

# DASAR PYTHON

## BAB 2 : VARIABLES, DATA TYPE, NUMBER & CASTING

---

### 2.1 Tujuan

Mahasiswa mengenal materi Variables, Data Type, Number dan Casting pada bahasa pemrograman python

### 2.2 Ulasan Materi

#### A. Variables

##### 1. Variables

Variabel adalah wadah untuk menyimpan nilai data..

##### 2. Membuat Variabel

Python tidak memiliki perintah untuk mendeklarasikan variabel. Variabel dibuat saat pertama kali menetapkan nilai. Contoh :

```
x = 5
y = "John"
print(x)
print(y)
```

Output :

```
5
John
```

Variabel tidak perlu dideklarasikan dengan tipe tertentu, dan bahkan dapat diubah tipenya setelah ditetapkan. Contoh :

```
x = 4          # x is of type int
x = "Sally"    # x is now of type str
print(x)
```

Output :

```
Sally
```

### 3. Casting

Jika ingin menentukan tipe data dari suatu variabel, ini dapat dilakukan dengan casting. Contoh :

```
x = str(3)      # x will be '3'
y = int(3)      # y will be 3
z = float(3)    # z will be 3.0
print(x)
print(y)
print(z)
```

Output :

```
3
3
3.0
```

### 4. Mendapatkan Type Data

Anda bisa mendapatkan tipe data variabel dengan fungsi `type()`. Contoh :

```
x = 5
y = "John"
print(type(x))
print(type(y))
```

Output :

```
<class 'int'>
<class 'str'>
```

### 5. Single atau Double Quotes

Variabel string dapat dideklarasikan dengan menggunakan tanda kutip tunggal atau ganda. Contoh :

```
x = "John"
# is the same as
x = 'John'
```

Output :

```
John
John
```

## 6. Case-Sensitive

Nama variable case-sensitive. Contoh membuat dua variable :

```
a = 4
A = "Sally"

print(a)
print(A)
```

Output :

```
4
Sally
```

## 7. Nama Variabel

Sebuah variabel dapat memiliki nama pendek (seperti x dan y) atau nama yang lebih deskriptif (umur, carname, total\_volume). Aturan untuk variabel Python :

- Nama variabel harus dimulai dengan huruf atau karakter garis bawah
- Nama variabel tidak boleh diawali dengan angka
- Nama variabel hanya boleh berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan \_)

- d. Nama variabel peka huruf besar/kecil (usia, Usia, dan AGE adalah tiga variabel berbeda)

```
myvar = "John"
my_var = "John"
_my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"

print(myvar)
print(my_var)
print(_my_var)
print(myVar)
print(MYVAR)
print(myvar2)
```

Output :

```
John
John
John
John
John
John
John
```

## 8. Nama Variabel Multi Words

Nama variabel dengan lebih dari satu kata bisa jadi sulit dibaca. Ada beberapa teknik yang dapat Anda gunakan untuk membuatnya lebih mudah dibaca:

- a. Camel Case

Setiap kata, kecuali yang pertama, dimulai dengan huruf capital :

```
myVariableName = "John"
```

- b. Pascal Case

Setiap kata dimulai dengan huruf capital :

```
MyVariableName = "John"
```

c. Snake Case

Setiap kata dipisahkan oleh karakter garis bawah :

```
my_variable_name = "John"
```

## 9. Banyak Nilai ke Beberapa Variabel

Python memungkinkan untuk menetapkan nilai ke beberapa variabel dalam satu baris. Contoh :

```
x, y, z = "Orange", "Banana", "Cherry"  
print(x)  
print(y)  
print(z)
```

Output :

```
Orange  
Banana  
Cherry
```

## 10. Satu Nilai ke Beberapa Variabel

Dan juga dapat menetapkan nilai yang sama ke beberapa variabel dalam satu baris.

Contoh :

```
x = y = z = "Orange"  
print(x)  
print(y)  
print(z)
```

Output :

```
Orange
Orange
Orange
```

## 11. Membongkar Koleksi

Jika Anda memiliki kumpulan nilai dalam daftar, tuple dll. Python memungkinkan Anda untuk mengekstrak nilai ke dalam variabel. Ini disebut membongkar (unpacking). Contoh membongkar list :

```
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

Output :

```
apple
banana
cherry
```

## 12. Variabel Keluaran (Output Variables)

Fungsi Python print() sering digunakan untuk output variabel. Contoh :

```
x = "Python is awesome"
print(x)
```

Output :

```
Python is awesome
```

Dalam fungsi print(), menampilkan beberapa variabel, dipisahkan dengan koma.

Contoh :

```
x = "Python"  
y = "is"  
z = "awesome"  
print(x, y, z)
```

Output :

```
Python is awesome
```

Juga dapat menggunakan operator + untuk menampilkan banyak variable. Contoh :

```
x = "Python "  
y = "is "  
z = "awesome"  
print(x + y + z)
```

Output :

```
Python is awesome
```

Untuk angka, karakter + berfungsi sebagai operator matematika. Contoh :

```
x = 5  
y = 10  
print(x + y)
```

Output :

```
15
```

Dalam fungsi print(), ketika mencoba menggabungkan string dan angka dengan operator +, Python akan memberikan error. Contoh :

```
x = 5
y = "John"
print(x + y)
```

Output :

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Cara terbaik untuk menampilkan banyak variabel dalam fungsi print() adalah dengan memisahkannya dengan koma, yang bahkan mendukung tipe data yang berbeda. Contoh :

```
x = 5
y = "John"
print(x, y)
```

Output :

```
5 John
```

### 13. Variabel Global

Variabel yang dibuat di luar fungsi (seperti dalam semua contoh di atas) dikenal sebagai variabel global. Variabel global dapat digunakan oleh semua orang, baik di dalam fungsi maupun di luar. Contoh membuat variabel di luar fungsi, dan gunakan di dalam fungsi :

```
x = "awesome"

def myfunc():
    print("Python is " + x)

myfunc()
```



Output :

```
Python is awesome
```

Jika Anda membuat variabel dengan nama yang sama di dalam suatu fungsi, variabel ini akan bersifat lokal, dan hanya dapat digunakan di dalam fungsi tersebut. Variabel global dengan nama yang sama akan tetap seperti semula, global dan dengan nilai aslinya. Contoh membuat variabel di dalam fungsi, dengan nama yang sama dengan variabel global :

```
x = "awesome"

def myfunc():
    x = "fantastic"
    print("Python is " + x)

myfunc()

print("Python is " + x)
```

Output :

```
Python is fantastic
Python is awesome
```

#### 14. Kata Kunci global

Biasanya, ketika Anda membuat variabel di dalam suatu fungsi, variabel itu bersifat lokal, dan hanya dapat digunakan di dalam fungsi itu. Untuk membuat variabel global di dalam suatu fungsi, dapat menggunakan kata kunci global. Contoh jika menggunakan kata kunci global, variabel tersebut termasuk dalam lingkup global :

```
def myfunc():
    global x
    x = "fantastic"
```

```
myfunc()

print("Python is " + x)
```

Output :

```
Python is fantastic
```

Juga, gunakan kata kunci global jika ingin mengubah variabel global di dalam suatu fungsi. Contoh untuk mengubah nilai variabel global di dalam suatu fungsi, lihat variabel tersebut dengan menggunakan kata kunci global :

```
x = "awesome"

def myfunc():
    global x
    x = "fantastic"

myfunc()

print("Python is " + x)
```

Output :

```
Python is fantastic
```

## B. Data Type

### 1. Data Type Bawaan

Dalam pemrograman, tipe data merupakan konsep penting. Variabel dapat menyimpan data dari tipe yang berbeda, dan tipe yang berbeda dapat melakukan hal yang berbeda.

Python memiliki tipe data bawaan berikut secara default, dalam kategori ini :

- a. Text Type : str
- b. Numeric Types : int, float, complex
- c. Sequence Types : list, tuple, range
- d. Mapping Type : dict
- e. Set Types : set, frozenset
- f. Boolean Type : bool
- g. Binary Types : bytes, bytearray, memoryview
- h. None Type : NoneType

## 2. Mendapatkan Data Type

Anda bisa mendapatkan tipe data objek apa pun dengan menggunakan fungsi `type()`. Contoh print tipe data variabel `x` :

```
x = 5
print(type(x))
```

Output :

```
<class 'int'>
```

## 3. Men-setting Data Type

Jika ingin menentukan tipe data dari suatu variabel, ini dapat dilakukan dengan casting. Di Python, tipe data diatur saat menetapkan nilai ke variable. Contoh :

Example	Data Type
<code>x = "Hello World"</code>	str
<code>x = 20</code>	int
<code>x = 20.5</code>	float
<code>x = 1j</code>	complex
<code>x = ["apple", "banana", "cherry"]</code>	list
<code>x = ("apple", "banana", "cherry")</code>	tuple
<code>x = range(6)</code>	range
<code>x = {"name": "John", "age": 36}</code>	dict
<code>x = {"apple", "banana", "cherry"}</code>	set

x = frozenset({"apple", "banana", "cherry"})	frozenset
x = True	bool
x = b"Hello"	bytes
x = bytearray(5)	bytearray
x = memoryview(bytes(5))	memoryview

#### 4. Mensetting Data Type Spesifik

Jika ingin menentukan tipe data, dapat menggunakan fungsi constructor berikut :

Example	Data Type
x = str("Hello World")	str
x = int(20)	int
x = float(20.5)	float
x = complex(1j)	complex
x = list(("apple", "banana", "cherry"))	list
x = tuple(("apple", "banana", "cherry"))	tuple
x = range(6)	range
x = dict(name="John", age=36)	dict
x = set(("apple", "banana", "cherry"))	set
x = frozenset(("apple", "banana", "cherry"))	frozenset
x = bool(5)	bool
x = bytes(5)	bytes
x = bytearray(5)	bytearray
x = memoryview(bytes(5))	memoryviews

### C. Number

#### 1. Number

Ada tiga tipe numerik dalam Python :

- a. int
- b. float
- c. complex

Variabel tipe numerik dibuat saat menetapkan nilai. Contoh :

```
x = 1      # int
y = 2.8    # float
z = 1j     # complex
```

Untuk memverifikasi jenis objek apa pun dengan Python, gunakan fungsi `type()`.

Contoh :

```
print(type(x))
print(type(y))
print(type(z))
```

Output :

```
<class 'int'>
<class 'float'>
<class 'complex'>
```

## 2. Int

Int, atau bilangan bulat (integer), adalah bilangan bulat, positif atau negatif, tanpa desimal, dengan panjang tak terbatas. Contoh integer :

```
x = 1
y = 35656222554887711
z = -3255522
print(type(x))
print(type(y))
print(type(z))
```

Output :

```
<class 'int'>
<class 'int'>
<class 'int'>
```

## 3. Float

Float, atau "floating point number" adalah angka, positif atau negatif, yang mengandung satu atau lebih desimal. Contoh float :

```
x = 1.10
y = 1.0
z = -35.59
print(type(x))
print(type(y))
print(type(z))
```

Output :

```
<class 'float'>
<class 'float'>
<class 'float'>
```

Float juga bisa berupa scientific number dengan "e" untuk menunjukkan kekuatan 10. Contoh float :

```
x = 35e3
y = 12E4
z = -87.7e100
print(type(x))
print(type(y))
print(type(z))
```

Output :

```
<class 'float'>
<class 'float'>
<class 'float'>
```

#### 4. Complex

Complex number ditulis dengan "j" sebagai bagian imajiner. Contoh complex:

```
x = 3+5j
y = 5j
z = -5j
print(type(x))
print(type(y))
print(type(z))
```

Output :

```
<class 'complex'>
<class 'complex'>
<class 'complex'>
```

## 5. Type Conversion

Anda dapat mengonversi dari satu type ke type lainnya dengan metode `int()`, `float()`, dan `complex()`. Contoh Konversi dari satu type ke type lainnya :

```
x = 1      # int

y = 2.8    # float
z = 1j     # complex

#convert from int to float:
a = float(x)

#convert from float to int:
b = int(y)

#convert from int to complex:
c = complex(x)

print(a)
print(b)
print(c)
```

```
print(type(a))
print(type(b))
print(type(c))
```

Output :

```
1.0
2
(1+0j)
<class 'float'>
<class 'int'>
<class 'complex'>
```

## 6. Random Number

Python tidak memiliki fungsi random() untuk membuat angka acak, tetapi Python memiliki modul bawaan yang disebut random yang dapat digunakan untuk membuat angka acak. Contoh impor modul random, dan tampilkan nomor random antara 1 dan 9 :

```
import random
print(random.randrange(1, 10))
```

Output :

```
3
```

## D. Casting

### 1. Casting

Menentukan Variable Type. Mungkin ada saatnya Anda ingin menentukan tipe pada variabel. Ini bisa dilakukan dengan casting. Python adalah bahasa berorientasi objek, dan karena itu menggunakan kelas untuk mendefinisikan tipe data, termasuk tipe primitifnya.

Casting di python karena dilakukan dengan menggunakan fungsi constructor :



- a. `int()` - membangun bilangan integer dari literal bilangan bulat, literal float (dengan menghapus semua desimal), atau literal string (menyediakan string mewakili bilangan bulat)
- b. `float()` - membuat bilangan float dari literal integer, literal float, atau literal string (menyediakan string mewakili float atau integer)
- c. `str()` - membangun string dari berbagai tipe data, termasuk string, literal integer dan literal float

Contoh integers:

```
x = int(1)
y = int(2.8)
z = int("3")
print(x)
print(y)
print(z)
```

Output :

```
1
2
3
```

Contoh Floats:

```
x = float(1)
y = float(2.8)
z = float("3")
w = float("4.2")
print(x)
print(y)
print(z)
print(w)
```

Output :

```
1.0
2.8
3.0
4.2
```

Contoh string :

```
x = str("s1")
y = str(2)
z = str(3.0)
print(x)
print(y)
print(z)
```

Output :

```
s1
2
3.0
```

## 2.3 Praktikum

**Contoh : Menampilkan banyak variabel menggunakan operator +**

Pada contoh ini kita akan belajar membuat tiga buah variabel yaitu x, y dan z. Untuk variabel x bernilai string "Python ", variabel y bernilai string "is " dan variabel z bernilai string "awesome". Kemudian kita akan menggabungkan nilai variabel x, y dan z menggunakan operator +. Berikut ini adalah langkah-langkahnya :

1. Tuliskan kode yang mendefinisikan variabel dengan nama x dengan nilai string "Python "

```
x = "Python "
```

2. Tuliskan kode yang mendefinisikan variabel dengan nama y dengan nilai string "is "

```
y = "is "
```

3. Tuliskan kode yang mendefinisikan variabel dengan nama z dengan nilai string “awesome”

```
z = "awesome"
```

4. Buatlah fungsi bernama penggabungan, yang memiliki parameter variabel x, y, dan z

```
def penggabungan(x, y, z):
```

5. Kemudian didalam fungsi penggabungan tuliskan perintah return dan gabungkan variabel x, y dan z menggunakan operator +

```
    return x+y+z
```

6. Berikutnya tampilkan dengan perintah print dan panggil fungsi penggabungan yang memiliki parameter variabel x, y dan z seperti pada source code dibawah ini :

```
print(penggabungan(x, y, z))
```

#### **Percobaan 4 : Menampilkan banyak variabel menggunakan operator +**

Sekarang kerjakan percobaan 4 dengan mengikuti langkah-langkah dibawah ini !

1. Buatlah variabel ***nama*** yang memiliki nilai “Qorinda “
2. Buatlah variabel ***lengkap*** yang memiliki nilai “Yulvarisma”
3. Kemudian buat fungsi ***hasil*** yang memiliki parameter variabel nama dan variabel lengkap
4. Pada fungsi hasil tambahkan perintah return dan gabungkan variabel nama dan variabel lengkap menggunakan operator +
5. Selanjutnya tampilkan dengan menggunakan perintah print serta panggil fungsi hasil yang memiliki parameter nama dan lengkap
6. Jalankan kode program percobaan 4 diatas dengan menekan tombol Check Code Validity