

AI Mastery Course

Module 1

Introduction to everything



Section

Python operators & control flow



Learning Objectives

Di akhir modul ini, kita akan dapat:

- Memahami Python operators
 - Mengeksplorasi input function di Python
 - Menggunakan conditional programming dan loops
 - Memahami bagaimana menggunakan list comprehension di Python
- 
- 



Agenda

01

OPERATORS

- Python operators
- Input function

02

LOGIC

- Control flow: if else

03

LOOP

- Control flow: for loop
- Control flow: while loop

04

COMPREHENSION

- List comprehension
- Dictionary comprehension

05

KESIMPULAN

- Kuis
- Ringkasan



01

OPERATORS

- Python operators
- Input function

Python operators



Operators di Python

Arithmetic Operator

Comparison Operator

Logical Operator

Identity Operator

Membership Operator

Arithmetic operators

Operator	Definisi	Contoh
+	Menambahkan 2 operands atau unary plus	$x + y + 2$
-	Mengurangkan operand sisi kiri dengan operand sisi kanan atau unary minus	$x - y - 2$
*	Mengkalikan 2 operands	$x * y$
/	Membagikan operand sisi kiri dengan operand sisi kanan (selalu menghasilkan float)	x / y
%	Modulus - hasil sisa dari pembagian operand sisi kiri dengan operand sisi kanan	$x \% y$ (hasil sisa dari x/y)
//	Floor division - Pembagian yang menghasilkan pembulatan angka	$x // y$
**	Exponent - operand sisi kiri dipangkatkan dengan operand sisi kanan	$x ** y$ (x dipangkatkanr y)

Arithmetic operators

```
1 x = 15
2 y = 4
3
4 print('x + y =',x+y) # Addition
5 print('x - y =',x-y) # Subtraction
6 print('x * y =',x*y) # Multiplication
7 print('x / y =',x/y) # Division
8 print('x // y =',x//y) # Floor Division
9 print('x ** y =',x**y) # Exponent
```

```
x + y = 19
x - y = 11
x * y = 60
x / y = 3.75
x // y = 3
x ** y = 50625
```

Comparison operators

Operator	Definisi	Contoh
>	Lebih besar dari - True jika operand sisi kiri lebih besar dari sisi kanan	$x > y$
<	Kurang dari - True jika operand sisi kiri lebih kecil dari sisi kanan	$x < y$
==	Sama dengan - True jika kedua operand sama besar	$x == y$
	Tidak sama dengan - True jika kedua operand tidak sama dengan	$x != y$
>=	Lebih besar dari atau sama dengan - True jika operand sisi kiri lebih besar dari atau sama dengan operand sisi kanan	$x >= y$
<=	Lebih kecil dari atau sama dengan - True jika operand sisi kiri lebih kecil dari atau sama dengan operand sisi kanan	$x <= y$

Comparison operators

```
1 x = 10
2 y = 12
3
4 print('x > y is',x>y)    # Greater than
5 print('x < y is',x<y)    # Less than
6 print('x == y is',x==y)  # Equal to
7 print('x != y is',x!=y)  # Not equal to
8 print('x >= y is',x>=y)  # Greater than or equal to
9 print('x <= y is',x<=y)  # Less than or equal to
```

```
x > y is False
x < y is True
x == y is False
x != y is True
x >= y is False
x <= y is True
```

Logical operators

Operator	Definisi	Contoh
and	True jika kedua operand true	x and y
or	True jika salah satu operand true	x or y
not	True jika operand false	not x

Logical operators

```
1 x = True
2 y = False
3
4 print('x and y is',x and y) # AND Logic
5
6 print('x or y is',x or y) # OR Logic
7
8 print('not x is',not x) # NOT Logic
```

```
x and y is False
x or y is True
not x is False
```

Identity operators

Operator	Definisi	Contoh
is	True jika operand bersifat identik	x is True
is not	True jika operand tidak bersifat identik	x is not True

Identity operators

```
1 x1 = 5
2 y1 = 5
3 x2 = 'Hello'
4 y2 = 'Hello'
5 x3 = [1,2,3]
6 y3 = [1,2,3]
7
8 print(x1 is not y1) # Identity is not same?
9
10 print(x2 is y2) # Identity is same?
11
12 print(x3 is y3) # Identity is same?
```

```
False
True
False
```

Membership operators

Operator	Definisi	Contoh
in	True jika value/variable ada di dalam sequence	5 in x
not in	True jika value/variable tidak ada di dalam sequence	5 not in x

Membership operators

```
1 x = 'Hello world'
2 y = {1:'a',2:'b'}
3
4 print('H' in x)           # Is member?
5
6 print('hello' not in x)   # Is not member?
7
8 print(1 in y)             # Is member?
9
10 print('a' in y)          # Is member?
```

True

True

True

False

Input function di python

- input() function digunakan untuk menerima input dari user
- Nilai yang diberikan sebagai input, maka fungsi input akan mengubahnya ke bentuk string
- Jika nilai integer diberikan, fungsi input tetap akan mengubahnya ke bentuk string

```
: 1 # Taking input from the user  
2 name = input("Enter your name ")  
3  
4 # Output  
5 print("Hello", name)
```

Enter your name

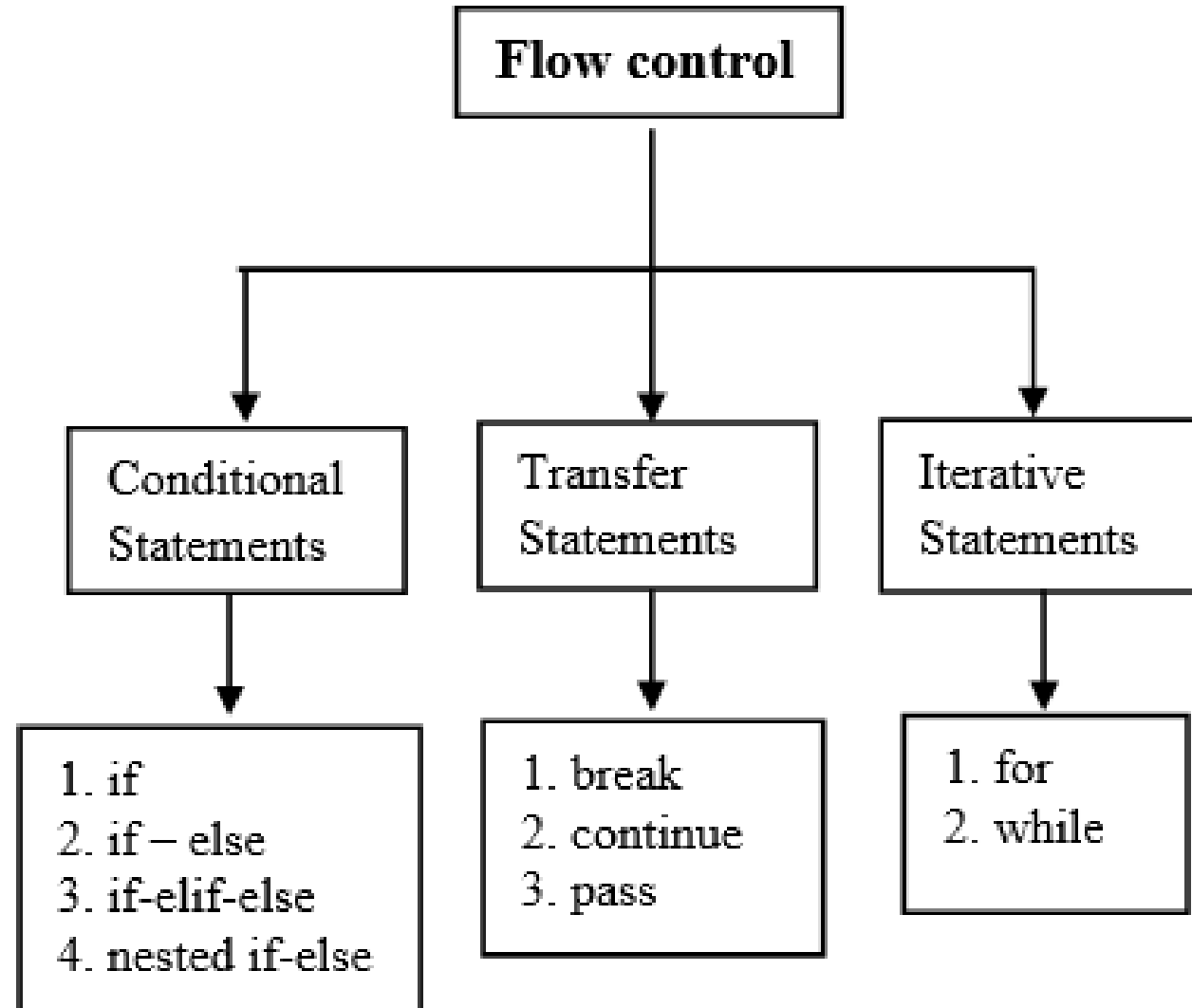


02

LOGIC

- Control flow: if else

Python: Control Flow



Control flow: if else

- Pernyataan bersyarat memutuskan aliran program berdasarkan beberapa aturan
- Ini dilakukan dengan bantuan pernyataan if-else



Control flow: if else

- Pernyataan python if digunakan untuk mengeksekusi blok kode secara kondisional
- Jika pernyataan berisi ekspresi logis atau kondisi, tergantung pada keputusan yang dibuat

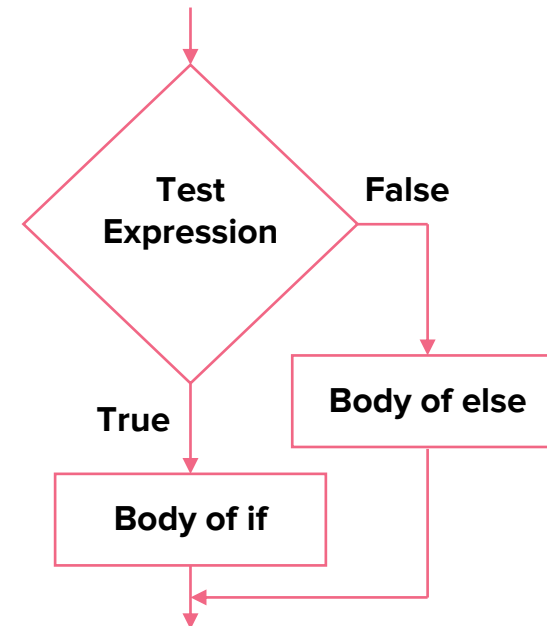


- Jika saya mendapatkan setidaknya 10 rupiah, saya akan membeli coklat
- Jika tidak maka saya akan membeli permen

Control flow: if else



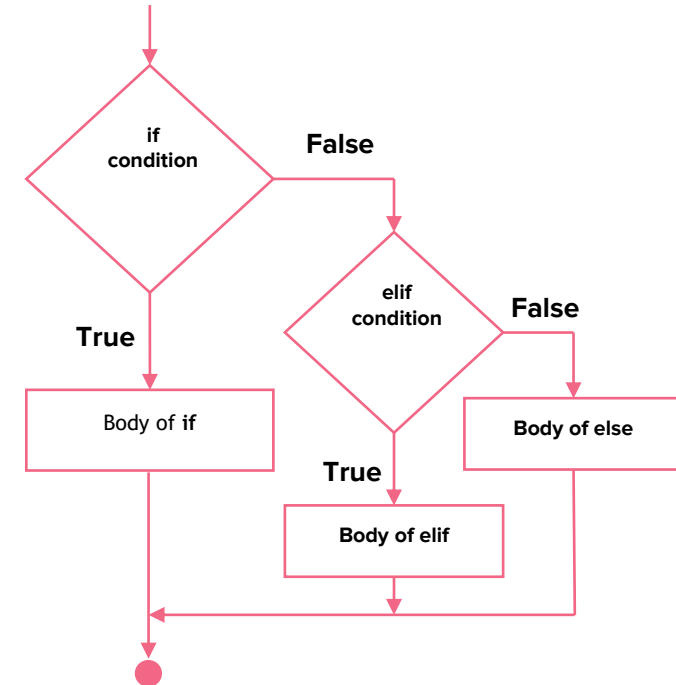
Syntax:
if (expression):
 statement
s(s)
else:
 statement
s(s)



Control flow: if else



Syntax:
if (expression):
 statement
(s)
elif (expression):
 statement
(s)
elif (expression):
 statement
(s)
else:
 statement
(s)



Control flow: if else

nested
if else

```
var = 50
if var < 200:
    print ("Expression value is less than 200")
    if var == 150:
        print ("Which is 150")
    elif var == 100:
        print ("Which is 100")
    elif var == 50:
        print ("Which is 50")
    elif var < 50:
        print ("Expression value is less than 50")
else:
    print ("Could not find true expression")

print ("Good bye!")
```

Expression value is less than 200
Which is 50
Good bye!



03

LOOP

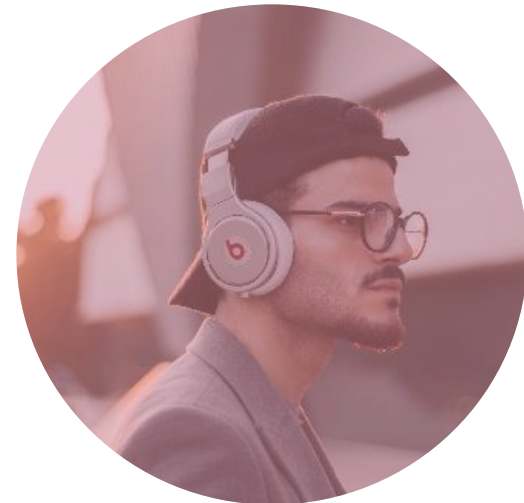
- Control flow: for loop
- Control flow: while loop

Loop

- Loop adalah blok kode yang dilakukan berulang kali dalam jumlah tertentu
- Ada dua implementasi loop di Python
- For loop – digunakan ketika jumlah pengulangan tetap
- While loop – digunakan ketika pengulangan akan dilakukan sampai suatu kondisi menjadi benar

Contoh penerapan konsep loops pada music player dan playlist

```
while (user does NOT press stop  
AND NOT end of playlist)  
{  
    get the next song  
    play the song  
}
```

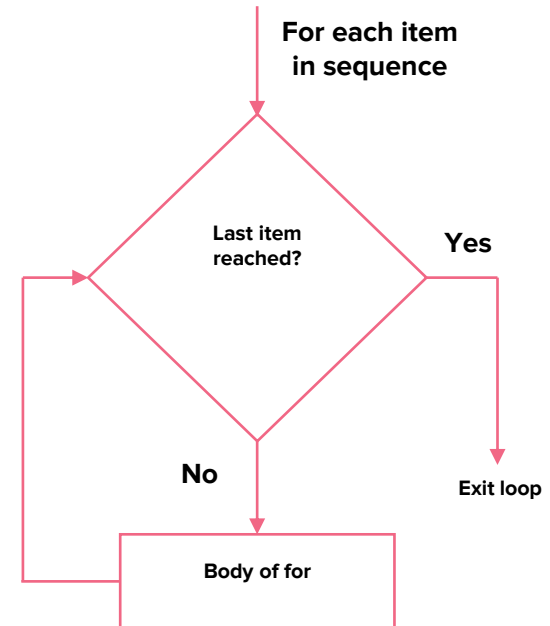


Loop: for loop

- for loop digunakan untuk mengulangi urutan yang berupa list, tuple, dictionary, atau set

Syntax:
for (initialization;
condition;
propagation)

```
{  
statements(s)  
}
```



Loop: for loop

Fungsi range() membuat urutan angka untuk diterasi

Ini dapat digunakan untuk membuat urutan penambahan/pengurangan

```
1 lst = ["wiley", 'python', "pyspark", "data", "Engineering"]
2 for i in range(len(lst)):
3     print(lst[i])
```

```
wiley
python
pyspark
data
Engineering
```

```
1 for j in range(0,10):
2     print(j)
```

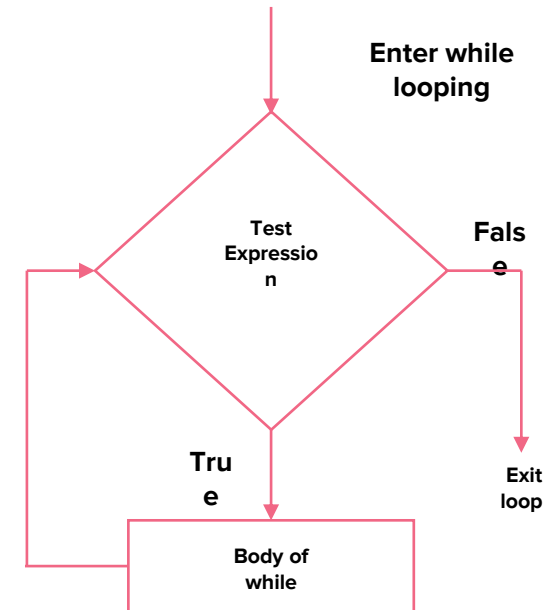
```
0
1
2
3
4
5
6
7
8
9
```

Loop: while loop

while Loop

While loop mengeksekusi blok kode di dalamnya, hingga kondisi mengembalikan True dan keluar dari loop setelah kondisi mengembalikan False

```
Syntax:  
while  
(condition)  
{  
statements(s)  
}
```



Loop: break

The break statement, seperti di bahasa C, akan menghentikan proses loop dimana break berada

```
1 for letter in 'for loop python':  
2  
3     # break the loop as soon it sees 'p'  
4     if letter == 'p':  
5         break  
6  
7     print('Current Letter :', letter)
```

```
Current Letter : f  
Current Letter : o  
Current Letter : r  
Current Letter :  
Current Letter : l  
Current Letter : o  
Current Letter : o
```

Loop: continue

Continue mengembalikan kontrol ke awal loop, digunakan bersama loop for dan while.

```
1 for letter in 'for loop python':  
2  
3     # break the loop as soon it sees 'p'  
4     if letter == 'p':  
5         continue  
6     print('Current Letter :', letter)
```

```
Current Letter : f  
Current Letter : o  
Current Letter : r  
Current Letter :  
Current Letter : l  
Current Letter : o  
Current Letter : o  
Current Letter :  
Current Letter : y  
Current Letter : t  
Current Letter : h  
Current Letter : o  
Current Letter : n
```

Loop: pass

Pass digunakan untuk menulis loop kosong. Pass juga digunakan untuk pernyataan kontrol kosong, fungsi dan kelas.

```
1 for letter in 'for loop python':  
2     pass  
3  
4 print("Hello from Python")
```

Hello from Python



04

COMPREHENSION

- List comprehension
- Dictionary comprehension

List comprehension

List comprehension menawarkan sintaks yang lebih pendek ketika membuat list baru berdasarkan nilai dari list yang sudah ada

Kita juga dapat menambahkan conditional statements pada list comprehension

```
1 # list comprehension without condition
2 # [f(x) for x in sequence]
3 mylist= [2,3,6,5,9]
4 sqlist = [x**2 for x in mylist]
5 print(sqlist)
6
7
8 # list comprehension with if condition
9 # [f(x) for x in sequence if condition]
10 sqevenlist = [x**2 for x in mylist if x%2==0]
11 print(sqevenlist)
12
13 # list comprehension with if condition and else
14 # template 3: [f(x) if condition else g(x) for x in sequence]
15 # for even numbers - get square, odd number get cube
16 customlist = [x**2 if x%2==0 else x**3 for x in mylist]
17 print(customlist)
18
```

```
[4, 9, 36, 25, 81]
```

```
[4, 36]
```

```
[4, 27, 36, 125, 729]
```

Dictionary comprehension

A dictionary comprehension berbentuk {key: value for (key, value) in iterable}

```
1 # Dictionary comprehension
2 # without condition: {key:value for vars in sequence}
3 mydic = {"apple":20,"mango":30,'banana':50,'grapes':40,'watermelon':80}
4 # add 10 with price of each fruit
5 newdic = {key:value+10 for (key,value) in mydic.items()}
6 print(newdic)
7
8 # adding conditions: {key:value for vars in sequence if condition}
9 # extracting dictionary of fruits having price more than 50
10 newdic = {key:value for (key,value) in mydic.items() if value>50}
11 print(newdic)
12
13 # adding conditions: {key:(value if condition else value2) for vars in sequence}
14 # increasing prices for fruits having price less than 50
15 newdic = {key:(value+10 if value<50 else value) for (key,value) in mydic.items()}
16 print(newdic)
```

```
{'apple': 30, 'mango': 40, 'banana': 60, 'grapes': 50, 'watermelon': 90}
{'watermelon': 80}
{'apple': 30, 'mango': 40, 'banana': 50, 'grapes': 50, 'watermelon': 80}
```



05

KESIMPULAN

- Kuis
- Ringkasan

Pertanyaan

Manakah dari keyword berikut yang digunakan untuk menginterupsi eksekusi loop?

- A. Break
- B. Continue
- C. Pass
- D. Wait



Pertanyaan

Manakah dari keyword berikut yang digunakan untuk menginterupsi eksekusi loop?

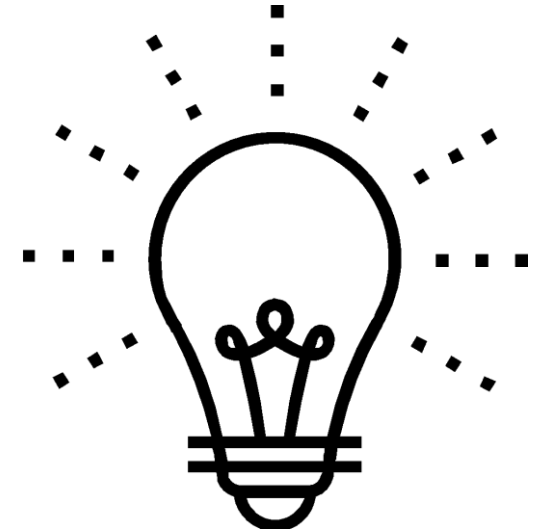
- A. Break
- B. Continue
- C. Pass
- D. Wait

Answer: A



Kesimpulan

- Operator di Python digunakan untuk melakukan manipulasi data
- Fungsi Input() dapat digunakan untuk menerima input dari pengguna
- Python control flow structure berupa – if else, for loop dan while loop
- Loop digunakan untuk mengulang eksekusi kode yang ada
- List comprehension adalah cara yang efektif untuk menggambarkan dan membangun list berdasarkan list yang sudah ada
- Secara umum, list comprehension lebih ringan dan sederhana daripada fungsi dan list



Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- 📍 Jakarta Selatan/Pusat
- 📍 Jakarta Barat/BSD
- 📍 Kota Bandung
- 📍 Kab. Bandung
- 📍 Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media



TERIMA KASIH