

Pemrograman Python (Part 1)

1. Objek

Setiap objek memiliki **TIPE** tipe ini menentukan nilai yang bisa dimilikinya. Tipenya berupa skalar atau non skalar.

Scalar

objek yang tidak dapat dipecah-pecah (indivisible)

- **int** untuk merepresentasikan bilangan bulat, misalkan -123, 0, 28000, 1000500, etc.
- **float** untuk merepresentasikan bilangan riil, misalkan 0.43, 2005.34, 12.25, etc.
- **bool** untuk merepresentasikan nilai True atau False
- **string** untuk merepresentasikan karakter
- **none** untuk merepresentasikan ketiadaan nilai

Nonscalar

objek dengan struktur internal. Contoh: lists, tuples, strings, instances of classes etc.

```
In [1]: # Kita dapat melakukan cek ke tipe dari suatu data menggunakan fungsi type()
a=23
type(23)

b='Ajaccio'
type(b)

c=234.56
type(c)
```

```
Out[1]: float
```

Objek dan operator dapat dikombinasikan untuk menghasilkan ekspresi.

2. Operator

1. Operator standar +, -, *
2. Operator sisa hasil bagi %
3. Operator perpangkatan **
4. Operator pembandingan

- ==(sama dengan),

- $!=$ (tidak sama dengan),
- $<$ (kurang dari),
- $>$ (lebih besar dari),
- $<=$ (kurang dari atau sama dengan),
- $>=$ (lebih dari atau sama dengan).

Ingat! tidak ada operator $<>$ di PYTHON!

2.1. Operator Pembagian

Terdapat dua operator untuk pembagian yakni $(/)$ dan $(//)$.
1. Pembagian Bilangan Bulat - sisa pembagian tidak dianggap. Contoh :

- $16//2$ akan menghasilkan 8
- $17//7$ akan menghasilkan 2

2. Pembagian Bilangan Real Contoh :

- $16/2$ akan menghasilkan 8
- $17/7$ akan menghasilkan 2.4285714285714284

2.2 Operator Inkremental

Operator untuk melakukan penjumlahan $(+=)$, pengurangan $(-=)$, pembagian $(/=)$, perkalian $(*=)$ secara inkremental. Contoh :

- $A = +1$ sama artinya dengan $A = A + 1$
- $umur = -2$ sama artinya dengan $umur = umur - 2$

2.3. Operator Modulo (Sisa Hasil Bagi)

$$a \bmod b \equiv c$$

Di dalam perhitungan pecahan dengan hasil yang tidak bulat kita akan menemukan sisa pembagian.
 Misalkan 31 roti dibagi tiga orang- masing-masing akan mendapatkan 9 roti dan sisa 4 roti : $\frac{31}{3} = 9\frac{4}{3}$

NOTASI MATEMATIS	BAHASA_INDONESIA	NOTASI PYTHON
$31 \bmod 3 \equiv 4$	31 dibagi 3 sisa 4	$31\%3$
$16 \bmod 2 \equiv 0$	16 dibagi 2 sisa 0	$16\%2$
$100 \bmod 37 \equiv 26$	100 dibagi 37 sisa 26	$100\%37$

In [2]: `###----TESTING`

Command/Statement - menginstruksikan interpreter untuk melakukan sesuatu.

In [3]:

```
# Sekumpulan command/statement
print ('Why did the British tea maker deliver the tea packages himself even though he was sick?')
print ('He was trying to fulfill his due-tea.')
print ('=====')
print ('If a British person is too relaxed during tea time', 'they can get injured or die.')
print ('One should avoid a casual -tea as much as possible.')
# interpreter akan menghasilkan output sebagai berikut.
```

Why did the British tea maker deliver the tea packages himself even though he was sick?
He was trying to fulfill his due-tea.

=====

If a British person is too relaxed during tea time they can get injured or die.

One should avoid a casual -tea as much as possible.

3. Variabel dan Penugasan

Variable merupakan suatu lokasi dimana kita menyimpan data.

In [4]:

```
#-----Deklarasi Variabel pada Bahasa Pemrograman C#-----
#string name = "John";
#int myNum = 15;
#Console.WriteLine(name);
#Console.WriteLine(myNum);
```

C++: Can not compare float and int

Python:



Python tidak memiliki perintah tersendiri untuk mendeklarasikan variabel. Variabel tercipta di saat sebuah nilai ditugaskan pada variabel tersebut.

```
In [5]: #-----Variabel pada Python-----  
heyTayo = 'Ada bebek 10 di kali 2 jadi berapa?'  
heyToyo = 8  
heyTayo, heyToyo
```

```
Out[5]: ('Ada bebek 10 di kali 2 jadi berapa?', 8)
```

3.1. Aturan penulisan variabel :

1. Nama suatu variabel harus dimulai dengan HURUF atau GARISBAWAH.
2. Nama variabel hanya boleh mengandung karakter ALFANUMERIK dan GARIS BAWAH

3. Nama variabel menganut kepekaan kapital (case sensitivity). </br> umurBapak, umuRbapak, UmurBapak, UMURBAPAK adalah empat variabel yang berbeda satu sama lain

Cara Penulisan Variabel

- **Camel Case** - semua suku kata menyatu, awalan kata-kata dimulai dengan huruf kapital kecuali kata pertama. </br> Contoh: alamatRumah, nomorKodePos, namaDepan </br> </br>
- **Pascal Case** - semua suku kata menyatu, awalan kata-kata dimulai dengan huruf kapital kecuali kata pertama.</br> Contoh: AlamatRumah, NomorKodePos, NamaDepan </br> </br>
- **Snake Case** - suku kata dipisahkan dengan garis bawah, kata disusun dengan huruf kecil semua atau hurufbesar semua.</br> Contoh: alamat_rumah, NOMOR_KODE_POS, nama_depan

3.2. Melakukan Perbaruan atas Nilai Variable

```
In [6]: jarakJktSby = 800
        id(jarakJktSby)
```

```
Out[6]: 2143612061808
```

```
In [7]: jarakJktSby = 812
        id(jarakJktSby)
```

```
Out[7]: 2143612061520
```

```
In [8]: jarakJktSby
```

```
Out[8]: 812
```

3.3. Multi Asssignment

```
In [9]: umurAyah, umurIbu, umurKakak, umurSaya, umurAdik=40, 38, 16, 14, 12
```

```
In [10]: umurKakak
```

```
Out[10]: 16
```

3.4. Konversi Tipe

Konversi Implisit

Python melakukan konversi data dari satu tipe ke tipe lain secara otomatis.

```
In [11]: num_int = 123
        num_flo = 1.23
        num_new = num_int + num_flo

        print("Tipe Data num_int:", type(num_int))
        print("Tipe Data of num_flo:", type(num_flo))
        print("Tipe Data setelah penjumlahan:", type(num_new))
```

```
Tipe Data num_int: <class 'int'>
Tipe Data of num_flo: <class 'float'>
Tipe Data setelah penjumlahan: <class 'float'>
```

Konversi Eksplisit (Typecasting)

User mengubah konversi tipe data dari tipe data awal ke tipe data yang diinginkan. Untuk melakukan hal ini, dengan fungsi awal `int()`, `float()`, `str()`.

```
In [12]: # int variable
var_char1 = '5'
print ('Tipe data saat ini', type(var_char1))

# typecast to float
var_float1 = float(var_char1)
print ('Tipe setelah konversi', type(var_float1))

# typecast to int
var_int1 = float(var_float1)
print ('Tipe setelah konversi', type(var_int1))
```

```
Tipe data saat ini <class 'str'>
Tipe setelah konversi <class 'float'>
Tipe setelah konversi <class 'float'>
```

Meminta User Memasukkan Masukan

```
In [ ]: hey=input()
```

4. Komentar dan Docstring

Komentar di python dapat dibuat dengan mengawali suatu baris dengan tanda tagar (**#**). Komentar dapat dibuat untuk membuat kode lebih dapat dibaca dan juga dapat digunakan sebagai alat bantu untuk mengeksekusi dalam melakukan pengetesan kode.

Petik tiga (triple quotes) sesuai dengan Python documentation merupakan `docstrings` dan tidak dianggap komentar. Apapun didalam petik tiga akan dibaca oleh interpreter. Python's triple quotes comes to the rescue by allowing strings to span multiple lines, including verbatim NEWLINEs, TABs, and any other special characters. Similarly, single triple quotes can also be used for the same purpose as shown below:

```
In [ ]: # KOMENTAR

# KOMENTAR INI DIBUAT DALAM
# BEBERAPA BARIS
# JADI KALAU MAU BUAT BEBERAPA BARIS, TAGARNYA DI SETIAP BARIS YA

a= """Ini adalah suatu
docstring yang digunakan untuk memfasilitasi
adanya string di beberapa baris. Ingat, ini bukan komentar.
Di sini, kita bisa juga masukkan spesial karakter seperti % $ & tanpa masalah"""
print(a)
```

5. Operasi String

5.1 Slicing String

```
In [ ]: word1='MISANTROPHE'
        len(word1)
```

```
In [ ]: print('Dua Huruf Pertama-----', word1[0:2])
        print('Tiga Huruf Pertama-----', word1[:3])
        print('Selain Tiga Huruf Pertama-----', word1[:-3])
        print('Huruf Ketiga sampai Huruf Kelima-----', word1[2:5])
```

```
In [ ]: 'U' in word1
```

```
In [ ]: word1.find('OP')
```

5.2 Format Operators

The format operator (%)

```
In [ ]: psrt_dklat= 40
        print('Ada %d peserta di diklat ini.' %psrt_dklat)
```

6. Pengandaian

```
In [ ]: angka=int(input("Kuy masukkan angkamu: "))
```

```
In [ ]: if angka%2==1:
        print("Angka",angka,"adalah angka ganjil")
        print("Terimakasih, kamu sudah selesai periksa angka.",
              "\n Mohon maaf kalau angkamu tidak ganjil keluarannya gak ada ya!")
```

```
In [ ]: if angka%2==1:
        print("Angka",angka,"adalah angka genap")
        else:
        print("Angka",angka,"adalah angka genap")
```

=====Notes=====

1. Pernyataan setelah `if` , `elif` atau `else` harus diakhiri dengan tanda titik dua (:)
2. Dalam satu paket `if` - `elif` - `else` , susunannya selalu segaris. </br>Jika tidak segaris maka bukanlah struktur utama dari pengandaian tersebut.
3. Selalu terdapat **4 spasi indentasi** antara `if` - `elif` - `else` dengan konsekuensinya.

```
In [ ]: angka2=int(input("Kuy masukkan angka pertamamu: "))
```

```
In [ ]: angka3=int(input("Kuy masukkan angka kedumu: "))
```

```
In [ ]:
if angka2 > angka3:
    print(angka2, 'LEBIH BESAR DARI', angka3 )
elif angka2 < angka3:
    print(angka2, 'LEBIH KECIL DARI', angka3 )
else:
    print(angka2, 'SAMA DENGAN', angka3 )
```

" Jangan lupa bahwa operator = adalah untuk penugasan. Untuk perbandingan gunakan operator perbandingan == "

=====

TUGAS UNTUK ASYNCHRONOUS 1

=====

Tugas 1

Buatlah sebuah program dengan masukan berupa tiga bilangan bulat dari user. Kemudian program akan melakukan perhitungan dan menghasilkan keluaran berupa jam dan menit sisanya bila ada.

Contoh keluaran bila input dari user 121:

Hours</br> 2</br> Minutes</br> 1</br>

Tugas 2

Berikut adalah suatu string **str = 'X-DSPAM-Confidence:0.8475'**</br> Gunakan 'find' dan 'string slicing' untuk mengekstrak porsi dari karakter setelah tanda kolon, kemudian gunakan ubahlah menjadi data float.

6.1 Pengandaian Singkat

KONSEKUENSI 1 if PENGANDAIAAN 1 else KONSEKUENSI 2

```
In [ ]:
kondisi = 'Gak Punya Uang'
print('Perut Kenyang') if kondisi =='Sudah Makan' else 'Masih Lapar'
```

7. Perulangan (Iterasi)

7.1. while Loops

```
In [ ]:
counter = 0
while (counter<5):
    counter = counter+1
    print('Ini adalah hasil iterasi ke-',counter)
```

7.2. for Loops

```
In [ ]:
kaunter=5
```



```
for kaunter in range(0,kaunter):  
    print('Ini adalah hasil iterasi ke-', kaunter)
```

7.3. Continue statement

Apabila ketemu dengan suatu syarat, maka nilai ekspresi pada syarat itu tidak di evaluasi, evaluasi dilanjutkan langsung ke nilai berikutnya.

```
In [ ]: for nilaiA in range(5):  
        if nilaiA == 3:  
            continue  
        print('Nilai saat ini', nilaiA)
```

7.4. Break Statement

Apabila ketemu dengan suatu syarat, maka evaluasi dihentikan.

```
In [ ]: for nilaiB in range(5):  
        if nilaiB == 3:  
            break  
        print('Nilai saat ini', nilaiB)  
  
print('Nilai terakhir hanya sampai', nilaiB)
```

7.5. Pass Statement

Apabila ketemu dengan suatu syarat, maka evaluasi dilanjutkan

```
In [ ]: for nilaiB in range(5):  
        if nilaiB == 3:  
            pass  
        print('Di sini ada pass')  
        print('Nilai saat ini', nilaiB)
```

8. Fungsi

8.1 Build in Function

Python memiliki fungsi-fungsi yang sudah didefinisikan sendiri oleh pembuat bahasa python. Seperti, `max()`, `round()`, `len()`, `sum()`.

```
In [ ]: print(max(123,8856, 21, 4, 756, 22))  
        print(len('Honorificabilitudinitatibus'))
```

8.2 User Defined Function

User dapat membuat sendiri fungsi di python sesuai dengan yang diinginkan. Untuk membuat fungsi kita menggunakan reserved word `def`. Bentuk umum dari UDF :

```
def NAMA_FUNGSI (PARAMETER_1,PARAMETER_2, ...,PARAMETER_N):  
    ALGORITMA_FUNGSI
```

```
In [ ]: def hitung_luas_lingkaran(jejari):  
        luas=22/7*jejari**2  
        return luas  
print ("Luas Lingkaran", hitung_luas_lingkaran2(33))
```

```
In [ ]: def hitung_luas_lingkaran2(jejari2, pi=3.14):  
        luas2=pi*jejari2  
        return luas2  
print ("Luas Lingkaran", hitung_luas_lingkaran2(5, pi=3.14))
```

```
In [ ]: def hitung_volume_balok(p,l,t):  
        volume=p*l*t  
        return volume  
print ("Volume Balok", hitung_luas_lingkaran2(5, pi=3.14))
```

Unpacking operator (*)

digunakan untuk menerima masukan parameter sebanyak mungkin

```
In [ ]: def rata_rata(*args):  
        tot = 0  
        for a in args:  
            tot += a  
        return tot/len(args)
```

```
In [ ]: print('Hasil rata-rata-1:', rata_rata(1,545,678,32,89,357,777,82,34,47))  
print('Hasil rata-rata-2:', rata_rata(23,55,45))  
print('Hasil rata-rata-3:', rata_rata(-11, -22, 89, 77))
```

8.3 Ekspresi Lambda

Python memfasilitasi pembuatan fungsi anonim, tanpa harus memberikan nama dengan `lambda`. Fungsi `lambda` hanya boleh mengandung satu ekspresi. Bentuk umum dari ekspresi `lambda` adalah sebagai berikut :

`lambda arguments: expression`

```
In [ ]: hrg_akhir= lambda hrg_jual, pdiskon, ongkir : hrg_jual-(hrj_jual*pdiskon)+ongkir  
print(hrg_akhir(50000,0.05,10000))
```

```
In [ ]: nama_singkat = lambda n_depan, n_tengah, n_belakang
```

Daftar Referensi :

=====

1. Bell, Anna. 2018. **Get Programming Learn to Code With Python**. Manning
2. Deitel, Paul and Deitel, Harvey. 2020. **Intro to Python for Computer Science and Data Science 3rd Edition**. Pearson Education

3. Gutttag, John V. 2021. **Introduction to Computation and Programming Using Python**. MIT Press
4. Shaw, Zed.A. 2018. **Learn More Python 3 The Hard Way**. Addison-Weasley
5. Severance, Charles. 2009. **Exploring Python 3**. Ann Arbor

=====

TUGAS UNTUK ASYNCHRONOUS 2

=====

Latihan 3:

Banyaknya hari dalam satu bulan bervariasi dari 28 hingga 31. Buatlah program yang memasukkan nama bulan dari user (string). Kemudian program harus melakukan perhitungan jumlah hari dalam bulan tersebut. Untuk bulan februari keluaran berupa string **"28 atau 29 hari"** for agar berlaku juga untuk bulan kabisat.

Latihan 4:

Skala Richter Magnitude Descriptor Less than 2.0 Micro 2.0 to less than 3.0 Very Minor 3.0 to less than 4.0 Minor 4.0 to less than 5.0 Light 5.0 to less than 6.0 Moderate 6.0 to less than 7.0 Strong 7.0 to less than 8.0 Major 8.0 to less than 10.0 Great 10.0 or more Meteoric

- Buatlah program yang membaca besarnya magnitude dari user dan memberikan keluaran terkait grade dari kekuatan gempa.
- Contoh keluaran : **'Gempa berskala 5,5 berkategori Moderat'**

Latihan 5:

- Buatlah script yang menerima input berupa lima digit angka dari user.
- Pisahkan angka menjadi digit individual. Print secara terpisah dengan separator berupa tiga spasi
- Contoh: Jika user memasukkan 42339, maka hasil print
4 2 3 3 9
- Asumsikan bahwa user selalu menginput angka dalam panjang yang tepat.
- Use both the floor division and remainder operations to "pick off " each digit.

Latihan 6:

Sebuah bilangan palindrom adalah angka yang jika dibaca dari depan kebelakang sama. Contoh: 12321, 55555, 45554 and 11611. Tulislah sebuah program yang membaca lima digit bilangan bulat untuk menentukan apakah mereka palindrome atau bukan. [Hint: Use the // and % operators to separate the number into its digits.]

Latihan 7 :

Konversi derajat celcius ke fahrenheit diatur oleh persamaan berikut : $F = \left(\frac{9}{5}\right) * C + 32$ Gunakan fungsi ini untuk memprint suatu daftar konversi temperatur dari celcius untuk range 0–100. Gunakan presisi 1 digit.

Latihan 8 :

Pak Joni memiliki rencana untuk membeli sebuah rumah, tetapi sayangnya dia tidak punya banyak uang. Dia berencana untuk membeli tetapi dengan cicilan ke bank saja. Namun, sayangnya lagi-lagi untuk membayar DP pinjaman bankpun ia tidak punya. Oleh karena itu, dia berencana untuk menabung setiap bulan agar bisa membayar DP. Tugas anda : tentukan berapa lama dia harus menabung agar dia bisa membayar cicilan rumah tersebut.

1. Mintalah pak joni untuk membuat harga perkiraan rumah yang dia inginkan simpan pada variable, `total_cost`.
2. Biasanya untuk DP rumah, bank akan mensyaratkan pembayaran DP sebesar 25%, simpan pada variabel `portion_down_payment = 0.25`.
3. Tanyakan pak joni berapa uang yang sudah dia punya selama ini yang dicadangkan untuk membayar DP, simpan pada variable `current_savings` nilai defaultnya boleh anda set 0.
4. Asumsikan bahwa pak joni menginvestasikan tabungan dengan bijak, sehingga memperoleh bunga 5 % per tahun.
5. Simpan gaji tahunan pak joni dalam variabel `annual_salary`.
6. Tanyakan pak Joni persen dari gaji pak joni yang dia sisihkan untuk menabung simpan pada variable `portion_saved`. Variabel ini berbentuk decimal (contoh 0.1 untuk persentase 10%).
7. Pada akhir tiap bulan bulan, tabungan anda akan meningkat sesuai dengan besarnya investasi ditambah dengan porsi gaji yang ditabungkan (`gaji_tahunan/ 12`).

Pengayaan

Write a program that converts a binary (base 2) number to decimal (base 10). Your program should begin by reading the binary number from the user as a string. Then it should compute the equivalent decimal number by processing each digit in the binary number. Finally, your program should display the equivalent decimal number with an appropriate message.