

Praktikum 1 Elemen Dasar Pemrograman Python

Capaian Kompetensi : Mahasiswa menjelaskan elemen-elemen dasar bahasa Pemrograman Python.

Objektif :

1. Variabel &. Type Data
2. Operator
3. Ekspresi

=====

1. Variabel dan Tipe data

Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang tersimpan dalam variabel.

Python mendukung beberapa tipe data untuk keperluan penyimpanan data numerik. Data numerik yang dapat digunakan meliputi bilangan bulat, bilangan riil, dan bilangan kompleks. Semua objek dari tipe numerik tidak dapat diubah nilainya atau bersifat immutable.

Bagian ini menjelaskan tentang masing-masing tipe data tersebut.

1.1 Tipe Numerik

1.1.1 Tipe integer

Terdapat dua tipe data bilangan bulat yang didukung oleh python 3, yaitu integer(int) dan boolean(bool). Python 3 tidak memiliki nilai maksimum untuk tipe int. untuk mengonversi bilangan bulat ke string, gunakan fungsi str(), seperti berikut:

Sebaliknya jika anda ingin mengonversi nilai dari tipe string ke int, gunakan fungsi int().

Buka IDLE Phyton Shell, ketik dan jalankan Latihan praktikum berikut :

Latihan 1.1 int()

```
>>> a = '12345'
>>> type(a)
<class 'str'>
>>> b=int(a)
>>> b
12345
>>> type(b)
```

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

```
>>>
```

Latihan 1.2 str()

```
>>> a = 12345
>>> type(a)
<class 'int'>
>>> b = str(a)
>>> b
'12345'
>>> type(b)
<class 'str'>
>>>
```

Tipe bool digunakan untuk menyatakan tipe logika (boolean). objek dari tipe bool hanya dapat diisi dengan nilai True atau False (huruf T dan F harus di tulis dalam huruf besar). jika di konversikan ke tipe int, nilai true akan menghasilkan nilai 1 dan False menghasilkan nilai 0.

Latihan 1.3 bool

```
>>> int(True)
1
>>> int(False)
0
>>> a = True
>>> type(a)
<class 'bool'>
>>> int(a)
1
>>>
```

Tipe int digunakan untuk menyatakan bilangan bulat, seperti 1, 100, 123 dst. Bilangan bulat dapat ditulis dalam bentuk decimal (basis 10), biner (basis 2), octal (basis 8), dan heksadesimal (basis 16).

Latihan 1.4 basis-integer

```
>>> decimal = 123
>>> biner = 0b1111011
>>> octal = 0o173
>>> heksadesimal = 0x7b
>>>
>>> decimal == biner == octal == heksadesimal
True
>>> type(decimal)
<Class 'int'>
>>>
>>> type(biner)
<Class 'int'>
>>>
```

1.1.2 Tipe Riil

Untuk merepresentasikan data bertipe bilangan riil (mengandung angka di belakang koma), Python menyediakan tipe float. Bilangan dengan tipe float ditulis menggunakan tanda titik (.), Anda juga dapat menulis bilangan riil dalam bentuk eksponen, seperti berikut:

```
>>>r=8.9E-4
```

Notasi diatas menunjukkan nilai 8.9×10^{-4} untuk mengonversi bilangan dengan tipe float ke string, gunakan fungsi `str()`.

Sebaliknya, untuk mengonversi string ke tipe float, gunakan fungsi `float()`.

Latihan 1.5 float

```
>>> a = 123.456
>>> a
123.456
>>> type(a)
< class ' float'
```

Latihan 1.6 float

```
>>> a=8.9e-4
>>> a
0.00089
>>>
```

Latihan 1.7 float

```
>>> a=123.456
>>> type (a)
<class 'float'>
>>> b=str(a)
>>> b
'123.456'
>>> type (b)
<class 'str'>
>>> c= float(b)
>>> c
123.456
>>> type (c)
<class 'float' >
>>>
```

1.1.3 Tipe Bool

Tipe bool digunakan untuk menyatakan data logika, benar atau salah.

Latihan 1.8 Bool

```
>>> a = True
>>> b = False
>>>
>>> type(a)
<Class 'bool'>
>>>
```

```
>>> a and b
False
>>>
>>> not a
False
>>>
```

1.1.4 Tipe Kompleks

Bilangan kompleks adalah bilangan yang mengandung pasangan bilangan dari tipe float. Bagian pertama merupakan bagian riil dan bagian kedua merupakan bagian imajiner, kedua bagian tersebut digabung menggunakan tanda + atau - dan diakhiri dengan huruf j.

Latihan 1.9 Kompleks

```
>>> a=-9+17j
>>> a
(-9+17j)
>>> type(a)
<class 'complex'>
>>> a.real
-9.0
>>> a.imag
17.0
>>>
```

1.2 Tipe String

Dalam Python 3, teks atau yang sering dikenal dengan string, merupakan kumpulan karakter Unicode dan dinyatakan dengan tipe str. String dapat dibentuk dengan menggunakan tiga cara, yaitu :

- Menggunakan tanda petik tunggal
- Menggunakan tanda petik ganda
- Menggunakan tanpa petik satu atau dua yang ditulis sebanyak tiga kali.

Cara Terakhir biasanya hanya digunakan untuk membuat string panjang yang jumlahnya lebih dari satu baris.

Python mendukung adanya karakter escape sequence merupakan runtunan dua karakter atau lebih yang memiliki makna khusus, dan selalu diawali dengan karakter *backslash* ('*backslash*').

Escape Sequence	Keterangan
\\	Backslash
\'	Petik Tunggal
\"	Petik Ganda
\n	Backspace
\n	ASCII linefeed
\r	ASCII Carriage Return
\t	ASCII tab
\v	ASCII vertical tab

Latihan 1.10 String

```
>>> s1 = 'PyQt '
>>> s1
'PyQt '
>>> s2 = "Python "
>>> s2
' Python '
>>> s3 = """Pemrograman GUI dengan Python dan
PyQt """
>>> s3
'Pemrograman GUI dengan Python dan PyQt'
>>> s4 = "\nPemrograman GUI dengan Python dan PyQt\n"
>>> s4
'\nPemrograman GUI dengan Python dan PyQt\n '
>>> b = 'python '
>>> b ' python
'
>>> b=b.capitalize()
>>> b
' Python '
>>> b=b.upper()
>>> b
'PYTHON'
```

```
>>> b=b.lower()
>>> b
'python '
>>> b.isupper()
False
>>> b.islower()
True
>>> s = ' '.join( ['saya','makan','ayam'])
>>> s
'saya makan ayam'
>>> nim= '16610021'
>>> nim.isnumeric() True
>>> nim= '16610o21'
>>> nim.isnumeric ( )
False
>>>
```

Karakter dalam suatu string, bisa diperoleh melalui indeks.

Latihan 1.11 string

```
>>> s="Dipanegara"
>>> s[0]
'D'
>>> s[-1]
'a'
>>> s[0:4]
'Dipa'
>>> s[-6:-1]
'negar'
>>>
```

1.3 Konversi tipe

Untuk mengkonversi nilai dari tipe str ke int, gunakan fungsi int().

Latihan 1.12 konversi

```
>>> a = '123' # bertipe str
>>> a
```

```
'123'
>>> type(a)
<class 'str'>
>>>
>>> b = int(a)
>>> b
123
>>> type(b)
<class 'int'>
>>>
```

1.4 Type Koleksi

Tipe list Tipe tuple Tipe dict Tipe set dan frozenset

2. Ekspresi dan Operator

Ekspresi adalah suatu bentuk penulisan dalam pemrograman yang menghasilkan suatu nilai. Ekspresi terdiri dari gabungan nilai data (Variabel) dan operator. Contoh : >>> 2 + 3 - 4

Operator adalah simbol khusus yang digunakan untuk melakukan operasi-operasi tertentu terhadap data yang dibutuhkan oleh program. Operator yang didukung oleh python 3 antara lain :

a. Operator Penugasan

Operator penugasan(assignment) adalah operator yang menugaskan suatu variable untuk menunjuk ke objek tertentu. Python menggunakan operator 'sama dengan' (=) untuk melakukan proses assignment. >>> a = 9

Perintah tersebut menugaskan variable a untuk menunjuk ke objek dari kelas int, yang bernilai 9.

Python juga mengenal operator Penugasan majemuk sbb :

Operator	Keterangan
a += b	Bentuk singkat dari a = a + b
a -= b	Bentuk singkat dari a = a - b
a *= b	Bentuk singkat dari a = a * b
a /= b	Bentuk singkat dari a = a / b
a //= b	Bentuk singkat dari a = a // b
a %= b	Bentuk singkat dari a = a % b
a **= b	Bentuk singkat dari a = a ** b

Contoh kode berikut akan menunjukkan penggunaan beberapa operator di atas :

Latihan 1.13 Opr_penugasan1

```
>>> a = 10
>>> b = 3
>>> a += b    # a = 10 + 3
>>> a
13
>>> a -= b
>>> a
10
>>> a *= b
>>> a
30
>>> a /= b    # a = 30 / 3
>>> a
10.0
>>>
```

Dalam python, nilai dapat di masukkan kedalam beberapa variabel secara simultan.

Latihan 1.14 opr_penugasan2

```
>>> x = y = z = 20
>>> x
20
>>> y
20
>>> z
20
```

b. Operator Aritmatika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian

/	Pembagian bilangan riil
//	Pembagian bilangan bulat
%	Sisa bagi
**	Perpangkatan

Latihan 1.15 opr_aritmatika

```
>>> 10 + 3
13
>>> 10 - 3
7
>>> 10 * 3
30
>>> 10 / 3
3.3333333333333335
>>> 10 // 3
3
>>> 10 % 3
1
>>> 10 ** 3
1000
```

c. Operator Relasional

Operator	Keterangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

Contoh :

Latihan 1.16 opr_relasi

```
>>> a = 10
>>> b = 9
>>> a == b
False
```

```
>>> a != b    # python 3 tidak mendukung operator <>
True
>>> a > b
True
>>> a < b
False
>>> a >= b
True
>>> a <= b
False
>>>
```

d. Operator Logika

Operator	Keterangan
and	Operator AND
or	Operator OR
not	Operator NOT

Latihan 1.17 opr_logika

```
>>> True and True
True
>>> True and False
False
>>> True or False
True
>>> not True
False
>>> not (not True)
True
>>>
```

e. Operator Bitwise

Operator bitwise adalah operator logika yang bekerja pada bilangan bulat yang akan dikonversi ke dalam bentuk biner (basis 2).

Operator	Keterangan	Contoh
&	Operator AND	$6 \& 7 = 6$
	Operator OR	$5 8 = 13$
^	Operator XOR	$4 \wedge 7 = 3$

>>	Geser bit ke kiri	5 << 1 10
>>	Geser bit ke kanan	6 >> 1 = 3

Contoh :

6 & 7 = 6 dari mana ? 6

← 00000110

7 ← 00000111

----- &

6 ← 00000110

Latihan 1.18 bitwise

```
>>> 6 & 7
```

```
6
```

```
>>>
```

f. Operator Keanggotaan

Operator	Keterangan
In	Memeriksa keanggotaan apakah suatu nilai merupakan anggota di dalam suatu koleksi/string atau tidak
not in	Mengembalikan nilai True jika suatu nilai bukan merupakan anggota di dalam suatu koleksi/string

Contoh :

Latihan 1.19 in

```
>>> data = [10, 20, 30, 40, 50]
```

```
>>> 10 in data
```

```
True
```

```
>>> 90 in data
```

```
False
```

```
>>> 90 not in data
```

```
True
```

```
>>>
```

g. Operator String dalam python, operator + dapat diterapkan untuk objek yang bertipe string, yang berperan sebagai penyambung dua buah objek string.

Latihan 1.20 String

```
>>> bahasa = 'Python'
>>> version = '3.6'
>>>
>>> print(bahasa + ' ' + versi)
Python 3.6
>>>
>>> 'Python 3 ' * 3
'Python 3 Python 3 Python 3 '
>>>
```

Latihan 1.21 String2

```
>>> for i in range(5):
    print('Dipanegara')
Dipanegara
Dipanegara
Dipanegara
Dipanegara
Dipanegara
>>>
```

TUGAS PRAKTIKUM 1 :

-
1. Berapakah nilai dari masing-masing variable setelah penugasan berikut :
 - a. A,B, C = 1, 2, 3
 - b. x, y, = y,x
 - c. c = b= a =50
 - d. n = n + 1
 - e. n+=2
 2. Mengapa variable pada Python dikatakan bertipe dinamis ?
 3. Apakah maksud dari penulisan >>> 2 + 3j
 4. Bagaimana cara mengetahui panjang suatu string ?
 5. Apa hasil perintah berikut : >>> s="Hai..." # Apa kabar ?

6. Apa arti dari ekspresi berikut : `>>> 4^3`
7. Bisakah dua pernyataan ditulis dalam sebuah baris?, jika ya bagaimana caranya ?
8. Tuliskan hasil dari `>>> 7%2`
9. Tuliskan hasil dari `>>> ("x" >= "a") and ("x" <= "t")` 10. Apa hasil dari :
 - a. `>>> x="STMIK\nDipaneegara"`
`>>> x`
 - b. `>>> print (x)`

>> oOo <<