



Python Programming

Operators dan Conditional Statement



01

Operators

Operators

- Operator adalah symbol standar yang digunakan untuk melakukan operasi pada nilai dan variable
- Di Python terdapat beberapa operator antara lain:
 1. Arithmetic operators
 2. Comparison operators
 3. Assignment operators
 4. Logical operators
 5. Membership operators
 6. Identity operators
 7. Bitwise operators



02

Arithmetic

Operators: Arithmetic

Operator	Nama	Penjelasan	Contoh
+	Penjumlahan	Melakukan penjumlahan nilai dari bilangan	$2 + 2 = 4$
-	Pengurangan	Melakukan pengurangan nilai dari bilangan	$5 - 4 = 1$
*	Perkalian	Menghasilkan perkalian nilai dari bilangan	$3 * 5 = 15$
/	Pembagian	Menghasilkan pembagian nilai dari bilangan	$10 / 2 = 5$
//	Floor Division	Pembagian dengan membulatkan hasilnya	$15 // 2 = 7$
%	Modulus	Menghasilkan nilai sisa pembagian ketika nilai di sebelah kiri dibagi oleh nilai di sebelah kanan	$13 \% 3 = 1$
**	Eksponensial	Memangkatkan nilai di sebelah kiri dengan nilai di sebelah kanan	$2 ** 3 = 8$

Operators: Arithmetic

- Contoh aplikasi operator penjumlahan

```
1 x = 2 + 3  
2 print(x)
```

5

- Contoh aplikasi operator pengurangan

```
1 x = 10 - 8  
2 print(x)
```

2

- Contoh aplikasi operator perkalian

```
1 x = 2 * 5  
2 print(x)
```

10

Operators: Arithmetic

- Contoh aplikasi operator pembagian dan floor division

```
1 # Contoh aplikasi operator pembagian
2 print(17 / 5)
3
4 # Contoh aplikasi operator floor division
5 print(17 // 5)
```

3.4

3

- Contoh aplikasi operator modulus dan eksponensial

```
1 # Contoh aplikasi operator modulus
2 print(10 % 3)
3
4 # Contoh aplikasi operator eksponensial
5 print(2 ** 5)
```

1

32



03

Comparison

Operators: Comparison

Operator	Nama	Penjelasan	Contoh
==	Equal	Memeriksa apakah kedua nilai memiliki nilai yang sama , jika kondisi benar maka akan bernilai True	2 == 2
!=	Not equal	Memeriksa apakah kedua nilai memiliki nilai yang tidak sama / berbeda, jika kondisi benar maka akan bernilai True	2 != 3
>	Greater than	Memeriksa apakah nilai di sebelah kiri lebih besar dari nilai sebelah kanan, jika kondisi benar maka akan bernilai True	7 > 3
<	Less than	Memeriksa apakah nilai di sebelah kiri lebih kecil dari nilai sebelah kanan, jika kondisi benar maka akan bernilai True	5 < 1
>=	Greater than or equal to	Memeriksa apakah nilai di sebelah kiri lebih besar dari atau sama dengan nilai sebelah kanan , jika kondisi benar maka akan bernilai True	2 >= 2
<=	Less than or equal to	Memeriksa apakah nilai di sebelah kiri lebih kecil dari atau sama dengan nilai sebelah kanan , jika kondisi benar maka akan bernilai True	6 <= 6

Operators: Comparison

- Contoh aplikasi operator comparison “equal”

```
1 # Contoh comparison equal
2 print(2 == 2)
3 print(3 == 2)
```

True
False

- Contoh aplikasi operator comparison “not equal”

```
1 # Contoh comparison not equal
2 print(2 != 2)
3 print(3 != 2)
```

False
True

Operators: Comparison

- Contoh aplikasi operator comparison “greater than”

```
1 # Contoh comparison greater than
2 print(2 > 2)
3 print(3 > 2)
```

False

True

- Contoh aplikasi operator comparison “less than”

```
1 # Contoh comparison less than
2 print(2 < 2)
3 print(3 < 2)
```

False

False

Operators: Comparison

- Contoh aplikasi operator comparison “greater than or equal to”

```
1 # Contoh comparison greater than or equal to
2 print(2 >= 2)
3 print(3 >= 2)
```

True

True

- Contoh aplikasi operator comparison “less than or equal to”

```
1 # Contoh comparison less than
2 print(2 <= 2)
3 print(3 <= 2)
```

True

False



04

Assignment

Operators: Assignment

Operator	Contoh	Artinya	Penjelasan
=	x = 2	x = 2	Memberikan nilai sebelah kiri pada variable sebelah kanan
+=	x += 2	x = x + 2	Melakukan penjumlahan antara variable itu sendiri dengan nilai di sebelah kanan
-=	x -= 2	x = x - 2	Melakukan pengurangan antara variable itu sendiri dengan nilai di sebelah kanan
*=	x *= 2	x = x * 2	Melakukan perkalian antara variable itu sendiri dengan nilai di sebelah kanan
/=	x /= 2	x = x / 3	Melakukan pembagian antara variable itu sendiri dengan nilai di sebelah kanan
//=	x //= 2	x = x // 2	Melakukan floor division antara variable itu sendiri dengan nilai di sebelah kanan
%=	x %= 2	x = x % 2	Melakukan floor division antara variable itu sendiri dengan nilai di sebelah kanan
**=	x **= 2	x = x ** 2	Melakukan eksponensial antara variable itu sendiri dengan nilai di sebelah kanan

Operators: Assignment

- Contoh aplikasi operator assignment “=”

```
1 x = 2
2 print(x)
```

2

- Contoh aplikasi operator assignment “+=”

```
1 x = 2
2 x += 2
3 print(x)
```

4

- Contoh aplikasi operator assignment “-=”

```
1 x = 5
2 x -= 2
3 print(x)
```

3

Operators: Assignment

- Contoh aplikasi operator assignment “*=”

```
1 x = 5
2 x *= 2
3 print(x)
```

10

- Contoh aplikasi operator assignment “/=”

```
1 x = 10
2 x /= 2
3 print(x)
```

5.0

- Contoh aplikasi operator assignment “//=”

```
1 x = 13
2 x //= 2
3 print(x)
```

6

Operators: Assignment

- Contoh aplikasi operator assignment “%=”

```
1 x = 13  
2 x %= 2  
3 print(x)
```

1

- Contoh aplikasi operator assignment “**=”

```
1 x = 2  
2 x **= 3  
3 print(x)
```

8



05

**Logical,
Membership,
dan Identity**

Operators: Logical

Operator	Contoh	Penjelasan
and	$x = 2$ and $x < 5$	Menghasilkan True apabila kedua statement benar, apabila salah satu statement salah maka hasilnya adalah False
or	$x > 2$ or $x < 4$	Menghasilkan True apabila salah satu statement benar, apabila kedua statement salah maka hasilnya adalah False
not	not ($x < 0$ and $x > 5$)	Menghasilkan True apabila salah satu atau kedua statement salah. Namun bernilai False apabila kedua statement benar

Operators: Logical

- Contoh aplikasi operator logikal “and”

```
1 x = 2
2 y = 2
3
4 print(x > 0 and y > 0)
```

True

- Contoh aplikasi operator logikal “or”

```
1 x = 2
2 y = 0
3
4 print(x > 0 or y > 0)
```

True

- Contoh aplikasi operator logikal “not”

```
1 x = 2
2 y = 2
3
4 print(not(x > 0 and y < 0))
```

True

Operators: Membership

Operator	Contoh	Penjelasan
in	x in y	Menghasilkan nilai True apabila urutan dengan nilai yang ditentukan ada pada object
not in	x not in y	Menghasilkan nilai True apabila urutan dengan nilai yang ditentukan tidak ada pada object

Operators: Membership

- Contoh aplikasi operator membership “in”

```
1 x = [1, 2, 3, 4]
2
3 print(2 in x)
4 print(5 in x)
```

True
False

- Contoh aplikasi operator membership “not in”

```
1 x = [1, 2, 3, 4]
2
3 print(2 not in x)
4 print(5 not in x)
```

False
True

Operators: Identity

Operator	Contoh	Penjelasan
Is	x is y	Menghasilkan nilai True apabila kedua variable merupakan object yang sama
Is not	x is not y	Menghasilkan nilai True apabila kedua variable merupakan object yang berbeda

Operators: Identity

- Contoh aplikasi operator identity “is”

```
1 x = [1, 2, 3, 4, 5]
2 y = [1, 2, 3, 4, 5]
3 z = x
4
5 print(x is y)
6 print(x is z)
```

False

True

- Contoh aplikasi operator membership “is not”

```
1 x = [1, 2, 3, 4, 5]
2 y = [1, 2, 3, 4, 5]
3 z = x
4
5 print(x is not y)
6 print(x is not z)
```

True

False

Quiz

1. Manakah logical operator yang akan bernilai True apabila seluruh kondisinya terpenuhi?
 - a. not
 - b. or
 - c. if
 - d. and
 - e. must
2. Berapa hasil dari x jika $x = 17 // 2$?
 - a. $x = 15$
 - b. $x = 8$
 - c. $x = 1$
 - d. $x = 8.5$
 - e. $x = 19$



06

Conditional Statement

Pernyataan Berkondisi: If

- Pada Bahasa Pemrograman Python, terdapat statement percabangan IF seperti Bahasa pemrograman lainnya
- Pernyataan disimpan setelah if dan hasil ditentukan dari nilai pernyataan tersebut
- Jika hasil pernyataan bernilai True, maka perintah di dalam statement di dalam if akan dieksekusi, namun jika tidak maka program akan masuk ke block selanjutnya.
- Block if memiliki indentasi setelah tanda titik dua (:)

```
1 x = 1
2 if (x > 0):
3     print("Eksekusi perintah ini")
```

Eksekusi perintah ini

Pernyataan Berkondisi: If

- Kondisi if → mengeksekusi program jika kondisi bernilai True. Jika kondisi bernilai False maka tidak akan dieksekusi.

```
1 x = 5
2 if (x > 1):
3     print("nilai x lebih besar daripada 1")
```

nilai x lebih besar daripada 1

```
1 x = 0
2 if (x > 1):
3     print("nilai x lebih besar daripada 1")
```

```
1 x = 3
2 if (x > 0 and x < 5):
3     print("nilai x lebih besar dari 0 dan lebih kecil dari 5")
```

nilai x lebih besar dari 0 dan lebih kecil dari 5

Pernyataan Berkondisi: Else

- Else harus dikombinasikan dengan if statement sebagai kondisi yang dieksekusi apabila pernyataan bernilai False
- Else bersifat opsional, namun dalam penggunaannya harus didahului dengan if statement
- Apabila nilai tidak memenuhi kondisi if statement, maka program secara otomatis mengeksekusi perintah di else statement

```
1 x = 1
2 if (x < 0):
3     print("Perintah ini tidak akan dieksekusi")
4 else:
5     print("Eksekusi perintah ini")
```

Eksekusi perintah ini

Pernyataan Berkondisi: Else

- Kondisi if, else → melakukan pengambilan keputusan jika memenuhi suatu kondisi dan menentukan tindakan jika kondisi tidak sesuai.

```
1 nilai = 80
2 if (nilai > 75):
3     print("Selamat Anda lulus")
4 else:
5     print("Ayo belajar lebih giat lagi")
```

Selamat Anda lulus

```
1 nilai = 80
2 absen = 70
3 if (nilai > 75 and absen > 80):
4     print("Selamat Anda lulus")
5 else:
6     print("Ayo belajar lebih rajin lagi")
```

Ayo belajar lebih rajin lagi

Pernyataan Berkondisi: Elif

- Elif merupakan singkatan dari “Else if” yang digunakan sebagai alternatif untuk if bertingkat .
- Apabila terdapat beberapa kondisi bertingkat maka diperlukan elif
- Dalam penggunaannya, elif tidak dibatasi dan bersifat opsional

```
1 umur = 16
2
3 if (umur > 17):
4     print("Anda boleh membuat SIM")
5 elif (umur < 17):
6     print("Anda belum boleh membuat SIM")
```

Anda belum boleh membuat SIM

Pernyataan Berkondisi: Elif

- Kondisi elif → pengambilan keputusan dengan menyeleksi beberapa kemungkinan yang terjadi

```
1 a = 2
2 b = 5
3
4 if (a > b):
5     print("Nilai a lebih besar dari b")
6 elif (a < b):
7     print("Nilai a lebih kecil dari b")
8 else:
9     print("Nilai a sama dengan b")
```

Nilai a lebih kecil dari b

```
1 a = 2
2 b = 2
3
4 if (a > b):
5     print("Nilai a lebih besar dari b")
6 elif (a < b):
7     print("Nilai a lebih kecil dari b")
8 else:
9     print("Nilai a sama dengan b")
```

Nilai a sama dengan b

Pernyataan Berkondisi: Short Hand

- Jika hanya ada satu kondisi untuk dieksekusi, maka penulisan if dapat dilakukan dengan cara:

```
1 a = 2
2 b = 1
3
4 if a > b: print("Nilai a lebih besar dari b")
```

Nilai a lebih besar dari b

- Penulisan program secara singkat untuk satu if dan else

```
1 a = 2
2 b = 3
3
4 print("Nilai a lebih besar") if a > b else print("Nilai b lebih besar")
```

Nilai b lebih besar

Pernyataan Berkondisi: Short Hand

- Penulisan program secara singkat untuk if else dengan beberapa kondisi

```
1 a = 5
2 b = 3
3
4 print("a") if a > b else print("a == b") if a == b else print("b")
```

a

```
1 a = 3
2 b = 3
3
4 print("a") if a > b else print("a == b") if a == b else print("b")
```

a == b

```
1 a = 2
2 b = 3
3
4 print("a") if a > b else print("a == b") if a == b else print("b")
```

b

Pernyataan Berkondisi: If and Operatos

- Program if juga bisa dikombinasikan dengan beberapa operator, contohnya menggunakan logical operator and, or, dan not

```
1 nilai = 80
2 absen = 70
3
4 if (nilai > 75 and absen > 80):
5     print("Selamat Anda lulus karena nilai dan absen anda memenuhi")
6 elif (nilai > 75 and absen < 80):
7     print("Anda belum lulus ujian karena absen Anda belum memenuhi")
8 elif (nilai < 75 and absen > 80):
9     print("Anda belum lulus ujian karena nilai Anda belum memenuhi")
10 else:
11     print("Anda belum lulus ujian karena nilai dan absen anda belum memenuhi")
```

Anda belum lulus ujian karena absen Anda belum memenuhi

Pernyataan Berkondisi: If and Operatos

- Contoh kondisi if dengan penggunaan logical operator “or” dan “not”

```
1 a = 2
2 b = 3
3 c = 5
4
5 if (a > b or a > c):
6     print("Ada satu kondisi yang dipenuhi")
7 else:
8     print("Tidak ada kondisi yang terpenuhi")
```

Tidak ada kondisi yang terpenuhi

```
1 a = 0
2 b = 2
3
4 if not(a > b):
5     print("Nilai a lebih kecil")
6 else:
7     print("Nilai b lebih kecil")
```

Nilai a lebih kecil

Pernyataan Berkondisi: Nested If

- Nested if atau if bersarang adalah kondisi dimana adanya statement if di dalam statement if.

```
1 a = 10
2
3 if (a > 0):
4     print("Nilai a lebih besar dari 0")
5     if (a > 5):
6         print("Nilai a juga lebih besar dari 5")
7     else:
8         print("Nilai a lebih kecil dari 5")
```

```
Nilai a lebih besar dari 0
Nilai a juga lebih besar dari 5
```

Pernyataan Berkondisi: Nested If

- Penggunaan nested if tidak terbatas dan sesuai dengan kebutuhan

```
1 a = 9
2
3 if (a > 0):
4     print("Nilai a lebih besar dari 0")
5     if (a > 5):
6         print("Nilai a lebih besar dari 5")
7         if (a > 10):
8             print("Nilai a lebih besar dari 10")
9         else:
10            print("Tapi, nilai a lebih kecil dari 10")
11    else:
12        print("Tapi, nilai a lebih kecil dari 5")
13 else:
14    print("Nilai a lebih kecil dari 0")
```

Nilai a lebih besar dari 0
Nilai a lebih besar dari 5
Tapi, nilai a lebih kecil dari 10

Pernyataan Berkondisi: Pass Statement

- Pernyataan if tidak boleh kosong, tetapi untuk kasus tertentu kadang kita memiliki pernyataan if tanpa konten. Untuk menghindari *error* pada program, maka dapat menggunakan “pass”

```
1 a = 5
2
3 if (a > 5):
4
5 print("Eksekusi perintah ini")
```

```
Cell In[34], line 5
    print("Eksekusi perintah ini")
    ^
```

IndentationError: expected an indented block after 'if' statement on line 3

```
1 a = 5
2
3 if (a > 5):
4     pass
5
6 print("Eksekusi perintah ini")
```

Eksekusi perintah ini



07

Studi Kasus

Studi Kasus

- Buatlah sebuah program untuk melakukan pengecekan nilai yang di *input* oleh *user*, apabila **nilai ≥ 90** maka bernilai A, apabila **$90 < \text{nilai} \geq 80$** bernilai B, apabila **$80 < \text{nilai} \leq 70$** maka bernilai C, dan apabila **nilai < 70** dari maka nilainya D
- Buatlah sebuah program yang melakukan pengecekan *input* dari *user* dengan kondisi:
 1. Cek angka yang di *input* *user*, apabila bernilai ganjil maka *output*-nya “Angka <input_angka> merupakan angka ganjil”
 2. Apabila bernilai ganjil maka *output*-nya “Angka <input_angka> merupakan angka genap”
 3. Apabila *user* memasukan nilai selain angka, maka *output* yang dihasilkan “Maaf silahkan masukan angka”

Terimakasih

Do you have any questions?

addyouremail@freepik.com

+91 620 421 838

yourcompany.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution