ALGORITMA PEMROGRAMAN

02. Variabel, Tipe Data, & Operator

Fakultas Teknik
Universitas Trunojoyo Madura

04. Tipe Data, Variabel, Konstanta, Operator

1. Variabel
2. Tipe Data
3. Konstanta
4. Operator
5. Konversi

1. Variabel

a. Definisi Variabel
b. Deklarasi Variabel
c. Aturan Penamaan Variabel
d. Keyword

a. Definisi Variabel

- Nama lainnya Identifier.
- Tempat yang digunakan untuk menyimpan data.
- Di matematika sering digambarkan dengan x, y.
- Variabel bersifat ***mutable***, artinya nilainya bisa berubah-ubah.
- Dapat berisi data/objek, baik itu bilangan bulat (integer), pecahan (float), karakter (string), dll
- Deklarasi/pembuatan secara otomatis saat memberi/menugaskan (=) suatu nilai ke variabel.

b. Deklarasi Variabel

- Formatnya :

```
nama_variabel = <nilai>
```

- Contohnya :

```
print("================================================================================")
namaDepan = "Doni"
namaTengah = "Abdul"
namaBelakang = "Fatah"
nama = namaDepan + " " + namaTengah + " " + namaBelakang
print("Nama lengkap : ", nama)
print("================================================================")
```

```
print("#Menghitung Persegi Panjang")
panjang = 10 # tipe data integer
lebar = 20.5 # tipe data float
luas = panjang*lebar # tipe string
print("panjang = ",panjang)
print("lebar = ",lebar)
print("luas Persegi Panjang = ",luas)
print("================================================================================")
```

```
==================================
#Menghitung Persegi Panjang
panjang =  10
lebar =  20.5
luas Persegi Panjang =  205.0
==================================
==================================
Nama lengkap :  Doni Abdul Fatah
==================================
```

b. Aturan Penamaan Variabel

1. Penulisan nama variabel **karakter pertama** harus diawali dengan **huruf** atau **garis bawah / undescore** (_)
   Contoh : `variabel, _variabel`
2. Penulisan nama variabel karakter selanjutnya dapat berupa huruf, angka atau garis bawah/undescore (_)
   Contoh : `namaVariabel, nama_variabel, __Var, variabel_3, variabel4`
3. Penulisan nama variabel bersifat **sensitif** (*case-sensitif*). Artinya penggunaan huruf besar (kapital) dengan huruf kecil dianggap berbeda atau dibedakan penggunaanya.
   Contoh : `nuricahyono, nuriCahyono, nuri_cahyono, nuri_Cahyono.`
   ke empat contoh tersebut di anggap sebagai **variabel yang berbeda**.
4. Penulisan nama variabel tidak boleh menggunakan kata kunci atau perintah yang ada di dalam pemrograman python.
   Contoh : `if, while, for, else`

c. Keyword pada Bahasa Python

True	def	if	raise
False	del	import	return
None	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

Variabel

```
print ("#proses memasukan data ke dalam variabel")
#proses memasukan data ke dalam variabel
nama = "Matkul Alpro"
#proses mencetak variabel
print(nama)
print ("============================================================")
print ("#nilai dan tipe data dalam variabel  dapat diubah")
#nilai dan tipe data dalam variabel  dapat diubah
semester = 1                #nilai awal
print("Semester ",semester) #mencetak nilai semester
type(semester)              #mengecek tipe data semester
semester = "Ganjil"         #nilai setelah diubah
print("Semester ",semester) #mencetak nilai semester
type(semester)              #mengecek tipe data semester
print ("============================================================")
namaDepan = "Bahasa"
namaBelakang = "Python"
nama = namaDepan + " " + namaBelakang
semester = "Satu"
kelas = "A,B,C,D,E"
print("Biodata Matkul : \n", "nama ", nama, "\n", "semester",semester, "\n","kelas", kelas)
print ("============================================================")
print ("#contoh variabel lainya")
#contoh variabel lainya
inivariabel = "Halo"
ini_juga_variabel = "Hai"
_inivariabeljuga = "Hi"
inivariabel222 = "Bye"
print ("============================================================")
```

2. Tipe Data

a. Definisi Tipe Data
b. Tipe-Tipe Data
c. Value
d. Fungsi pengubah tipe data
e. Kode Escape

a. Definisi Tipe Data

- Jenis data yang digunakan untuk mendefinisikan isian dari variabel.
- Jenis data yang tersimpan dalam variabel.
- Cara untuk **memberitahu komputer** untuk **mengelompokkan data** berdasarkan apa yang **dipahami oleh komputer**.
- **Python** secara **otomatis mengenali** tipe data yang tersimpan dalam sebuah variabel.

b. Tipe-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

Tipe Data Primitif

Secara umum dalam python dibagi menjadi tiga jenis:

Tipe Data Angka :

1.int (Integer): bilangan bulat, contoh 32, 22, 12, 10, dsb.

2.float: bilangan pecahan, contoh 1.3, 4.2, 22.3, dsb.

Tipe Data Teks :

1. Char: Karakter, contoh 'R'.
2. String: Kumpulan karakter, contoh "aku lagi makan".

- Penulisan tipe data teks harus diapit dengan tanda petik. Bisa menggunakan petik tunggal ('...'), ganda ("..."), dan tiga ('''...''' atau """..."""").

Tipe data boolean

- Hanya memiliki dua nilai yaitu True dan False atau 0 dan 1.
- Penulisan True dan False, huruf pertamnya harus kapital dan tanpa tanda petik.

b. Tipe-Tipe Data

typedata.py > ...

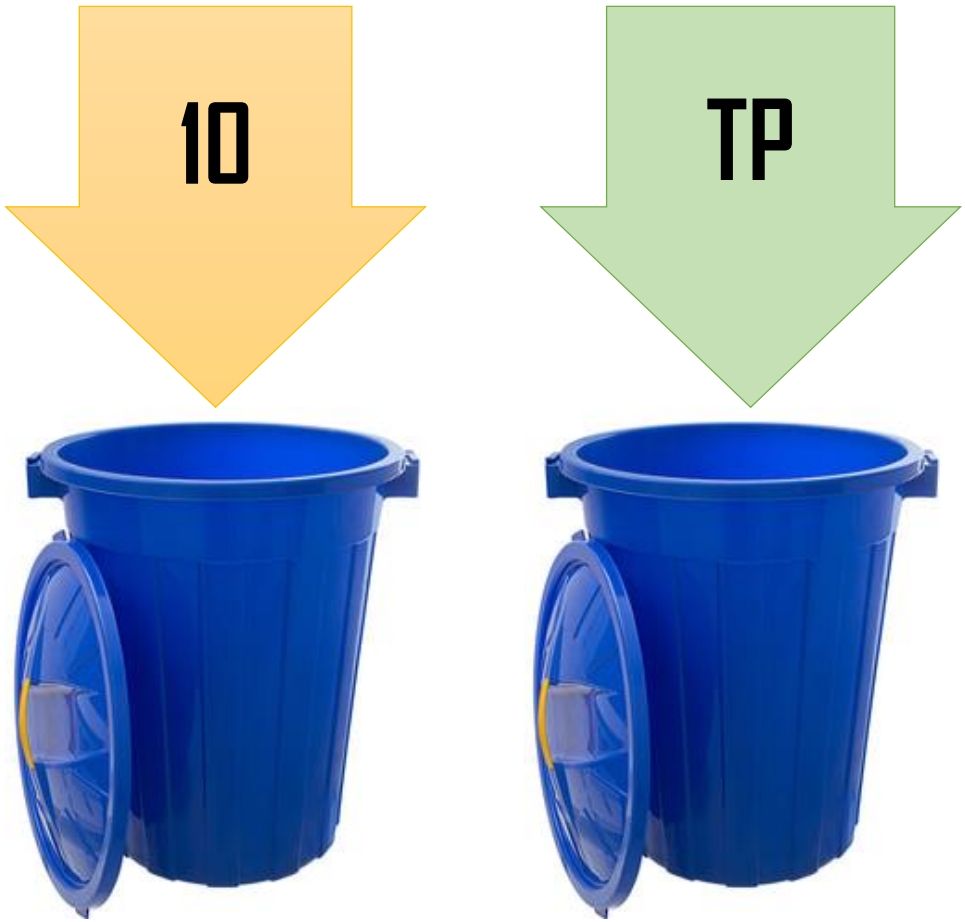
```python
print("================================================")
print("============== type data Python ==============","\n")
#tipe data Boolean
a=True
print(a, "tipenya adalah ", type(a),"\n")
#tipe data String
b= "Ayo belajar Python"
print(b, "tipenya adalah ", type(b))
c='nama saya Python'
print(c, "tipenya adalah ", type(c),"\n")
x = 5  #tipe data Integer
print(x, "tipenya adalah ", type(x),"\n")
x = 3.14 #tipe data Float
print(x, "tipenya adalah ", type(x),"\n")
x = 1+2j  #tipe data Complex
print(x, "tipenya adalah ",type(x),"\n")
#tipe data list
a = [5,10,15,20,25,30,35,40]
print("a[1] = ", a[1])
# a[0:3] = [5, 10, 15]
print("a[0:3] = ", a[0:3])
# a[5:] = [30, 35, 40]
print("a[5:] = ", a[5:])
print(a, "tipenya adalah ",type(a),"\n")
```

b. Tipe-Tipe Data

```
26  #tipe data tuple
27  white = (255,255, 255)
28  print(white, "tipenya adalah ",type(white))
29  red = (255,0,0)
30  print(white)
31  print(red[0])
32  print(red[1])
33  print(red, "tipenya adalah ",type(red),"\n")
34
35  #tipe data set
36  # set integer
37  my_set = {1,2,3}
38  print(my_set)
39  print(my_set, "tipenya adalah ",type(my_set),"\n")
40  # set dengan menggunakan fungsi set()
41  my_set = set([1,2,3])
42  print(my_set)
43  # set data campuran
44  my_set = {1, 2.0, "Python", (3,4,5)}
45  print(my_set)
46  # bila kita mengisi duplikasi, set akan menghilangkan salah satu
47  # output: {1,2,3}
48  my_set = {1,2,2,3,3,3}
49  print(my_set)
50
51  #tipe data Dictionary
52  d = {1:'satu', 2:'dua', 'tiga':3}
53  print("d[1] = ", d[1])
54  print("d['tiga'] = ", d['tiga'])
55  print(d, "tipenya adalah ",type(d),"\n")
56  print("\n")
57  print("==========================================","\n")
```

c. Value

- Adalah isi atau nilai dalam variable.
- Tipe data sering disebut **objek**.
- Tipe data string, tuple, dan list masuk ke dalam tipe data yang disebut tipe data **berurut / ordered** atau **sekuens ial / sequence**.
- Tipe data dictionary disebut data **tidak berurut / unordered**.



d. Fungsi Pengubah Format Tipe Data

NO	KODE	KEGUNAAN
1	*int()*	Mengubah menjadi integer
2	*long()*	Mengubah menjadi integer panjang
3	*float()*	Mengubah menjadi float
4	*bool()*	Mengubah menjadi boolean
5	*chr()*	Mengubah menjadi karakter
6	*str()*	Mengubah menjadi string
7	*bin()*	Mengubah menjadi bilangan biner.
8	*hex()*	Mengubah menjadi bilangan heksadesimal
9	*oct()*	Mengubah menjadi bilangan okta.

d. Fungsi Pengubah Format Tipe Data

```
1    a = 10
2    b = 3
3    c = a / b
4    print(c, "tipenya adalah ",type(c),"\n")
5    print ("====================")
6
7    a = 10
8    b = 3.5
9    c = int(a) / int(b)
10   print(c, "tipenya adalah ",type(c),"\n")
```

```
3.3333333333333335 tipenya adalah  <class 'float'>

====================
3.3333333333333335 tipenya adalah  <class 'float'>
```

Kode *Escape*

- Karakter yang dinotasikan dengan awalan berupa backslash “\”.
- Biasanya digunakan dalam penulisan string

```
print('python sedang \'trending\' saat ini')
```

Kode *Escape*

Kode Escape	Keterangan
\\	Backslash
\'	Petik Tunggal
\"	Petik Ganda
\b	Backspace
\e	Escape
\n	New Line
\0	Null
\t	Tab Horizontal
\r	Carriage Return
\f	Formfeed
\v	Tab vertikal
\xnn	Notasi Hexadesimal

3. Konstanta

a. Definisi konstanta
b. Contoh Konstanta

a. Definisi Konstanta

- bahasa Python tidak mengenal adanya **konstanta**
- Sebuah variabel dengan nilai/value tetap.
- Konstanta sangat berguna sebagai wadah yang dapat menyimpan informasi yang tidak dapat diubah.
- Ditulis dengan huruf kapital dan underscore untuk memisahkan kata.
- Misal:
  - PI=3,14
  - MATKUL_KU =“Alpro”
  - FAKULTAS =“Teknik”

b. Contoh konstanta

main.py ● konstanta.py ×

konstanta.py > ...

```
1   PI = 3.14
2   JARI_JARI=5
3   JURUSAN = "Teknik Informatika"
4   MATA_KULIAH = "Algoritma Pemrograman"
```

main.py ● konstanta.py

main.py > ...

```
1   import konstanta
2   print(konstanta.PI)
3   print(konstanta.JARI_JARI)
4   print(konstanta.JURUSAN)
5   print(konstanta.MATA_KULIAH,"\n")
```

```
3.14
5
Teknik Informatika
Algoritma Pemrograman
```

main.py ● konstanta.py

main.py > ...

```
6
7   print("Mencoba mengganti nilai konstanta")
8   konstanta.PI=input("Masukan Phi :")
9   konstanta.JARI_JARI=input("Masukan jari-jari : ")
10  luas = konstanta.PI*konstanta.JARI_JARI
11  print(luas)
```

```
Mencoba mengganti nilai konstanta
Masukan Phi :10
Masukan jari-jari : 10
Traceback (most recent call last):
  File "e:/Kuliah UTM/#Semester Ganjil 2021/Alpro/3.
 Tugas/00.Template/main.py", line 10, in <module>
    luas = konstanta.PI*konstanta.JARI_JARI
TypeError: can't multiply sequence by non-int of typ
e 'str'
```

4. Operator

a. Operator Aritmatika (Arithmetic Operators)
b. Operator Perbandingan (Comparison (Relational) Operators)
c. Operator Penugasan (Assignment Operators)
d. Operator Logika (Logical Operators)
e. Operator Bitwise (Bitwise Operators)
f. Operator Keanggotaan (Membership Operators)
g. Operator Identitas (Identity Operators)

Operator

- Simbol-simbol yang digunakan untuk melakukan operasi tertentu.

a. Operator Aritmetic / Aritmatika

Operator	Penjelasan	Contoh
+	Penambahan	20 + 6, hasil: 26
–	Pengurangan	20 – 6, hasil: 14
*	Perkalian	20 * 6, hasil: 120
/	Pembagian (real/pecahan)	20 / 6, hasil: 3.3333
//	Pembagian (dibulatkan ke bawah)	20 // 6, hasil: 3
%	Modulus (sisa hasil bagi)	20 % 6, hasil: 2
**	Pemangkatan	20 ** 6, hasil: 64000000

a. Operator Aritmetic / Aritmatika (script)

operator.py > ...

```python
a = 21
b = 10
c = a + b
print ("Hasil Penjumlahan adalah", c)
c = a - b
print ("Hasil Pengurangan adalah", c )
c = a * b
print ("Hasil Perkalian adalah", c)
c = a / b
print ("Hasil Pembagian adalah", c )
c = a % b
print ("Sisa Pembagian adalah", c)
c = a**b
print ("Hasil Pemangkatan adalah", c)
c = a//b
print ("Hasil Pembulatan Pembagian adalah", c)
```

```
Hasil Penjumlahan adalah 31
Hasil Pengurangan adalah 11
Hasil Perkalian adalah 210
Hasil Pembagian adalah 2.1
Sisa Pembagian adalah 1
Hasil Pemangkatan adalah 16679880978201
Hasil Pembulatan Pembagian adalah 2
```

b. Operator Comparison / Perbandingan

bersifat pertanyaan dan hasilnya merupakan Boolean(benar/salah).

Operator	Penjelasan	Contoh	Hasil
==	Sama dengan	5 == 5	True
!=	Tidak sama dengan	5 != 5	False
>	Lebih besar	5 > 6	False
<	Lebih kecil	5 < 6	True
>=	Lebih besar atau sama dengan	5 >= 3	True
<=	Lebih kecil atau sama dengan	5 <= 5	True

b. Operator Comparison / Perbandingan (script)

```
a = 5
b = 4
c = a == b
print ("Apakah a sama dengan b?:", c)
c = a != b
print ("Apakah a tidak sama dengan b?:", c)
c = a > b
print ("Apakah a lebih besar dari b?:", c)
c = a < b
print ("Apakah a lebih kecil dari b?:", c)
c = a <= b
print ("Apakah a lebih kecil sama dengan b?:", c)
c = a >= b
print ("Apakah a lebih besar sama dengan b?:", c)
```

```
Apakah a sama dengan b?: False
Apakah a tidak sama dengan b?: True
Apakah a lebih besar dari b?: True
Apakah a lebih kecil dari b?: False
Apakah a lebih kecil sama dengan b?: False
Apakah a lebih besar sama dengan b?: True
```

d. Operator Assignment / Penugasan

Operator yang digunakan untuk memberi nilai ke variabel.

Operator	Contoh	Penjelasan
+=	a += b	a = a + b
-=	a -= b	a = a – b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b
&=	a &= b	a = a & b
|=	a |= b	a = a | b
^=	a ^= b	a = a ^ b
<<=	a <<= b	a = a << b
>>=	a >>= b	a = a >> b

d. Operator Assignment / Penugasan (Script)

```
a = 5
b = 3
b = b + 1
c = a + b
d = c + c + a
e = (c + d)* a

print('Isi variabel a:',a)
print('Isi variabel b:',b)
print('Isi variabel c:',c)
print('Isi variabel d:',d)
print('Isi variabel e:',e)
```

```
Isi variabel a: 5
Isi variabel b: 4
Isi variabel c: 9
Isi variabel d: 23
Isi variabel e: 160
```

```
x = 10
x += 5
print('x += 5  :',x)

x = 10
x /= 5
print('x /= 5  :',x)

x = 10
x **= 5
print('x **= 5 :',x)

x = 10
x <<= 2
print('x <<= 2 :',x)
```

```
x += 5   : 15
x /= 5   : 2.0
x **= 5  : 100000
x <<= 2  : 40
```

c. Operator Logic / Logika

Operator yang dipakai untuk membuat kesimpulan logis dari 2 kondisi boolean: **true** atau **false**. operasi logika seperti AND, OR dan NOT.

Operator	Penjelasan	Contoh	Hasil
and	True jika kedua operand bernilai True	True and True	True
or	True jika salah satu operand bernilai True	True or False	True
not	True jika operand bernilai False	not False	True

c. Operator Logic / Logika (Script)

```python
a = True
b = False
# Logika AND
c = a and b
print ("c adalah",c)
# Logika OR
c = a or b
print ("c adalah",c)
# Logika Not
c = not a
print ("c adalah",c)
```

```
c adalah False
c adalah True
c adalah False
```

```python
print('Hasil dari True and True   :', True and True)
print('Hasil dari True and False  :', True and False)
print('Hasil dari False and True  :', False and True)
print('Hasil dari False and False :', False and False)

print('\n')

print('Hasil dari True or True   :', True or True)
print('Hasil dari True or False  :', True or False)
print('Hasil dari False or True  :', False or True)
print('Hasil dari False or False :', False or False)

print('\n')

print('Hasil dari not True  :', not True)
print('Hasil dari not False :', not False)
```

```
Hasil dari True and True   : True
Hasil dari True and False  : False
Hasil dari False and True  : False
Hasil dari False and False : False


Hasil dari True or True   : True
Hasil dari True or False  : True
Hasil dari False or True  : True
Hasil dari False or False : False


Hasil dari not True  : False
Hasil dari not False : True
```

e. Operator Bitwise (Bitwise Operators)

Operator khusus untuk melakukan operasi bilangan biner dalam bentuk bit.

Operator	Nama	Contoh	Biner	Hasil (biner)	Hasil (decimal)
&	And	10 & 12	1010 & 1100	1000	8
|	Or	10 | 12	1010 | 1100	1110	14
^	Xor	10 ^ 12	1010 ^ 1100	0110	6
~	Not	~ 10	~1010	0101	-11 (two complement)
<<	Left shift	10 << 1	1010 << 1	10100	20
>>	Right shift	10 >> 1	1010 >> 1	101	5

e. Operator Bitwise (Bitwise Operators) (Script)

```
print("\n","\n")
# deklarasi nilai a dan b
a = 10
b = 12
# operasi bitwise AND
c = a & b
print ("a & b =", c)

# operasi bitwise OR
c = a | b
print ("a | b =", c)

# operasi bitwise XOR
c = a ^ b
print ("a ^ b =", c)

# operasi bitwise negasi
c = ~a
print ("~a =", c)

# operasi bitwise left shift
c = a << b
print ("a << b =", c)

# operasi bitwise right shift
c = a >> b
print ("a >> b =", c)
```

```
a & b = 8
a | b = 14
a ^ b = 6
~a = -11
a << b = 40960
a >> b = 0
```

f. Operator Keanggotaan (Membership Operators)

Operator yang digunakan untuk mengecek apakah suatu nilai ada di dalam suatu himpunan tertentu atau tidak.

Operator	Penjelasan
in	Bernilai `True` jika nilai yang di cari ada di dalam suatu himpunan
not in	Bernilai `True` jika nilai yang di cari tidak ada di dalam suatu himpunan

f. Operator Keanggotaan (Membership Operators)(script)

```
nama = "teknik informatika"
member = "t" in nama
print ("t in nama =", member)

list_nama =["s","e","u","r","a","m","o","i","d"]
member = "z" not in list_nama
print ("z in list_nama =",member)
```

```
t in nama = True
z in list_nama = True



a is b : True
a is c : False
a is not c : True


i is j : False
i is not j : True


x is y : False
x is not y : True
```

```
a = 5
b = 5
c = 6
print('a is b :', a is b)
print('a is c :', a is c)
print('a is not c :', a is not c)
print('\n')

i = 'Matkul Alpro'
j = 'teknik informatika'
print('i is j :', i is j)
print('i is not j :', i is not j)
print('\n')

x = ['a','b','c']
y = ['a','b','c']
print('x is y :', x is y)
print('x is not y :', x is not y)
```

g. Operator Identitas (Identity Operators)

Operator yang bisa dipakai untuk memeriksa apakah nilai sebuah variabel ada di tempat yang sama (di memory) atau tidak.

Operator	Penjelasan
is	Bernilai `True` jika variabel di kedua sisi operator merujuk ke objek yang sama
is not	Bernilai `True` jika variabel di kedua sisi operator merujuk ke objek yang beda

g. Operator Identitas (Identity Operators) (script)

```
a = 20
b = 20

c = a is b
print ("a is b =",c)

c = a is not b
print ("a is not b =",c)
```

```
a is b = True
a is not b = False
```

```
a = 5
b = 5
c = 6
print('a is b :', a is b)
print('a is c :', a is c)
print('a is not c :', a is not c)
print('\n')

i = 'Matkul Alpro'
j = 'teknik informatika'
print('i is j :', i is j)
print('i is not j :', i is not j)
print('\n')

x = ['a','b','c']
y = ['a','b','c']
print('x is y :', x is y)
print('x is not y :', x is not y)
```

```
a is b : True
a is c : False
a is not c : True


i is j : False
i is not j : True


x is y : False
x is not y : True
```

5. Konversi

Prioritas Eksekusi Operator di Python

Operator	Keterangan
**	Eksponensial (Pemangkatan), Aritmatika
~, +, -	Komplemen, unary plus (+@) dan unary minus (-@), Bitwise
*, /, %, //	Perkalian, pembagian, modulus dan floor division (pembagian pembulatan ke bawah), Aritmatika
+, -	Penambahan dan pengurangan, Aritmatika
>>, <<	Operator bitwise kiri dan operator bitwise kanan, Bitwise
&	AND, Bitwise
^, |	OR, Bitwise exclusive
<=, <, >, >=	Operator perbandingan
<>, ==, !=	Operator persamaan
=, %=, /=, //=, -=, +=, *=, **=	Operator penugasan
is, is not	Operator identitas
in, not in	Operator membership (Keanggotaan)
not, or, and	Operator logika

Konversi

Konversi Tipe Data Number

- int(x) untuk meng-konversi x menjadi plain integer.
- long(x) untuk meng-konversi x menjadi long integer.
- float(x) untuk meng-konversi x menjadi floating point number.
- complex(x) untuk meng-konversi x menjadi complex number dengna real part x dan imaginary part zero.
- complex(x, y) untuk meng-konversi x dan y menjadi complex number dengan real part x dan imaginary part y. x dan numeric expressions y.

Konversi

Operator Spesial String

Operator	Contoh Penjelasan	
+	a + b	akan menghasilkan BelajarPython Concatenation - Menambahkan nilai pada kedua sisi operator
*	a*2	akan menghasilkan BelajarBelajar Pengulangan - Membuat string baru, menggabungkan beberapa salinan dari string yang sama
[]	a[1]	akan menghasilkan e Slice - Memberikan karakter dari indeks yang diberikan
[:]	a[1:4]	akan menghasilkan ela Range Slice - Memberikan karakter dari kisaran yang diberikan
in	B in a	akan menghasilkan 1 Keanggotaan - Mengembalikan nilai true jika ada karakter dalam string yang diberikan
not in	Z not in a	akan menghasilkan 1 Keanggotaan - Mengembalikan nilai true jika karakter tidak ada dalam string yang diberikan
r/R	print r'\n' prints \n dan print R'\n'prints \n Raw String -	Menekan arti aktual karakter Escape. Sintaks untuk string mentah sama persis dengan string biasa kecuali operator string mentah, huruf "r", yang mendahului tanda petik. "R" bisa berupa huruf kecil (r) atau huruf besar (R) dan harus ditempatkan tepat sebelum tanda kutip pertama.
%		Format - Melakukan format String

Konversi

Operator Format String %

Operator	Penjelasan
%c	character
%s	Konversi string melalui str () sebelum memformat
%i	Dianggap sebagai bilangan bulat desimal
%d	Dianggap sebagai bilangan bulat desimal
%u	Unsigned decimal integer
%o	Bilangan bulat oktal
%x	Bilangan bulat heksadesimal (huruf kecil)
%X	Bilangan bulat heksadesimal (huruf besar)
%e	Notasi eksponensial (dengan huruf kecil 'e')
%E	Notasi eksponensial (dengan huruf besar 'E')
%f	Bilangan real floating point
%g	Yang lebih pendek dari% f dan% e
%G	Lebih pendek dari% f dan% E