

BAB 2

Data Types, Collection & Operator

Dalam pemrograman, data adalah elemen utama yang harus dipahami komputer melalui **data types**, seperti angka bulat, pecahan, teks, atau logika. Data sering kali tersusun dalam **collection**, yaitu wadah yang menyimpan banyak data sekaligus, misalnya daftar angka atau pasangan kunci–nilai. Untuk mengolahnya digunakan **operator**, yaitu simbol khusus untuk melakukan perhitungan, perbandingan, maupun operasi logika.

A. Konsep Dasar Data Types

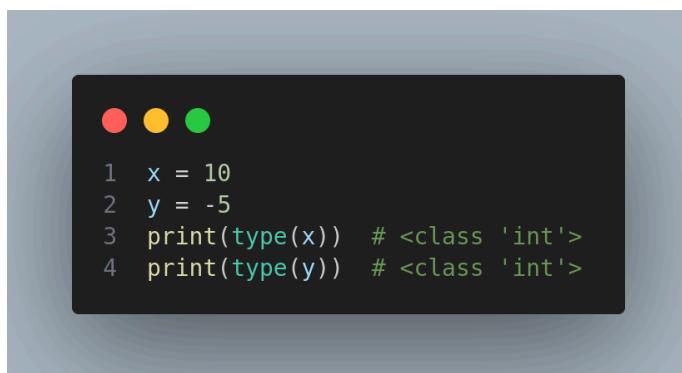
Data types (tipe data) adalah klasifikasi data dalam Python yang menentukan jenis nilai, serta operasi apa yang dapat dilakukan terhadap nilai tersebut. Dengan mengetahui tipe data, programmer dapat mengelola dan memproses data secara tepat.

1. Jenis-jenis Tipe Data Dasar di Python

a. Integer (int)

Bilangan bulat, positif atau negatif.

Contoh:

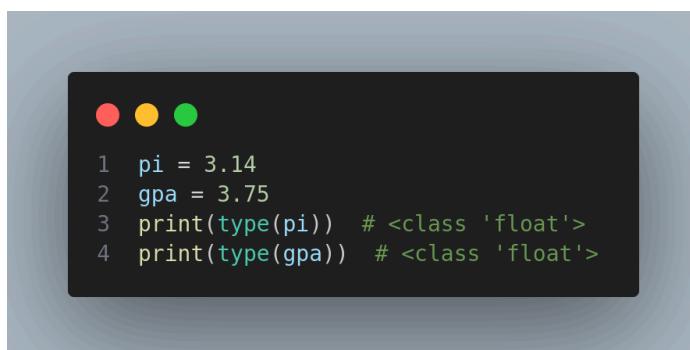


```
1 x = 10
2 y = -5
3 print(type(x)) # <class 'int'>
4 print(type(y)) # <class 'int'>
```

b. Float (float)

Bilangan desimal atau pecahan.

Contoh:



```
1 pi = 3.14
2 gpa = 3.75
3 print(type(pi)) # <class 'float'>
4 print(type(gpa)) # <class 'float'>
```

c. String (str)

Sekumpulan karakter (teks).

Contoh:

```
● ● ●  
1 name = "Kevin"  
2 print(type(name)) # <class 'str'>  
3
```

d. Boolean (bool)

Nilai logika: True atau False.

Contoh:

```
● ● ●  
1 is_student = True  
2 print(type(is_student)) # <class 'bool'>  
3
```

2. Konversi Tipe Data (Type Casting)

Python memungkinkan konversi antar tipe data. Python mendukung dua jenis konversi:

a. Konversi Implisit (Implicit Conversion)

Dilakukan otomatis oleh Python saat operasi melibatkan tipe berbeda.

Contoh:

```
● ● ●  
1 x = 5      # int  
2 y = 2.5    # float  
3 z = x + y # int otomatis jadi float  
4 print(z)   # 7.5  
5
```

b. Konversi Ekplisit (Explicit Conversion / Casting)

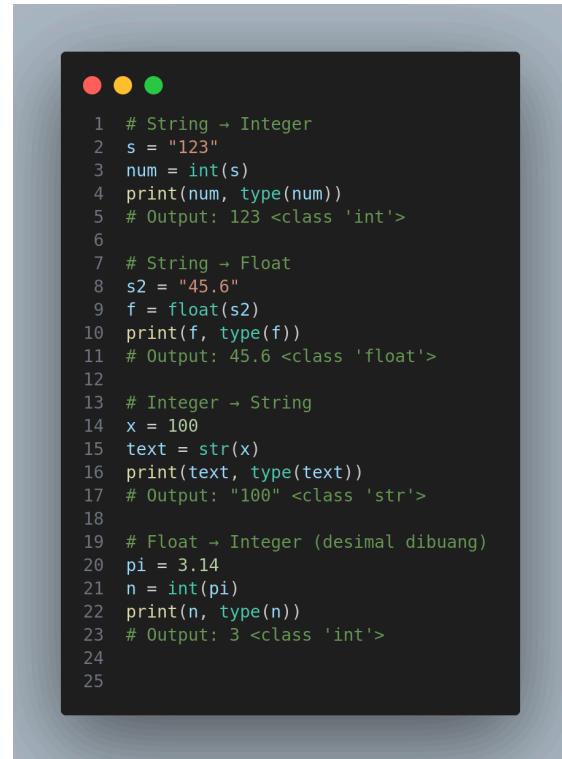
Dilakukan secara langsung menggunakan fungsi bawaan Python.

Fungsi yang umum digunakan:

- int() : ke bilangan bulat
- float() : ke bilangan pecahan
- str() : ke string

- `bool()` : ke logika True/False

Contoh:



```
1 # String -> Integer
2 s = "123"
3 num = int(s)
4 print(num, type(num))
5 # Output: 123 <class 'int'>
6
7 # String -> Float
8 s2 = "45.6"
9 f = float(s2)
10 print(f, type(f))
11 # Output: 45.6 <class 'float'>
12
13 # Integer -> String
14 x = 100
15 text = str(x)
16 print(text, type(text))
17 # Output: "100" <class 'str'>
18
19 # Float -> Integer (desimal dibuang)
20 pi = 3.14
21 n = int(pi)
22 print(n, type(n))
23 # Output: 3 <class 'int'>
24
25
```

B. Collection (Struktur Data Dasar)

Python punya beberapa tipe koleksi bawaan yang sering dipakai:

1. List

- Menyimpan banyak nilai dalam urutan.
- Bisa berisi berbagai tipe data.
- Mutable (bisa diubah).

Contoh:



```
1 buah = ["apel", "mangga", "pisang"]
2 print(buah[0])      # akses elemen
3 buah.append("jeruk") # tambah elemen
4 print(buah)
5
```

2. Tuple

- Mirip list, tapi immutable (tidak bisa diubah).
- Cocok untuk data yang sifatnya tetap.

Contoh:

```
1 warna = ("merah", "hijau", "biru")
2 print(warna[1])
3 # warna[0] = "kuning" # ✗ Error karena tuple tidak bisa diubah
4
```

3. Set

- Kumpulan data tanpa urutan dan unik (tidak ada duplikat).
- Berguna untuk operasi himpunan (union, intersection, difference).

Contoh:

```
1 angka = {1, 2, 3, 3, 4}
2 print(angka) # Output: {1, 2, 3, 4}
3 angka.add(5)
4 print(angka)
5
```

4. Dictionary (dict)

- Menyimpan data dalam bentuk key-value.
- Sangat berguna untuk data terstruktur.

Contoh:

```
1 mahasiswa = {
2     "nama": "Kevin",
3     "nim": "12345",
4     "jurusan": "Sistem Informasi"
5 }
6 print(mahasiswa["nama"])
7 mahasiswa["angkatan"] = 2023 # tambah key-value
8 print(mahasiswa)
9
```

C. Operator

Operators adalah suatu simbol yang berguna untuk menjalankan operasi aritmatika atau logika. Value yang dioperasikan oleh operator disebut operand. Dalam python ada beberapa operator, sebagai berikut:

a. Arithmetic Operators

Operator aritmatik digunakan untuk menjalankan operasi matematika seperti penjumlahan, pengurangan, perkalian, dll.

Operator	Arti	Contoh
+	Menjumlahkan dua operand atau unary plus	$x + y + 2$
-	Mengurangi operand kanan dari kiri atau unary minus	$x - y - 2$
*	Mengalikan operand kiri dengan kanan	$x * y$
/	Membagikan operand kiri dengan kanan(akan menghasilkan float)	x / y
%	Modulus – hasil bagi dari operand kiri dengan kanan	$x \% y$
//	Floor Division – pembagian yang hasilnya menjadi bilangan bulat disesuaikan ke kiri pada garis bilangan	$x // y$
**	Pangkat – operand kiri dipangkatkan oleh operand kanan	$x ** y$

a. Comparison (Relational) Operators

Operator komparasi digunakan untuk membandingkan value dan akan mengembalikan nilai True atau False berdasarkan kondisinya.

Operator	Arti	Contoh
----------	------	--------

>	Lebih besar dari – True jika operand kiri lebih besar dari kanan	$x > y$
<	Kurang dari – True jika operand kiri lebih kecil dari kanan	$x < y$
==	Sama dengan – True jika kedua operand bernilai sama	$x == y$
!=	Tidak sama dengan – True jika kedua operand tidak bernilai sama	$x != y$
>=	Lebih besar atau sama dengan – True jika operand kiri lebih besar atau sama dengan kanan	$x >= y$
<=	Lebih kecil atau sama dengan – True jika operand kiri lebih kecil atau sama dengan kanan	$x <= y$

b. Logical (Boolean) Operators

Logical Operators digunakan untuk melakukan operasi antara dua atau lebih ekspresi boolean.

Operator	Arti	Contoh
and	True jika kedua operand bernilai True	$x \text{ and } y$
or	True jika kedua atau salah satu operand bernilai True	$x \text{ or } y$
not	True jika operand bernilai False	$\text{not } x$

c. Bitwise Operators

Operator bitwise bertindak pada operand seolah-olah mereka adalah string digit binary.

Pada tabel di bawah misalkan $x = 10$ (0000 1010) dan $y = 4$ (0000 0100)

Operator	Arti	Contoh
&	Bitwise AND	$x \& y = 0$ (0000 0000)
	Bitwise OR	$x y = 14$ (0000 1110)
~	Bitwise NOT	$\sim x = -11$ (1111 0101)
^	Bitwise XOR	$x ^ y = 14$ (0000 1110)
>>	Bitwise right shift	$x >> 2 = 2$ (0000 0010)
<<	Bitwise left shift	$x << 2 = 40$ (0010 1000)

d. Assignment Operators

Operator assignment digunakan dalam python untuk menetapkan nilai ke variabel.

Operator	Contoh	Arti
=	$x = 5$	$x = 5$
+=	$x += 5$	$x = x + 5$
-=	$x -= 5$	$x = x - 5$
*=	$x *= 5$	$x = x * 5$
/=	$x /= 5$	$x = x / 5$
%=	$x \%= 5$	$x = x \% 5$
//=	$x //= 5$	$x = x // 5$
**=	$x **= 5$	$x = x ** 5$
&=	$x \&= 5$	$x = x \& 5$
=	$x = 5$	$x = x 5$
^=	$x ^= 5$	$x = x ^ 5$

<code>>>=</code>	<code>x >>= 5</code>	<code>x = x >> 5</code>
<code><<=</code>	<code>x <<= 5</code>	<code>x = x << 5</code>

e. Identity Operator

Operator identitas dalam Python digunakan untuk memeriksa apakah dua variabel atau objek merujuk pada objek yang sama.

Operator	Arti	Contoh
<code>is</code>	True jika kedua operand identik	<code>x is True</code>
<code>is not</code>	True jika operand tidak identik	<code>x is not True</code>

f. Membership Operator

Membership operator dalam Python digunakan untuk memeriksa keberadaan suatu elemen dalam kumpulan data.

Operator	Arti	Contoh
<code>in</code>	True jika value/variable ditemukan dalam collections	<code>x in listY</code>
<code>not in</code>	True jika value/variable tidak ditemukan dalam collection	<code>x not in listY</code>

g. Unary Operator

Unary Operator adalah operator yang hanya membutuhkan satu operand untuk melakukan operasi. Operand ini bisa berupa konstanta, variabel, atau ekspresi yang digunakan untuk melakukan perubahan nilai pada operand yang diberikan. Berikut ini contohnya: Misal $x = 10$ (0000 1010) dan $y = \text{True}$

Operator	Arti	Contoh
<code>-</code>	Mengubah tanda dari operand dan digunakan untuk membuat angka menjadi positif dan negatif	<code>-x = -10</code>

\sim	Sebuah operator yang membalikkan setiap bit dari operandnya	$\sim x = -11$
not	Mengembalikan True jika operand adalah False dan sebaliknya	not y = False