

PEMROGRAMAN

BERORIENTASI

OBJEK

Pertemuan 1

Statement Terminator

- Bahasa Python tidak perlu menggunakan tanda titik koma (;) di akhir setiap perintah.
- Bahasa Python menggunakan karakter new line sebagai pemisah perintah. Tanda titik koma biasanya digunakan jika menulis beberapa perintah dalam 1 baris kode program. Namun penulisan seperti ini tidak disarankan karena akan kesulitan dalam membacanya.

Komentar

- Comment merupakan catatan yang bisa digunakan untuk menulis keterangan atau informasi dari kode tersebut.
- Comment tidak akan diproses oleh interpreter python.
- Comment akan mempermudah programmer lain untuk memahami maksud dari kode yang ditulis.
- Pembuatan comment di dalam Python diawali sebuah baris dengan tanda **hash** atau tanda pagar (#).

Variabel

- Digunakan untuk menyimpan suatu data. Dengan menyimpan data di variabel, maka data tersebut dapat digunakan kapanpun diperlukan dengan cara memanggilnya.
- Penulisan variabel dapat dimulai dengan huruf (A-Z atau a-z) atau dengan garis bawah (_) yang dikombinasi dengan angka dan huruf.
- Penamaan suatu variabel tidak boleh dimulai dengan angka.
- Python merupakan bahasa pemrograman yang bersifat *case sensitive*.

Variabel

- Kata kunci berikut ini tidak dapat dijadikan sebagai nama variabel, karena kata kunci tersebut merupakan instruksi-instruksi Python.

and	elif	global	or
assert	else	if	pass
break	except	import	print
class	exect	in	raise
continue	finally	is	return
def	for	lambda	try
del	from	not	while

Operator

- Adalah suatu symbol yang digunakan dalam program untuk melakukan suatu operasi atau manipulasi data atau operand.
- Contoh: $x+y$, symbol $+$ merupakan operator untuk melakukan operasi penjumlahan dua buah operand yaitu x dan y .
- Sifat operator:

Sifat	Keterangan	Contoh
Unary	Operator yang melibatkan 1 operand	$+1$, -2
Binary	Operator yang melibatkan 2 operand	$x+y$, $a-b$, $2*3$
Ternary	Operator yang melibatkan 3 operand	$(a+b)>c$

Operator Aritmatika

- Operator yang digunakan untuk berbagai operasi aritmatika atau operasi matematis.
- Operator binary dan unary aritmatik:

Operator	Keterangan	Contoh
+	Penjumlahan	1+2
-	Pengurangan	5-4
*	Perkalian	6*7
/	Pembagian	12/6
%	Sisa pembagian (modulus)	8%3
**	Perpangkatan	7**2
+	Plus	+4
-	Minus	-1
~	Negasi (<i>Inversion</i>)	~5

Prioritas Operator

- Setiap operator aritmatika mempunyai prioritas yang berbeda.
- Penulisan posisi operator yang berbeda akan menghasilkan output yang berbeda pula.
- Operator binary dan unary aritmatik:

Operator	Prioritas	Keterangan
**	Tertinggi (prioritas pertama)	
*, /, %	Prioritasnya sama (prioritas kedua)	Operasi dimulai dari kiri ke kanan
+, -	Terendah (prioritas ketiga)	Operasi dimulai dari kiri ke kanan

Operator Bitwise

- Operator yang digunakan untuk memanipulasi data dalam bentuk bit
- Operator bitwise:

Operator	Keterangan	Contoh
&	AND	$6 \& 7 = 6$
	OR	$5 8 = 13$
^	XOR (Exclusive OR)	$4 ^ 7 = 3$
<<	Geser bit ke kiri (left shift)	$5 << 1 = 10$
>>	Geser bit ke kanan (right shift)	$6 >> 1 = 3$

Operator Bitwise - AND

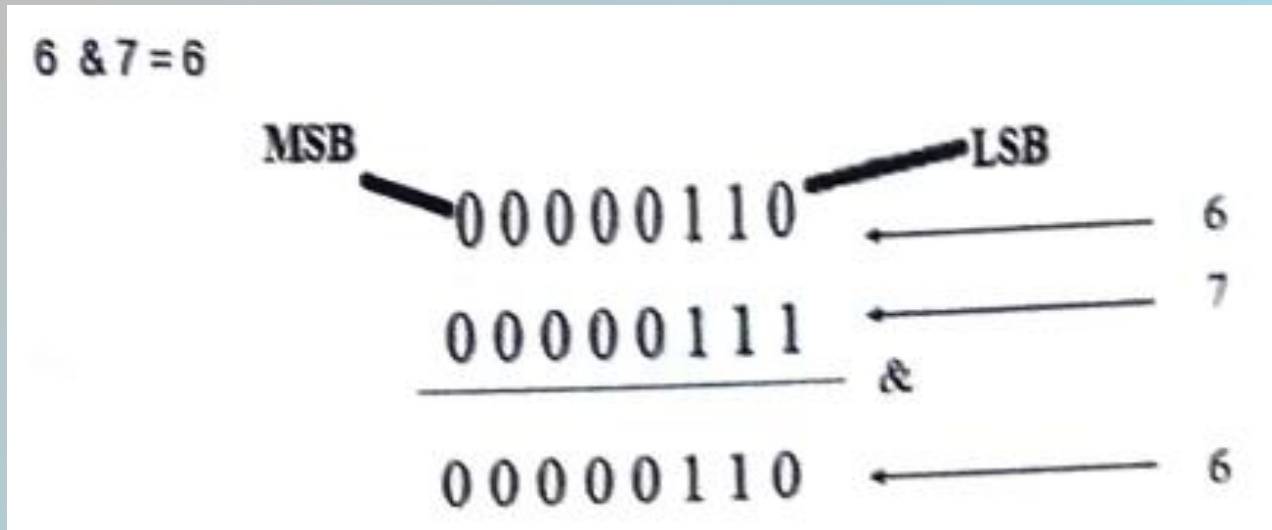
Operand 1 & Operand 2

- Hasil operasi per bit pada penggunaan operator AND, yaitu:
 - ✓ Output akan bernilai 0 apabila salah satu dari input/bit yang dibandingkan bernilai 0
 - ✓ Output akan bernilai 1 apabila semua input/bit yang dibandingkan bernilai 1
- Operator bitwise:

Bit Operand 1	Bit Operand 2	Bit Output
0	0	0
0	1	0
1	0	0
1	1	1

Operator Bitwise - AND

- Ilustrasi menggunakan operator logika AND:



Operator Bitwise - OR

Operand 1 | Operand 2

- Hasil operasi per bit pada penggunaan operator OR, yaitu:
 - ✓ Output akan bernilai 1 apabila salah satu dari input/bit yang dibandingkan bernilai 1
 - ✓ Output akan bernilai 0 apabila semua input/bit yang dibandingkan bernilai 0
- Operator bitwise:

Bit Operand 1	Bit Operand 2	Bit Output
0	0	0
0	1	1
1	0	1
1	1	1

Operator Bitwise - OR

- Ilustrasi menggunakan operator logika OR:

$$5 | 8 = 13$$

0 0 0 0 0 1 0 1	←	5
0 0 0 0 1 0 0 0	←	8
<hr/>		
0 0 0 0 1 1 0 1	←	13

Operator Bitwise - XOR

Operand 1 \wedge Operand 2

- Hasil operasi per bit pada penggunaan operator XOR, yaitu:
 - ✓ Output akan bernilai 1 apabila input/bit yang dibandingkan bernilai berlawanan (1-0 atau 0-1)
 - ✓ Output akan bernilai 0 apabila input/bit yang dibandingkan bernilai sama (0-0 atau 1-1)
- Operator bitwise:

Bit Operand 1	Bit Operand 2	Bit Output
0	0	0
0	1	1
1	0	1
1	1	0

Operator Bitwise - XOR

- Ilustrasi menggunakan operator logika XOR:

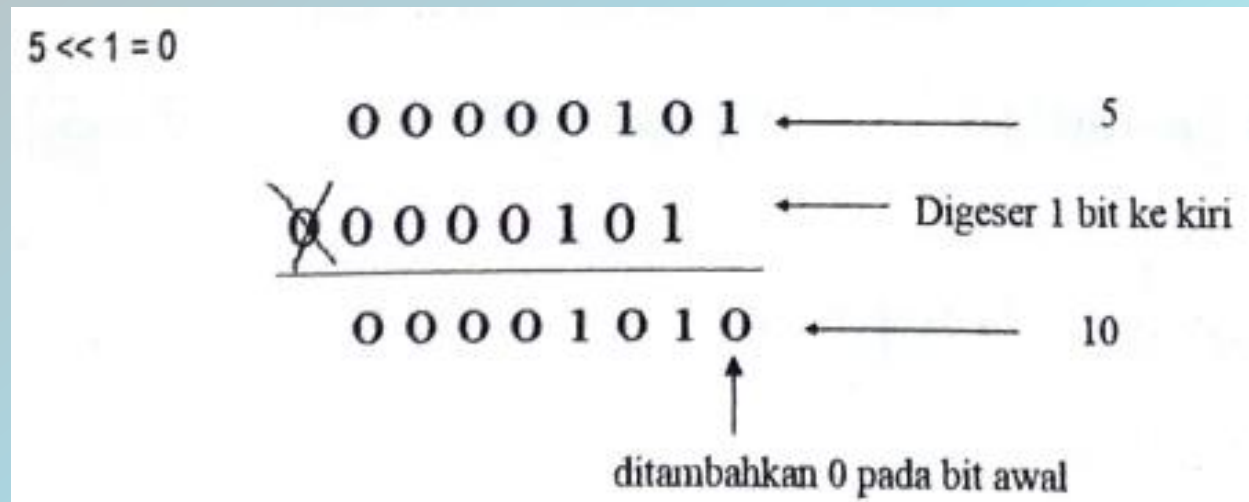
$4 \wedge 7 = 3$

0 0 0 0 0 1 0 0	←	4
0 0 0 0 0 1 1 1	←	7
<hr/>		
0 0 0 0 0 0 1 1	←	3

Operator Bitwise – Geser Bit ke Kiri

Operand << jumlah pergeseran bit

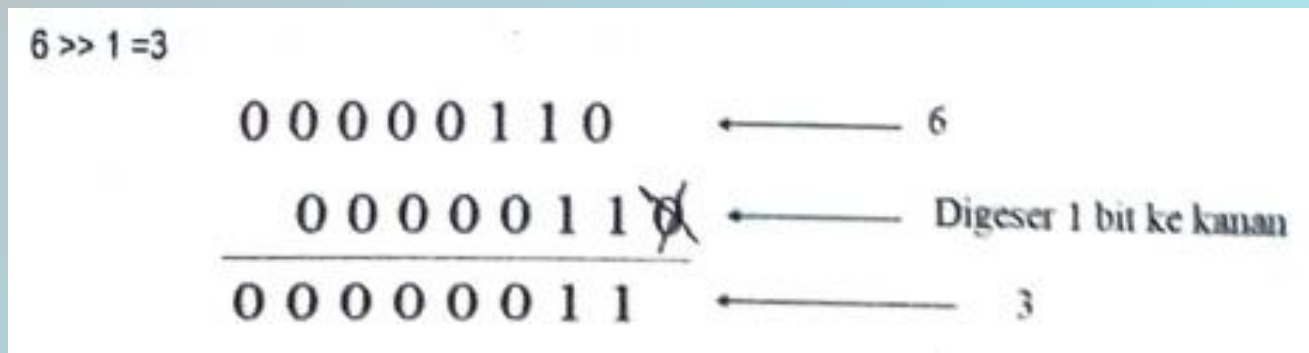
- Operator ini digunakan untuk menggeser bit ke kiri sejauh jumlah bit yang diinginkan.
- Ilustrasi pergeseran bit ke kiri:



Operator Bitwise – Geser Bit ke Kanan

Operand >> jumlah pergeseran bit

- Operator ini digunakan untuk menggeser bit ke kanan sejauh jumlah bit yang diinginkan.
- Ilustrasi pergeseran bit ke kiri:



Operator Perbandingan

- Operator yang digunakan untuk membandingkan dua buah operand atau lebih.
- Operator perbandingan akan menampilkan hasil berupa benar (true) atau salah (false).
- Operator perbandingan:

Operator	Keterangan	Contoh
>	Lebih besar dari	4 > 5 (False)
<	Lebih kecil dari	6 < 10 (True)
==	Sama dengan	20 == 20 (True)
!=	Tidak sama dengan	3 != 2 (True)
<=	Lebih kecil sama dengan	2.0 <= 2.0002 (True)
>=	Lebih besar sama dengan	6.3 >= 6.30001 (False)

Type Data Numerik

- Python mendukung 3 tipe data numerik dasar untuk keperluan operasi matematis.
- Berikut tipe data numerik:
 - ✓ Bilangan bulat (Integer): int, long
 - ✓ Bilangan pecahan (Float)
 - ✓ Bilangan kompleks (Complex Number)

```
1.foo = 100
2.bar = 30.23
3.baz = 4j
4.
5.print(type(foo))
6.print(type(bar))
7.print(type(baz))
```

```
Hasil kode program
python:
1.<class 'int'>
2.<class 'float'>
3.<class 'complex'>
```

Tugas

- Buat perhitungan matematika mengenai bangun ruang 2D atau 3D
- Upload ke github
- Deadline: 10 Februari 2022, 12.20

Terima Kasih