



# **MODUL PRAKTIKUM**

# **DASAR PEMROGRAMAN**

## **INF1004AA**

Oleh :

*INDRIAN*

*1911102441003*

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2019

# Laporan Praktikum 2:

# Dasar Bahasa Python

---

## Pokok Bahasan:

- ❖ Variable
- ❖ Tipe Data
- ❖ Operator

## Tujuan Pembelajaran:

- ✓ Memahami jenis-jenis variable, tipe data dan operator Python 3

## Build-in Tipe Data Atomic:

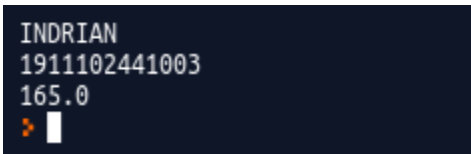
### 1. Variable

#### Percobaan & Latihan 2.1

```
1 a = "INDRIAN" # Tipe data String
2 b = 1911102441003 # Tipe data Integer
3 c = 165.0 # Tipe data Float
4
5 print(a)
6 print(b)
7 print(c)
```

## Jawab:

- a. Output



```
INDRIAN
1911102441003
165.0
>
```

b. Menentukan tipe data

Variable a tipe datanya adalah **String**

Variable b tipe datanya adalah **Int**

Variable c tipe datanya adalah **Float**

## 2. Operator Aritmatika

### Percobaan & Latihan 2.2

```
9   x = 3
10  y = 2
11
12  print(x + y)
13  print(x - y)
14  print(x * y)
15  print(x / y)
16  print(x % y)
17  print(x ** y)
18  print(x // y)
```

**Jawab:**

a) Output



```
5
1
6
1.5
1
9
1

```

b) Penjelasan Fungsi dari operator

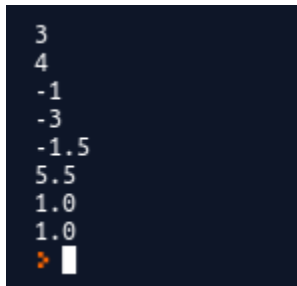
- Operator + Adalah Penjumlahan
- Operator - Adalah Pengurangan
- Operator \* Adalah Perkalian
- Operator / Adalah Penbagian
- Operator \*\* Adalah Pangkat
- Operator // Adalah Penbagian kebawah

### 3. Operator Assignment

#### Percobaan & Latihan 2.3

```
1  x = 3
2
3  print(x)
4
5  x += 1
6  print(x)
7
8  x -= 5
9  print(x)
10
11 x *= 3
12 print(x)
13
14 x /= 2
15 print(x)
16
17 x %= 7
18 print(x)
19
20 x //= 3
21 print(x)
22
23 x **= 3
24 print(x)
```

**a) Output**



```
3
4
-1
-3
-1.5
5.5
1.0
1.0
```

**b) Penjelasan fungsi dari tiap operator assignment**

- Operator **=** Memberikan suatu nilai pada variabel awalnya x belum memiliki nilai diberi operator ini maka akan memiliki nilai, contoh  $x = 3$ , maka x sudah bernilai 3,
- Operator **+=** Memberikan suatu nilai pada variabel yang telah ada dengan hasil penjumlahan variabel tersebut dengan suatu nilai misal  $x = 3$ , dan  $x += 1$  maka hasilnya 4, cara ini juga sama halnya dengan  $x + 1$
- Operator **-=** Memberikan suatu nilai pada variabel yang telah ada dengan hasil pengurangan variabel tersebut dengan suatu nilai, misal  $x = 4$ , dan  $x -= 5$  maka hasilnya -1, cara ini sama halnya dengan  $x - 5$
- Operator **\*=** Memberikan suatu nilai pada variabel yang telah ada dengan hasil perkalian variabel tersebut dengan suatu nilai, misal  $x = -1$ , dan  $x *= 3$  maka hasilnya -3, cara ini sama halnya dengan  $x * 3$
- Operator **/=** Memberikan suatu nilai pada variabel yang telah ada dengan hasil pembagian variabel tersebut dengan suatu nilai, misal  $x = -3$ , dan  $x /= 2$  maka hasilnya -1.5, cara ini sama halnya  $x / 2$
- Operator **%=** Memberikan suatu nilai pada variabel yang telah ada dengan sisa hasil pembagian variabel tersebut dengan suatu nilai, misal  $x = -1.5$ , dan  $x \% 7$  maka hasilnya 5.5, cara ini sama halnya  $x \% 7$
- Operator **//=** Memberikan suatu nilai pada variabel yang telah ada dengan hasil pembagian pembulatan kebawah variabel tersebut dengan suatu nilai, misal  $x = 5.5$ , dan  $x //= 3$  maka hasilnya 1.0, cara ini sama halnya  $x // 1$
- Operator **\*\*=** Memberikan suatu nilai pada variabel yang telah ada dengan hasil eksponen variabel tersebut dengan suatu nilai, misal  $x = 1.0$ , dan  $x **= 1$  maka hasilnya 1.0, cara ini sama halnya  $x ** 1$

## 4. Operator Perbandingan

### Percobaan & Latihan 2.4

#### a) Idle dan Output

```
>>> x = 3
>>> y = 16
>>> z = 10
>>>
>>> x == z
False
>>> x == y
False
>>> x != y
True
>>> x != z
True
>>> x > y
False
>>> x > z
False
>>> x < y
True
>>> x < z
True
>>> x >= y
False
>>> x >= z
False
>>> x <= y
True
>>> x <= z
True
>>> |
```

**b) Penjelasan fungsi dari tiap operator perbandingan sesuai dengan output**

- Operator `==` Untuk membandingkan apakah dua buah nilai sama, misal `x = 3` dan `y = 16`, Dibandingkan `x == y` maka hasilnya `false` karena mereka tidak sama, dan misal `x = 16` dan `y = 16` maka hasilnya `true` karena kedua nilai tersebut sama
- Operator `!=` Untuk membandingkan apakah dua buah nilai tidak sama, misal `x = 3`, dan `y = 16`, Dibandingkan `x != y`, maka hasilnya `true`. Karena mereka tidak sama
- Operator `>` Untuk membandingkan apakah suatu nilai (`x`) lebih besar dari nilai yang lain (`y`), misal `x = 3` dan `y = 16`. Dibandingkan `x > y` maka hasilnya `false`. Karena `x` lebih kecil dari pada `y`
- Operator `<` Untuk membandingkan apakah suatu nilai (`x`) lebih besar dari nilai yang lain (`y`), misal `x = 3` dan `y = 16`. Dibandingkan `x < y` maka hasilnya `true`. Karena `y` lebih besar dari pada `x`
- Operator `>=` Untuk membandingkan apakah suatu nilai (`x`) lebih besar dari nilai yang lain (`y`)
- Operator `<=` Untuk membandingkan apakah suatu nilai (`x`) lebih besar dari nilai yang lain (`y`)

## 5. Operator Logika

### Percobaan & Latihan 2.5

**a) Idle dan Output**

```
>>> x = 3
>>> x < 5 and x < 10
True
>>> x < 5 or x < 4
True
>>> not x < 5 and x < 10
False
>>> |
```

**b) Penjelasan Fungsi Operator Logika**

- AND** = Akan menghasilkan `True` jika semua kondisi bernilai `True`, dan menghasilkan nilai `false` jika minimal satu atau semua kondisi bernilai `False`
- OR** = Akan menghasilkan `True` jika salah satu kondisi bernilai `True`, dan menghasilkan nilai `false` jika semua kondisi bernilai `False`
- NOT** = Digunakan untuk membalikan suatu nilai, jika bernilai `false` akan dibalikan menjadi nilai `true` dan sebaliknya

## 6. Operator Identitas

### Percobaan & Latihan 2.6

#### a) Idle Dan Output

```
>>> x = 3
>>> y = 16
>>> x is y
False
>>> x is not y
True
>>> |
```

#### b) Penjelasan Operator Identitas

- Is = Mengembalikan nilai True jika kedua variabel merupakan objek yang sama, dan False jika sebaliknya
- Is not = Mengembalikan nilai True jika kedua variabel merupakan objek yang berbeda, dan False jika sebaliknya

## 7. Operator Keangotaan

### Percobaan & Latihan 2.7

#### a) Idle Dan Output

```
>>> x = "indrian"
>>>
>>> "a" in x
True
>>> "i" not in x
False
>>> "u" in x
False
>>> "u" not in x
True
>>> |
```



### b) Penjelasan Operator Keangotaan

- `in` = Mengembalikan True jika sebuah “sequence” dengan nilai spesifik terdapat didalam suatu objek
- `not in` = Mengembalikan True jika sebuah “sequence” dengan nilai spesifik tidak terdapat didalam suatu objek

## 8. Operator Bitwise

### Percobaan & Latihan 2.8

#### a) Idle Dan Output

```
>>> x = 3
>>> y = 12
>>>
>>> x & y
0
>>> x | y
15
>>> x ^ y
15
>>> x << y
12288
>>> x >> y
0
>>> |
```

## b) Penjelasan Operator Bitwise

- Operator **&**, Nama **AND**, Operator ini menyalin sebuah bit kehasil jika terdapat pada kedua operan
- Operator **|**, Nama **OR**, Operator ini menyalin sebuah bit kehasil jika terdapat pada salah satu operan.
- Operator **^**, Nama **XOR**, Operator ini menyalin sebuah bit kehasil jika terdapat pada kedua operan.
- Operator **~**, Nama **NOT**, Menghasilkan kebalikan nilai dari setiap bit
- Operator **<<**, Nama **Zero fill left shift**, Menggeser setiap bit kearah kiri dan/atau menambahkan digit 0 pada bagian kanan bilangan
- Operator **>>**, Nama **Signed right shift**, Menggeser setiap bit kearah kanan

## 9. Perintah Komentar Pada Python

### Percobaan & Latihan 2.9

```
name = "INDRIAN" # Variable ini bertipekan String yang berisikan nama saya
date = 16 # Variable ini bertipekan Int yang berisikan tanggal lahir saya

# Menggunakan Operator Aritmatika
print(date - 12) # date bernilai 16 dikurang dengan 12, maka hasilnya 4

# Menggunakan Operator Assignment
date += 16
print(date) # date bernilai 16 ditambah sama dengan 4, hasilnya 32

# Menggunakan Operator Perbandingan
print(name != date) # name berisi INDRIAN dan tidak sama dengan date = 16.

# Menggunakan Operator Logika
logic = not 16 < 4 # 4 lebih besar dari pada 16 adalah salah maka nilainya FALSE
print(logic) # dengan operator not maka nilainya berubah sebaliknya

# Menggunakan Operator Identitas
print(name is not "INDRYAN") # INDRYAN tidak sama dengan INDRIAN maka dia true

# Menggunakan Operator Keanggotaan
print("R" in name) # Pada sequence "R" terdapat nilai spesifik dengan INDRIAN
# Maka hasilnya TRUE

# Menggunakan Operator Bitwise
print(date | 4) # date bernilai 32(100000) digantikan dengan 4(100)
# Menggunakan operator | (or) maka hasilnya 100100(36)
```

### Output

```
4
32
True
True
True
True
20
>>> |
```

## Summary Dan Analisa

### Latihan 2.1

Variabel adalah lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ini berarti bahwa ketika kita membuat sebuah variabel Anda memesan beberapa ruang di memori. Variabel menyimpan data yang dilakukan selama program dieksekusi, yang nantinya isi dari variabel tersebut dapat diubah oleh operasi - operasi tertentu pada program yang menggunakan variabel. Variabel dapat menyimpan berbagai macam tipe data. Di dalam pemrograman Python, variabel mempunyai sifat yang dinamis, artinya variabel Python tidak perlu dideklarasikan tipe data tertentu dan variabel Python dapat diubah saat program dijalankan.

### Latihan 2.2

Operator Aritmatika adalah operator matematis dalam pemrograman yang digunakan untuk menyelesaikan operasi matematika dalam pemrograman. Terdapat beberapa jenis operator aritmatika, yaitu: tambahan, pengurangan, perkalian, pembagian, modulo, dll.

### Latihan 2.3

Operator Assignment digunakan dalam Python untuk menetapkan nilai ke variabel. `a = 5` adalah operator assignment sederhana yang memberikan nilai 5, Ada berbagai operator gabungan di Python seperti `+= 5` yang menambah variabel dan kemudian menetapkan nilainya langsung.

#### **Latihan 2.4**

Operator perbandingan dipakai untuk membandingkan 2 buah nilai, apakah nilai tersebut sama besar, lebih kecil, lebih besar, dst. Hasil dari operator perbandingan ini adalah boolean True atau False.

Operator perbandingan ini biasa dipakai dalam proses pengambilan keputusan atau percabangan kode program. Sebagai contoh, jika angka pertama lebih besar dari kedua, maka jalankan perintah ini. Atau jika variabel password == 'qwerty' maka berikan hak akses. Operator perbandingan kadang disebut juga dengan operator relasional karena kita membandingkan hubungan (relasi) sebuah nilai dengan nilai lainnya.

#### **Latihan 2.5**

Singkatnya Operator logika digunakan untuk membuat operasi logika, seperti logika AND, OR, dan NOT.

#### **Latihan 2.6**

Operator identitas adalah operator yang bisa dipakai untuk memeriksa apakah nilai sebuah variabel ada di tempat yang sama (di memory) atau tidak. Operator ini dikenal juga sebagai identity operators.

#### **Latihan 2.7**

Operator keanggotaan adalah operator yang dipakai untuk memeriksa apakah suatu nilai ada di dalam sebuah himpunan atau tidak. Himpunan yang dimaksud terdiri dari tipe data seperti string, int, float. Operator ini dikenal juga sebagai membership operators.

#### **Latihan 2.8**

Bitwise adalah operator khusus untuk menangani operasi logika bilangan biner dalam bentuk bit. Bilangan biner sendiri merupakan jenis bilangan yang hanya terdiri dari 2 jenis angka, yakni 0 dan 1. Jika nilai asal yang dipakai bukan bilangan biner, akan dikonversi secara otomatis oleh bahasa Python menjadi bilangan biner. Misalnya 7 desimal = 0111 dalam bilangan biner. Dalam penerapannya, operator bitwise tidak terlalu sering dipakai, kecuali anda sedang membuat program yang harus memproses bit-bit komputer. Selain itu operator ini cukup rumit dan harus memiliki pemahaman tentang sistem bilangan biner.

#### **Latihan 2.9**

Komentar merupakan baris kode yang tidak akan dieksekusi. Komentar digunakan untuk memberikan informasi tambahan dan untuk menonaktifkan kode.

## **Kesimpulan**

Variable dan operator pada bahasa python penggunaan sangat gampang dipahami sama halnya pada bahasa pemograman modern seperti sekarang seperti kotlin, dart, yang selalu sudah menerapkan clean, simple, readable. Variable kita akan siap memulai membuat program pada tingkat selanjutnya