



MODUL PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

INF1008

Penyusun :

Naufal Azmi Verdikha, M.Eng.

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2018

Praktikum 1:

Data Numerik

Pokok Bahasan:

- ❖ Review *Python* 3
- ❖ Tipe Data Numerik

Tujuan Pembelajaran:

- ✓ Meninjau kembali Bahasa Pemrograman *Python* 3.
- ✓ Mengenal tipe data numerik.
- ✓ Menggunakan *syntax* numerik *Python* 3.

Teori Dasar:

1. Review *Python* 3

Python adalah bahasa pemrograman yang modern, mudah dipelajari, dan berorientasi objek. *Python* memiliki seperangkat tipe data bawaan yang kuat dan konstruksi kontrol yang mudah digunakan. *Python* adalah bahasa interpreter, hal ini paling mudah ditinjau dengan hanya melihat dan menggambarkan sesi interaktif. Perlu diingat bahwa interpreter menampilkan *prompt* `>>>` yang familiar dan kemudian mengevaluasi konstruksi *Python* yang akan diberikan. Berikut ini adalah contoh penggunaan *Python* 3 yang menampilkan *prompt*, fungsi `print`, hasil output, dan *prompt* berikutnya:

Contoh 1.1

```
>>> print("Algoritma dan Struktur Data")
Algoritma dan Struktur Data
>>>
```

Terdapat banyak jenis data yang dapat disimpan di dalam memori. Misalnya, usia seseorang disimpan sebagai nilai numerik dan alamatnya disimpan sebagai karakter alfanumerik. *Python* memiliki berbagai tipe data standar yang digunakan untuk menentukan operasi yang mungkin dilakukan pada data dan metode penyimpanan untuk masing-masing data. Berikut adalah tipe data standar yang dimiliki oleh *Python*:

- *Numbers*
- *String*
- *List*
- *Tuple*
- *Dictionary*

2. Tipe Data Numerik

Bahasa pemrograman *Python* 3 mendukung 3 tipe data numerik yang berbeda, yaitu:

- *integer* (bilangan integer)
- *float* (bilangan pecahan atau desimal)
- *complex* (bilangan kompleks)

Tabel 1. 1

int	float	complex
10	0.0	3.14j
100	15.20	45.j
-786	-21.9	9.322e-36j
080	32.3+e18	.876j
-0490	-90.	-.6545+0j
-0x260	-32.54e100	3e+26j
0x69	70.2-E12	4.53e-7j

2.1. Integer

Python merepresentasikan *integer* menggunakan tipe *int*. Ukuran bilangan *integer* pada *Python* dibatasi oleh memori mesin (rentangnya tergantung pada kompiler bawaan *C* atau *Java*), dengan panjang setidaknya 32-bit (4 byte)¹. Untuk mengetahui seberapa banyak byte yang terdapat di sebuah bilangan *integer*, dapat menggunakan metode `int.bit_length()`, berikut contoh penggunaannya:

Contoh 1. 2

```
>>> (999).bit_length()
10
```

Untuk dapat mengubah sebuah objek *s* kedalam bentuk *integer*, dapat menggunakan metode `int(s, base)`:

Contoh 1. 3

```
>>> s = '11'
>>> d = int(s)
>>> print(d)
11
>>> b = int(s, 2)
>>> print(b)
3
```

2.2. Float

Python merepresentasikan bilangan pecahan atau desimal menggunakan tipe *float*. Ketika menggunakan *single precision*, sebuah 32-bit *float* direpresentasikan sebagai: 1 bit untuk *sign* (tanda bilangan, 1 untuk tanda negatif, 0 untuk tanda positif), ditambah dengan 23 bit untuk *significant digits* (*mantissa*), dan ditambah 8 bit untuk bilangan eksponen. Untuk dapat mengubah sebuah objek *s* dengan menggunakan metode `float(s)`, berikut contoh penggunaannya:

Contoh 1. 4

```
>>> s = "23"
>>> d = float(s)
>>> print(d)
23.0
```

2.3. Complex

Python merepresentasikan tipe data bilangan *complex* dengan memberikan sebuah *value* = *real* + *imag**1j (contoh : $z = 3 + 4j$). *Python* dapat mengonversi sebuah data *string* atau sebuah data numerik menjadi bilangan *complex*. Jika parameter pertama adalah *string*, maka akan ditafsirkan sebagai bilangan kompleks dan fungsinya harus dipanggil tanpa parameter kedua. Parameter kedua tidak pernah menjadi *string*. Setiap argumen dapat berupa tipe numerik apa pun (termasuk kompleks). Jika *imag* dihilangkan, *default*-nya adalah nol dan konstruktor berfungsi sebagai konversi numerik seperti *int* dan *float*. Jika kedua argumen dihilangkan, menghasilkan 0j. Contoh penggunaan:

Contoh 1. 5

```
>>> complex(3,5)
(3+5j)
>>> complex('1+2j')
(1+2j)
```

Tugas Pendahuluan:

1. Jelaskan parameter yang terdapat di *syntax* dibawah ini:
 - a) `int(x, base)`
 - b) `float(x)`
 - c) `complex([real,[imag]])`
2. Sebutkan dan jelaskan metode lain untuk merepresentasikan 3 tipe data numerik (selain yang terdapat pada contoh diatas)!

Percobaan & Latihan:

1. Berikan nilai beberapa objek sebagai berikut :

x = ... //isi dengan digit ke-10 & ke-11 dari NIM anda

y = ... //isi dengan digit ke-12 & ke-13 dari NIM anda, jika kedua digit tersebut menghasilkan 00, maka ubah dengan nilai 14.

Jalankan perintah berikut:

- a) Ha = x+y; print(Ha)
- b) Hb = Ha + 10.5; print("Hb = ",Hb)
- c) Hc = int(Hb); print(Hc)
- d) Hd = Ha / Hb; print(Hd)
- e) isinstance(Hd, int)
- f) isinstance(Hd, float)
- g)

```
>>> if isinstance(Hd, int):  
        print("Hd adalah bilangan integer!")  
else: print("Hd adalah bilangan float!")
```

2. Jalankan perintah-perintah berikut:

(999).bit_length()

(998).bit_length()

(99).bit_length()

(11).bit_length()

3. Jalankan perintah representasi dari nilai i = 555:

bin(i)

hex(i)

oct(i)

Laporan Resmi:

1. Buatlah jawaban dari **Tugas Pendahuluan** .
2. Buatlah summary dari **Percobaan & Latihan** 1 sampai dengan 3.
3. Analisa hasil dari **Percobaan & Latihan** 1 sampai dengan 3.
4. Berikan kesimpulan dari praktikum 1.