Pertemuan 6 : Tipe Data, Variable dan Operator

Bahasa Pemrograman

Agung Nugroho, M.Kom
Teknik Informatika – S1
Fakultas Teknik
Universitas Pelita Bangsa

Agung Nugroho, S.Kom, M.Kom

1994 | SDN Pulau Panggung, OKU Sumsel

1997 | MTs Lab Fak Tarbiah IAIN SUKA, Yogyakarta

2000 | SMK PIRI 1, Yogyakarta

2004 | Ilmu Komputer, Universitas Ahmad Dahlan, Yogyakarta

2016 | Magister Komputer, STMIK Eresha, Jakarta

2012 - Present | Freelance Web Developer

2011 - 2012 | Web Developer at BP Indonesia

2010 - 2011 | OSS Core Engineer at PT Ericsson Indonesia

2008 - 2009 | Radio Database Planner at PT. NextWave subcon NSN

2005 - 2008 | Software Developer at PT Gamatechno Indonesia

2004 - 2005 | Web Programmer at PT Reftindo Sarana

- ➔ www.linkedin.com/in/kangmasagung
- ➔ www.fb.me/agung.n
- ➔ www.koding.web.id

Tipe Data, Variable dan Operator

Pertemuan 6

Apa itu Tipe Data?

Tipe Data

- Tipe data adalah suatu media atau memori pada komputer yang digunakan untuk menampung informasi.
- Python sendiri mempunyai tipe data yang cukup unik bila kita bandingkan dengan bahasa pemrograman yang lain.

Tipe Data

Tipe Data	Contoh	Penjelasan
Boolean	True atau False	Menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0
String	"Ayo belajar Python"	Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
Integer	25 atau 1209	Menyatakan bilangan bulat
Float	3.14 atau 0.99	Menyatakan bilangan yang mempunyai koma
Hexadecimal	9a atau 1d3	Menyatakan bilangan dalam format heksa (bilangan berbasis 16)
Complex	1 + 5j	Menyatakan pasangan angka real dan imajiner
List	['xyz', 786, 2.23]	Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	('xyz', 768, 2.23)	Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
Dictionary	{'nama': 'adi','id':2}	Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

Konversi Tipe Data

Suatu tipe data dapat dikonversi lagi ke tipe data lain dengan menggunakan beberapa function berikut:

- chr(), mengubah angka ke karakter
- unichr(), mengubah angka ke karakter unicode
- str(), mengubah angka ke string
- complex(), mengubah angka ke bilangan kompleks
- float(), mengubah angka ke bilangan koma
- hex(), mengubah angka ke bentuk heksadesimal
- oct(), mengubah angka ke bentuk octal
- int(), mengubah tipe data lain ke angka

Variable?

- Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang terseimpan dalam variabel.
- Variabel adalah lokasi di memori yang digunakan untuk menyimpan nilai.
- Variabel bersifat ***mutable***, artinya nilainya bisa berubah-ubah.

Membuat Variabel di Python

- Variabel di python dapat dibuat dengan format sebagai berikut:

```
nama_variable = <nilai>
```

- Untuk melihat isi variabel, dapat menggunakan fungsi **print**

```
print nama_variable
```

Aturan Penulisan Variabel

- Nama variabel boleh diawali menggunakan huruf atau garis bawah (_), contoh: nama, _nama, namaKu, nama_variabel.
- Karakter selanjutnya dapat berupa huruf, garis bawah (_) atau angka, contoh: __nama, n2, nilai1.
- Karakter pada nama variabel bersifat sensitif (*case-sensitif*). Artinya huruf besar dan kecil dibedakan. Misalnya, variabel_Ku dan variabel_ku, keduanya adalah variabel yang berbeda.
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti if, while, for, dsb.

Operator pada Python

Apa itu Operator?

- Operator adalah simbol khusus pada Python yang melakukan perhitungan aritmatika atau logika.
- Nilai yang dioperasikan operator disebut operand.

Contoh:

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

Angka 2 dan 3 disebut operand, sedangkan tanda + adalah operator.

Tipe Operator

Python language supports the following types of operators.

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Arithmetic Operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

Operator	Meaning	Example
+	Add two operands or unary plus	x + y +2
-	Subtract right operand from the left or unary minus	x - y -2
*	Multiply two operands	x * y
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Comparison Operators

Comparison operators are used to compare values. It either returns True or False according to the condition.

Operator	Meaning	Example
>	Greater that - True if left operand is greater than the right	x > y
<	Less that - True if left operand is less than the right	x < y
==	Equal to - True if both operands are equal	x == y
!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to - True if left operand is greater than or equal to the right	x >= y
<=	Less than or equal to - True if left operand is less than or equal to the right	x <= y

Assignment Operators

- Assignment operators are used in Python to assign values to variables.
- a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

Operator	Example	Equivatent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

Logical Operators

Logical operators are the and, or, not operators.

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

Bitwise Operators

- Bitwise operators act on operands as if they were string of binary digits. It operates bit by bit, hence the name.
- Assume if a = 60; and b = 13; Now in binary format they will be as follows:
  - a = 0011 1100
  - b = 0000 1101

  - a&b = 0000 1100
  - a|b = 0011 1101
  - a^b = 0011 0001
  - ~a = 1100 0011

Operator	Meaning	Example
&	Bitwise AND	x& y = 0 (00000000)
|	Bitwise OR	x | y = 14 (00001110)
~	Bitwise NOT	~x = -11 (11110101)
^	Bitwise XOR	x ^ y = 14 (00001110)
>>	Bitwise right shift	x>> 2 = 2 (00000010)
<<	Bitwise left shift	x<< 2 = 40 (00101000)
note: x = 10 (0000 1010 in binary) and y = 4 (0000 0100 in binary)		

Membership Operators

- in and not in are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

Operator	Meaning	Example
in	True if value/variable is found in the sequence	5 in x
not in	True if value/variable is not found in the sequence	5 not in x

Identity Operators

Identity operators compare the memory locations of two objects. There are two Identity operators explained below

Operator	Meaning	Example
is	True if the operands are identical (refer to the same object)	x is True
is not	True if the operands are not identical (do not refer to the same object)	x is not True

Prioritas Eksekusi Operator

- Dari sekian banyaknya operator yang telah disebutkan, masing - masing mempunyai prioritas pemroresan yang dapat dilihat pada tabel berikut.
- Prioritas tersebut makin ke bawah makin akhir untuk dieksekusi.
- Paling atas adalah yang akan didahulukan daripada operator lain, sedangkan paling bawah adalah operator yang paling terakhir dieksekusi

Operator	Keterangan
**	Aritmatika
~,+,-	Bitwise
*,/,%,//	Aritmatika
+,-	Aritmatika
>>,<<	Bitwise
&	Bitwise
^,|	Bitwise
<=,<,>,>=	Perbandingan
<>,==,!=	Perbandingan
=,%=,/=,//=,-=,+=,*=,**=	Penugasan
is,isnot	Identitas
in,notin	Membership
not,or,and	Logika

I/O (*Input dan Output*)

Output

- Python Output Using print() function
- Example:

```
print("Hello World")
a = 5
print("Variable a bernilai", a)
```

- Secara default, kita dapat menambahkan spasi diantara variable. akan tetapi kita juga bisa mengubahnya.

```
print(*objects, sep=' ', end='\n')
```

- *objek -> variable, sep -> separatornya, end -> akhir dari perintah

Output Formatting

- Untuk merubah format output, menggunakan fungasi str.format()

```
>>> x = 5; y = 10
>>> print('The value of x is {} and y is {}'.format(x,y))
The value of x is 5 and y is 10
```

- Kurung kurawal {} digunakan sebagai placeholder.
- Kita bisa menentukan urutan yang dicetak dengan menggunakan angka (tuple index).

Input

- Untuk mengambil input menggunakan fungsi input()
- input() -> untuk mengambil nilai sesuai variablenya.

- **Contoh:**

```
>>> nama = input("Masukkan nama:")
Masukkan nama: ari
>>> print(nama)
ari
```

Lab 1

```
1 # penggunaan end
2 print('A', end='')
3 print('B', end='')
4 print('C', end='')
5 print()
6 print('X')
7 print('Y')
8 print('Z')
9
10 # penggunaan separator
11 w, x, y, z = 10, 15, 20, 25
12 print(w, x, y, z)
13 print(w, x, y, z, sep=',')
14 print(w, x, y, z, sep='')
15 print(w, x, y, z, sep=':')
16 print(w, x, y, z, sep='-----')
17
```

```
# string format
print(0, 10**0)
print(1, 10**1)
print(2, 10**2)
print(3, 10**3)
print(4, 10**4)
print(5, 10**5)
print(6, 10**6)
print(7, 10**7)
print(8, 10**8)
print(9, 10**9)
print(10, 10**10)

# string format
print('{0:>3} {1:>16}'.format(0, 10**0))
print('{0:>3} {1:>16}'.format(1, 10**1))
print('{0:>3} {1:>16}'.format(2, 10**2))
print('{0:>3} {1:>16}'.format(3, 10**3))
print('{0:>3} {1:>16}'.format(4, 10**4))
print('{0:>3} {1:>16}'.format(5, 10**5))
print('{0:>3} {1:>16}'.format(6, 10**6))
print('{0:>3} {1:>16}'.format(7, 10**7))
print('{0:>3} {1:>16}'.format(8, 10**8))
print('{0:>3} {1:>16}'.format(9, 10**9))
print('{0:>3} {1:>16}'.format(10, 10**10))
```

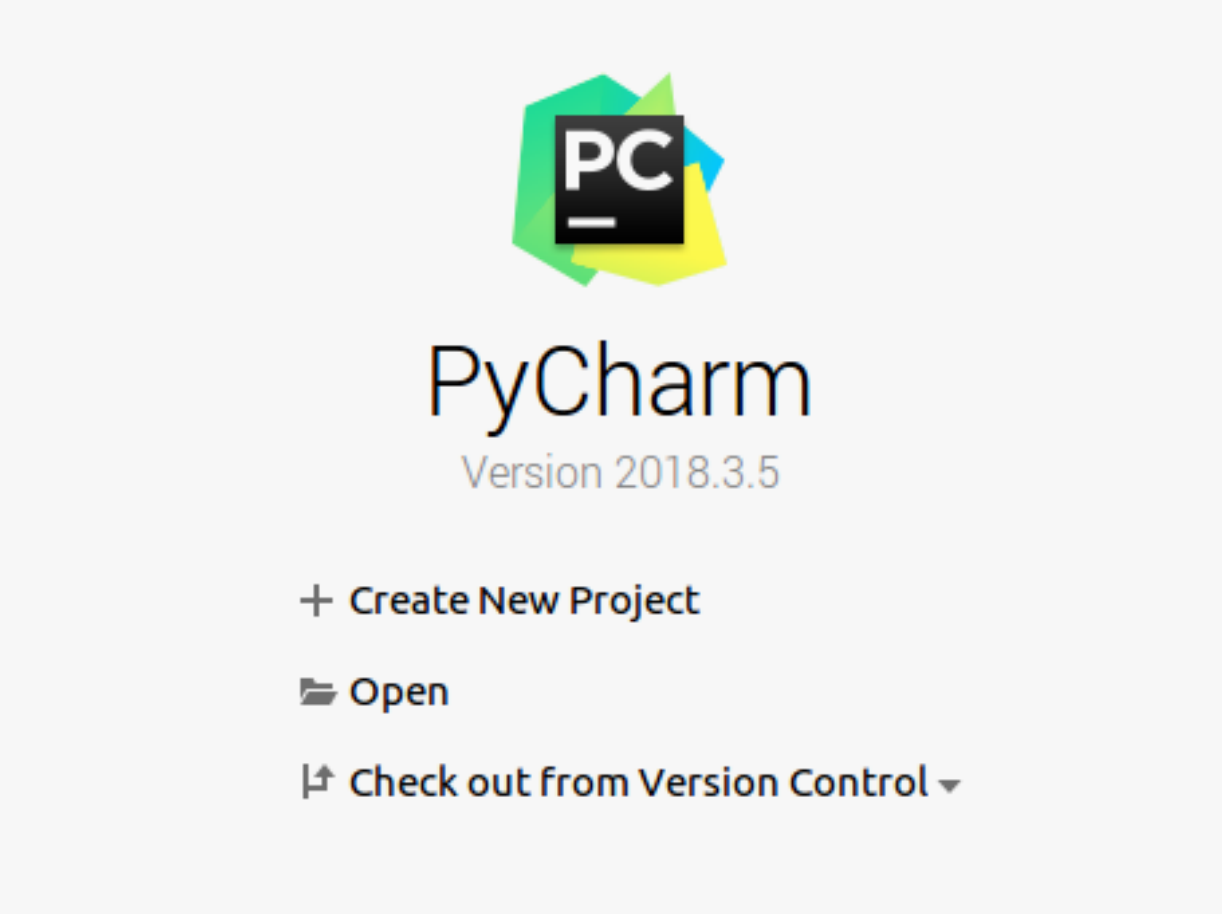
Lab 2

```
1 a=input("masukkan nilai a:")
2 b=input("masukkan nilai b:")
3 print("variable a=",a)
4 print("variable b=",b)
5 print("hasil penggabungan {1}&{0}=%d".format(a,b) %(a+b))
6
7 #konversi nilai variable
8 a=int(a)
9 b=int(b)
10 print("hasil penjumlahan {1}+{0}=%d".format(a,b) %(a+b))
11 print("hasil pembagian {1}/{0}=%d".format(a,b) %(a/b))
12
```


Any Question?

Tugas Latihan

- Buat program untuk menghitung luas dan keliling lingkaran menggunakan Python
- Simpan project Praktikum hari ini ke repository server.
- Buat penjelasan setiap Lab/latihannya pada file README.md
- Sertakan flowchart dan penjelasan program dan screenshot hasil eksekusi program



Terimakasih

Agung Nugroho

agung@pelitabangsa.ac.id

www.koding.web.id