterangan Program

Dalam proses debugging, suatu keterangan / komentar instruksi program sangat berguna sekali dalam pembacaan suatu kode. Pada umumnya komentar berisi keterangan tentang kegunaan suatu fungsi itu. Sintaksnya adalah tanda kres atau tanda pagar "#". Setelah meletakkan tanda tersebut, kita dapat mengetikkan kalimat apa saja yang berhubungan dengan suatu instruksi perintah, sebab apapun kalimat tersebut tidak akan mempengaruhi jalannya program dan tidak akan di proses oleh interpreter. Jika interpreter menemukan tanda ini maka mulai tanda ini sampai akhir baris akan dianggap sebagai keterangan.

```
print Hello! #Mencetak string "Hello!" ke layar.
# belajar python
```

## 2. Variabel

Sebuah variabel adalah sebuah nama yang mempunyai sebuah nilai. Pendeklarasian kalimat membuat sebuah variabel - variabel baru dan memberinya nilai.

```
>>> a = "belajar Python"  
>>> b = 5  
>>> phi = 3.14
```

Pada contoh di atas, pendeklarasian tersebut menciptakan 3 variabel baru. Pendeklarasian pertama, menunjukkan string "belajar Python" ke sebuah variabel yang bernama a. Kedua, variabel b diberi nilai 5 sebagai integer. Dan yang terakhir variabel phi diberi nilai 3.14 sebagai nilai pecahan.

Cara yang umum untuk pemberian nama variabel adalah dengan tanda panah menunjuk ke nilai variabel tersebut. Jenis ini dinamai dengan state diagram karena menunjukkan nilai - nilai yang merupakan nilai dari variabel - variabel tersebut, contohnya :

```
a => "belajar Python"  
b => 5  
phi => 3.14
```

perintah print juga berlaku untuk kalimat di atas.

```
>>> print a  
Belajar Python  
>>> print b  
5  
>>> print phi  
3.14
```

### 2.1. Nama Variabel

Pada umumnya, programmer memakai nama variabel sesuai dengan keterangan isi dari variabel tersebut dan variabel juga merupakan simbol yang mewakili nilai tertentu. Pembuatan variabel dalam python sangat sederhana. Berikut adalah ketentuan mengenai variabel dalam python,

- Variabel tidak perlu dideklarasikan mempunyai tipe data tertentu

- Jenis data dalam variabel dapat berubah-ubah
- Penulisan variabel harus diawali dengan huruf, dan untuk karakter selanjutnya bisa berupa huruf atau angka
- Dapat berupa huruf Kapital, tetapi bersifat case-sensitive, nama Kapital dengan kapital adalah variabel yang berlainan.
- Penulisan variabel tidak boleh dipisah oleh <spasi>
- Untuk variabel yang terdiri dari 2 suku kata, dapat dipisah dengan simbol underscore ( \_ ) seperti nama\_saya, nama\_variabel\_nama.

Statemen yang tidak boleh dijadikan nama variabel adalah keywords pada Python.

Contoh :

```
>>> 123satu = "angka"
```

variabel 123satu adalah penamaan variabel tidak benar karena diawali dengan sebuah angka.

```
>>> lebih$ = 50000
```

lebih\$ juga tidak benar karena terdapat karakter yang tidak semestinya ada dalam penamaan variabel.

## 2.2. Keyword / Kata Kunci

Kata kunci mendefinisikan aturan - aturan dan struktur bahasa, dan mereka tidak dapat digunakan sebagai nama variabel.

Python mempunyai 28 kata kunci:

and	Continue	else	for	Import	not	Raise
assert	def	except	from	In	or	return
break	del	exec	global	Is	pass	Try
class	elif	finally	if	Lambda	print	while

Anda mungkin ingin menyimpan daftar - daftar ini, pada saat interpreter mengeluarkan kesalahan sintaks dari salah satu nama variabel Anda dan Anda tidak mengetahui penyebabnya, lihat mereka pada daftar ini.

### 2.3. Mengevaluasi Ekspresi

Sebuah ekspresi adalah kombinasi dari nilai-nilai, variabel-variabel, dan operator-operator. Jika Anda mengetikkan sebuah ekspresi pada modus baris perintah, interpreter langsung mengevaluasinya dan menampilkan hasilnya.

```
>>> 2 + 3
5
```

Sebuah nilai juga dikenal sebagai ekspresi, begitu juga dengan variabel.

```
>>> 55
55
>>> x
6
```

Mengevaluasi sebuah ekspresi tidak sama dengan mencetak sebuah nilai.

```
>>> kalimat = "python adalah bahasa pemrograman"
>>> kalimat
"python adalah bahasa pemrograman"
>>> print kalimat
python adalah bahasa pemrograman
```

Pada saat Python menampilkan nilai dari sebuah ekspresi, format yang sama juga akan digunakan untuk menampilkan sebuah nilai. Contoh kasus pada string, yang berarti tanda kutip 2(“) juga ditampilkan pada saat mengevaluasi sebuah ekspresi. Tetapi pada saat mengevaluasi perintah print, print menampilkan nilai dari string tersebut (tanpa tanda kutip 2).

Pada sebuah script, ekspresi dapat berupa kalimat perintah yang benar, tetapi tidak akan menghasilkan nilai dan tampilan hasil apapun.

Contohnya :

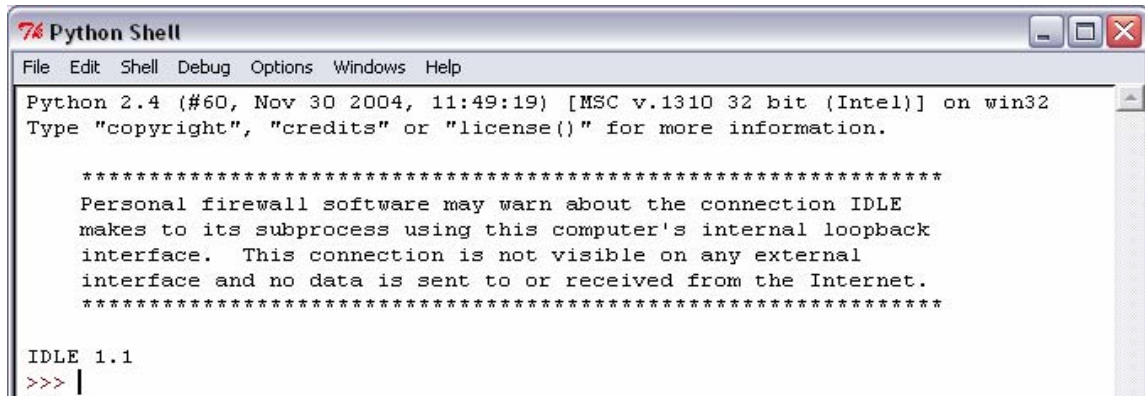
```
17
3.2
"Hello Python"
2 + 3
```

Tidak menghasilkan nilai apapun sama sekali.

### 3. Memulai Python

Bahasa ini sudah terinstal secara default di semua distro Linux. Jadi hanya menuju Terminal, lalu ketik python <ENTER> maka akan masuk ke lingkungan python. Jika pada Windows ketik python pada Command Prompt / run, diasumsikan bahwa python sudah terinstal pada sistem operasi Windows atau klik tombol start → Program → Python24 → IDLE(Python GUI).

Maka akan muncul prompt Interpreter dari Python,



Modus penulisan kode python dapat dilakukan dengan dua cara, yaitu

1. Menggunakan mode interaktif
2. Menggunakan Skrip

#### 3.1. Menggunakan Mode Interaktif

Menggunakan mode interaktif berarti kita bekerja menggunakan Prompt interpreter dari python. Penulisan kode python dilakukan per-statement, contohnya,

```
>>> print "Hello Python"
Hello Python
```

Statement “Hello Python” adalah hasil eksekusi dari perintah “print “Hello Python””. Dengan menggunakan prompt interpreter python, kita dapat melakukan uji coba setiap statement-statement atau fungsi-fungsi yang ada pada Python.

#### 3.2. Menggunakan Skrip

Dengan menyimpan perintah - perintah python dalam satu file, yang disebut selanjutnya sebagai script. Contohnya kita mengetikkan perintah-perintah python dengan menggunakan text

editor seperti vi, notepad, dll. Pada umumnya file yang berisi script tersebut di simpan dengan extension / akhiran ".py". Misalnya :

```
#Mencetak kata  
print "Contoh Program menggunakan skrip pada Python"
```

perintah tersebut kita simpan dengan nama file "contoh1.py", kemudian untuk mengeksekusinya kita panggil program interpreter python dengan cara :

file → open file yg akan dijalankan → lalu run (F5)

```
>>>
```

```
Contoh Program menggunakan skrip pada Python
```

maka interpreter mengeksekusi script tersebut dengan hasil Contoh Program menggunakan skrip pada Python.