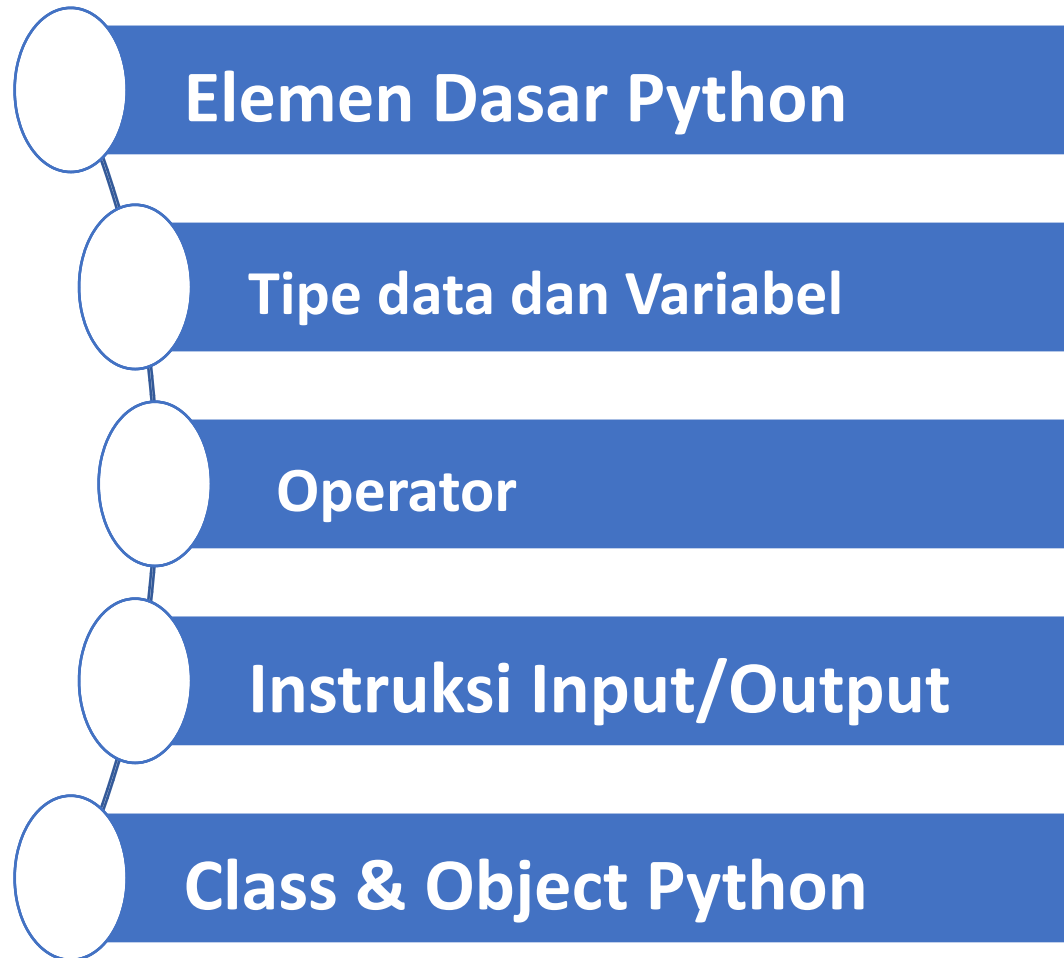




"Struktur Bahasa Python"

[@SUARGA | [Pertemuan 02]

OutLine





Elemen Dasar Bahasa Python

1. CharacterSet

#modul-2

2. Keywords

3. Tipe data dan Variabel

4. Operator

5. Instruksi Input-Output

6. Instruksi Seleksi

#modul-3

7. Instruksi Perulangan

8. Pembuatan Fungsi

9. Akses File



Character Set

Character set dari Python adalah standard ASCII yaitu dari kode 0 (NUL) hingga kode 127 (DEL) , namun Python bisa mengerti set karakter lain apabila dinyatakan terlebih dahulu misalnya kode utf-8 atau iso-8859-1. Cara menyatakan-nya sebagai berikut:

```
# -*- coding : utf-8 -*-
```

diawal coding program.

	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
1	LF	VT	FF	CR	SO	SI	DLE	DC1	DC2	DC3
2	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
3	RS	US	SP	!	"	#	\$	%	&	`
4	()	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[\]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{		}	~	DEL		

ASCII standard character (dalam desimal)

Keyword

- **Keyword** adalah perbendaharaan kata dari Python, dimana setiap kata merupakan suatu instruksi tertentu, oleh sebab itu pengguna hanya boleh memakainya sebagai instruksi, tidak untuk keperluan lain.
- **Keyword** pada dasarnya diperlukan pertama untuk mengetahui sintaks dari command, kedua untuk menjadi acuan ketika membuat variable agar tidak sama dengan keyword.

- **Keyword** dalam Python terdiri atas 35 kata, yaitu:

and	def	finally	in	or	try
as	del	for	is	pass	while
assert	elif	from	lambda	print	with
break	else	global	None	raise	yield
class	except	if	nonlocal	return	exec
continue	False		import	not	True

Keyword tdk boleh digunakan sebagai variabel

Variabel

- **Variable** adalah nama atau simbol yang mewakili suatu data/nilai. **Variabel** dalam Python harus diberi nama yang dimulai huruf, bukan angka. Sebagai contoh Nama dan Umur adalah variabel yang melekat pada seseorang, nilai-nya misalnya “Abdus Salam” dan “25 tahun”.
- Programmer Python biasanya memakai huruf kecil diawal nama variable atau huruf kecil pada kata pertama, dan dimulai huruf besar pada kata yang kedua.
- Perlu di-ingat bahwa Python membedakan nama dengan huruf kecil dan nama dengan huruf besar.
- Variabel juga tidak boleh menggunakan spasi.

- **Literal** adalah tetapan angka maupun string, misalnya: 10, 1.23, 9.2e-3, dan “ini adalah string”. Nilai literal adalah nilai yang diberikan kepada variabel.
- **Angka** dalam python bisa dibedakan empat macam, yaitu: **integer**, **long integer**, **floating point**, dan **complex number**, misalnya:
- Integer : 2, 4, 45, 127
- Long integer : 123456789654321L
#memakai huruf l atau L
- Floating point : 3.23, 50.78, 6.34E-4
- Complex number : (-5 + 4j) , (2.3 – 4.7j) #memakai huruf j atau J

- **Komentar** : baris-baris komentar harus diberi tanda # didepan tiap baris
- contoh: #program ini ditulis oleh Suarga
- **String** adalah untaian karakter. String dapat dibentuk menggunakan single-qoute seperti 'Quote me on this', atau memakai double-quote seperti "What's your name?". Disamping itu string yang terdiri atas beberapa baris dapat dibentuk memakai triple-quote ' ' ' atau " " ", contohnya sebagai berikut:

```

""" This is a multi-line string, and this is the first line
This is the second line
"What's your name?" I asked.
He said "Bond, James Bond."
"""

```

- Triple-quote juga dipakai untuk menyatakan komentar yang terdiri atas beberapa baris, dan komentar tersebut harus ditutup dengan triple-quote lagi.

“““ Berikut ini disajikan listing program sebagai dokumentasi dari program yang di tulis oleh Suarga.

“““

- Selanjutnya suatu string yang ingin ditampilkan dalam beberapa baris harus diberi tanda escape `\n` untuk memisahkan setiap baris, contoh:

```
>>> A = 'Tanda \n pada string membuat string  
dipisahkan \n beberapa baris'
```

```
>>> print(A)
```

Tanda

pada string membuat string dipisahkan
beberapa baris

```
>>>
```

Objek Python

Jenis Objek	Contoh literal
Angka (Numbers)	34215, 3.1415, 1.34E+02, 3+4j, 0b111, Decimal(), Fraction()
Strings	'Spam', "Bob's", b'a\x01c', u'sp\xc4m'
List	[1, [2, 'three'], 4, 5], list(range(10))
Dictionary	{'food' : 'spam', 'taste' : 'yum'}, dict(hours=10)
Tuple	(1, 'spam', 4, 'U'), tuple('spam')
File	Open('myfile.txt', 'a+'), open(r'C:\ham.bin','wb')
Set	Set('abc'), {'a', 'b', 'c'}
Tipe lainnya:	Boolean, types, None
Unit program	Function, Modules, Class

string ADT

- **str()** : mengubah integer ke string
- **int()** : mengubah string ke integer, `int(s,B)` dapat mengubah string `s` dalam bentuk sistem bilangan `B` menjadi integer, misalnya `int("10101",2)` memberikan nilai 21
- **chr()** : mengubah integer ke karakter, `chr(97)` adalah 'a'
- **ord()** : mengubah karakter ke nilai integer, `ord('A')` adalah 65
- **center()** : membuat string berada ditengah
- **ljust()** : membuat string rapat ke kiri
- **lright()** : membuat string rapat ke kanan

- **strip()** : membuang spasi dari dalam string
- **lstrip()** : membuang spasi di-depan string
- **rstrip()** : membuang spasi di-belakang string
- **count()** : menghitung frekuensi huruf dalam string
- **find()** : mencari suatu substring/karakter dalam string
- **startswith()** : memeriksa apakah huruf tertentu mengawali string
- **endswith()** : memeriksa apakah huruf tertentu meng-akhiri string
- **lower()** : mengubah string menjadi huruf kecil
- **upper()** : mengubah string menjadi huruf besar
- **isalpha()** : memeriksa string apakah abjad
- **isupper()** : memeriksa string apakah semuanya huruf besar
- **islower()** : memeriksa string apakah semuanya huruf kecil
- **isdigit()** : memeriksa string apakah semuanya angka

- Cara menggunakan atribut fungsi string ini adalah dengan menulis nama fungsi dibelakang objeknya, misalnya sebagai berikut:

```
>>> Nama = 'abdul rahman'    # definisi objek string
```

```
>>> Nama.isalpha()          # spasi bukan alpha?
```

```
False
```

```
>>> Nama.isupper()  # apa nama huruf besar?
```

```
False
```

```
>>> Nama.islower()  # apa nama huruf kecil?
```

```
True
```

```
>>> Nama.isdigit()   # apa nama adalah digit?
```

```
False
```

```
>>> Nama.startswith('a') # apa nama diawali huruf 'a'?
```

```
True
```

```
>>> Nama.endswith('a')  # apa nama diakhiri huruf 'a'?
```

```
False
```


Deklarasi Variabel

- Berbeda dengan bahasa program pada umumnya dimana tipe dari variable harus dinyatakan, **Python tidak memerlukan deklarasi tipe dari variable**, karena tipe variable akan disesuaikan secara otomatis dengan nilai data yang diberikan, misalnya:

```
n = 17    # n diberikan nilai integer 17
pi = 3.14159265 # pi diberikan nilai riil atau
              # floating point
message = 'ini adalah string' #message diberi
                               # nilai string
```

Operator

- Operator ***aritmetika*** dasar terdiri atas:
+ (tambah), - (kurang),
* (kali), / (bagi) dan
** (pangkat).

Contoh:

```
>>> a=5
```

```
>>> b=8
```

```
>>> c=2
```

```
>>> d=4
```

```
>>> e = a + b*c - d**c + a*b/c
```

```
>>> e
```

```
25.0
```

```
>>>
```

- Pada string operator $+$ berarti (concatenation) atau menyambung string, dan operator $*$ berarti (repeat) atau penggandaan, misalnya:

first = 'Selamat '

second = 'pagi'

maka:

first + second menghasilkan Selamat pagi.

third = first * 3, maka third akan sama dengan "SelamatSelamatSelamat".

- Python menggunakan tanda # untuk memberi komentar pada program, misalnya:

V = 5 # V diberikan nilai 5

v = 10 # laju partikel adalah 10

Python juga memanfaatkan escape letter seperti pada C dan Java, antara lain yang penting adalah:

\n - baris baru

\t - tabulasi

\' - single-coute dalam string, seperti 'What\'s your Name'

\” - double coute dalam string

Operator Aritmetik

+	plus	Menjumlah 2 objek	$3 + 5 \rightarrow 8$ 'a' + 'b' \rightarrow 'ab'
-	minus	Mengubah ke negative atau mengurangi dua objek	-5 ubah ke negative $50 - 24$ mengurangkan
*	multiply	Mengalikan dua objek	$5 * 8 \rightarrow 40$ 'la' * 3 \rightarrow 'lalala'
**	power	Memangkatkan bilangan	$3 ** 4 \rightarrow 81$
/	divide	Membagi dua bilangan	$4/3 \rightarrow 1$ $4.0/3 \rightarrow 1.33333$

Operator Aritmetik

//	Floor division	Pembulatan ke bawah hasil bagi	$4/3.0 \rightarrow 1.0$
%	modulo	Sisa pembagian bulat	$8\%3 \rightarrow 2$ $25.5 \% 2.25 \rightarrow 1.5$

Operator bit

<<	Left shift	Geser bit ke kiri	$2 \ll 2 \rightarrow 8$, karena 10 digeser 2 kali ke kiri menjadi 1000
>>	Right shift	Geser bit ke kanan	$11 \gg 1 \rightarrow 5$, karena 1011 geser kanan 1 bit menjadi 0101
&	Bitwise AND	Operasi bit dari AND	$5 \& 3 \rightarrow 1$
 	Bitwise OR	Operasi bit dari OR	$5 3 \rightarrow 7$
^	Bitwise XOR	Operasi bit dari XOR	$5 \wedge 3 \rightarrow 6$
~	Bitwise invert	$\sim x$ adalah $-(x+1)$	$\sim 5 \rightarrow -6$

Operator Relasi

<	Less than	Memeriksa apakah relasi lebih kecil terpenuhi	5 < 3 -> 0 (false) 3 < 5 -> 1 (true)
>	Greater than	Memeriksa apakah relasi lebih besar terpenuhi	5 > 3 -> 1 (true) 3 > 5 -> 0 (false)
<=	Less than or equal	Relasi lebih kecil atau sama	3 <= 6 -> 1 (true)
>=	Greater than or equal	Relasi lebih besar atau sama	4 >= 3 -> 1 (true) 3 >= 4 -> 0 (false)
==	Equal to	Memeriksa kesamaan	x=2; y=2, (x ==y) -> 1 'str' == 'stR' -> 0
!=	Not equal to	Memeriksa ketidaksamaan	2 != 3 -> 1 'str' != 'str' -> 0
Not	Boolean not	Membalikkan logik	
And	Boolean and	Relasi "DAN"	
Or	Boolean or	Relasi "ATAU"	

Indentation

- Seperti disinggung di awal tulisan bahwa **Python meringkas kelompok instruksi dengan memakai indentation**, sehingga sangat perlu suatu konsistensi dalam membentuk indentation ini. Indentation boleh dipilih salah satu dari: *tab*, *dua-spasi* kekanan, atau *empat-spasi* kekanan.
- Hanya perlu konsistensi. Perhatikan bahwa terkadang suatu error muncul hanya karena salah indentation. Apabila memakai “tab” maka seterusnya pakai tab, jangan dicampur dengan pemakaian spasi, demikian pula sebaliknya.

#contoh pemakaian indentation salah

a=5

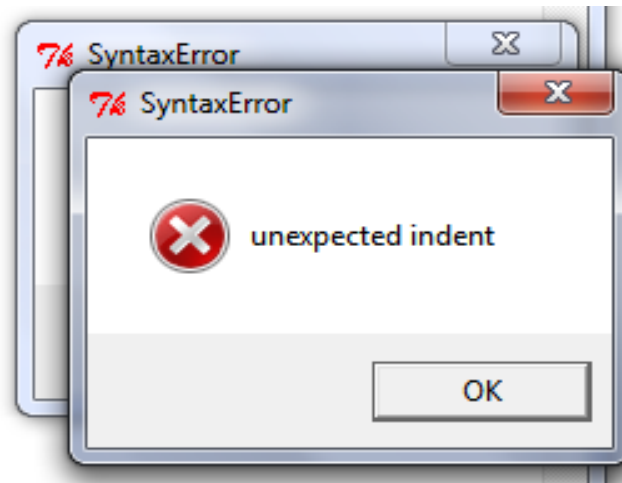
b=2

c=4

d=8

e = a + b*c - d

- Perhatikan bahwa baris-baris instruksi ini menggunakan indent yang tidak konsisten, maka ketika di-run akan memunculkan error,



Struktur Program Python

#komentar tentang program

```
import pustaka
```

#pemberian nilai awal

```
.. .. .
```

definisi fungsi

```
def fungsi-1:
```

```
    .. .. instruksi ..
```

```
..
```

```
def fungsi-2:
```

```
    .. .. instruksi ..
```

```
..
```

#program utama main()

```
def main():
```

```
    .. .. instruksi .. ..
```

#menjalankan fungsi main()

```
main()
```

Pustaka fungsi

- **string** pustaka untuk proses text string
- **textwrap** format text paragraph
- **re** regular expression
- **collections** container data types
- **array** pustaka untuk larik
- **heapq** heap sort
- **struct** binary data structures
- **time** jam, waktu
- **datetime** penanggalan
- **decimal** fixed and floating point
- **fractions** rational numbers
- **random** bilangan acak
- **math** fungsi matematika

Pustaka Fungsi

- **os** sistem operasi / path
- **glob** filename pattern matching
- **shutil** high level file operation
- **mmap** memory map files
- **codecs** string encoding and decoding
- **StringIO** text buffers with a file API
- **filecmp** compare files
- **pickle** object serialization
- **shelve** persistent storage of objects
- **anydbm** database

Lakukan “import” untuk menggunakan fungsi pustaka, misalnya:

```
import os      #untuk menggunakan fungsi OS
```

Instruksi Input

- Instruksi untuk menerima data bertipe character dari keyboard adalah :

var = input("keterangan : ");

contoh:

Nama = input('Masukkan nama anda : ')

umur = int(input('Masukkan umur-nya : '))

- bila tipe variabel input bukan char maka berikan tipe data-nya misalnya int.

Instruksi Output

- Instruksi untuk menampilkan sesuatu di monitor adalah:

```
print('keterangan', variabel);
```

```
print("nama : %s, umur:%d" % (nama,umur))
```

contoh:

```
x = int(input('Masukkan satu angka : '))
```

```
print('Angka yang dimasukkan = ', x)
```

Program Test operator

```
# testOperator.py
x=6
y=4
a=12.5
b=7

print('x = ', x, '; y = ', y)
print('x + y = ', x+y)
print('x - y = ', x-y)
print('x * y = ', x*y)
print('x / y = ', x/y)
print('x ** y = ', x**y)
print('x // y = ', x//y)
print('x % y = ', (x%y))
print()
print('a = ', a, '; b = ', b)
print('a > b is ', (a>b))
print('a < b is ', (a<b))
```


Hasil test operator

`x = 6 ; y = 4`

`x + y = 10`

`x - y = 2`

`x * y = 24`

`x / y = 1.5`

`x ** y = 1296`

`x // y = 1`

`x % y = 2`

`a = 12.5 ; b = 7`

`a > b is True`

`a < b is False`

Membuat program Python

1. Pikirkan logik/algoritma dari masalah yang akan di program
2. Aktifkan editor Python, kemudian edit program yang merupakan translasi algoritma ke Python
3. Rekam program, beri nama dgn ekstensi .py
4. Run program, periksa hasilnya

Contoh

- Buat program Python untuk konversi suhu Celcius ke suhu Fahrenheit
- Logika solusi:
 - masukkan suhu Celcius (C)
 - konversi dgn rumus: $F = 9/5 * C + 32$
 - Tampilkan nilai F

#C_to_F.py konversi Celcius ke Fahrenheit

```
def main():  
    C = float(input("Masukkan suhu Celcius: "))  
    F = 9.0/5.0 * C + 32.0  
    print("Untuk Suhu Celcius = ", C)  
    print("Fahrenheit = ", F)
```

main()

```
= RESTART: D:/USER/Documents/myLect  
at/MODUL_STRUKTURDATA/C_to_F.py  
Masukkan suhu Celcius: 25  
Untuk Suhu Celcius = 25.0  
Fahrenheit = 77.0
```

Beberapa Contoh Program

- [investasi.py](#) : data input: modal, bunga, jangka waktu (tahun), output: daftar investasi dari tahun awal ke tahun akhir
- [pisahdigit.py](#): data input, sebuah angka bulat, output: setiap digit-nya dipisah
- [ContohIO.py](#): data input: nama, alamat, umur, jenis kelamin, dan institusi, output: tampilan kembali dari data input

investasi.py

```
# investasi.py - menghitung bunga uang simpanan
modal = input('Masukkan simpanan awal : ')
modal = float(modal)
rate = 0.07
bunga = modal * rate
investasi = modal + bunga
print('Modal awal = %10.2f' % modal)
print('Bunga = %10.2f' % bunga)
print('Nilai investasi setelah 1 tahun = %10.2f' % investasi)
```

pisahdigit.py

```
#pisahdigit.py - memisah digit angka bulat
angka = int(input('Masukkan angka 4 digit : '))
d1 = angka//1000
sis = angka % 1000
d2 = sis//100
sis = sis % 100
d3 = sis //10
sis = sis % 10
d4 = sis
print('Digit pertama = ', d1)
print('Digit kedua   = ', d2)
print('Digit ketiga  = ', d3)
print('Digit keempat = ', d4)
```

ContohIO.py

```
#ContohIO.py
def main() :
    print("Input Data : ")
    nama = input("Nama : ")
    alamat = input("Alamat : ")
    umur = int(input("Umur : "))
    jenkel = input("Jenis kelamin: ")
    institusi = input("Masukkan institusi-nya : ")
    print()
    print("Output Data : ")
    print("Nama : ", nama)
    print("Alamat : ", alamat)
    thlahir = 2022 - umur
    print("Lahir tahun : " , thlahir)
    print("Seorang %s di institusi %s" % (jenkel, institusi))

main()
```


Instruksi Control

- Instruksi control terdiri atas:
- instruksi seleksi / pemilihan : **if/else**
- instruksi perulangan: **for dan while**
- Akan dibahas dalam modul-3

Preview Tipe data lanjut

- Python memiliki **struktur data internal** yang bersifat objek seperti array, string, list, tuple, dictionary, dan set.
- Secara rinci tipe data objek ini akan diuraikan pada Modul-5 nanti
- Pada modul ini hanya akan diperkenalkan saja sebagai tipe data lanjut.

list

- Tipe data “list” dimanfaatkan untuk menyimpan data seperti sebuah daftar.
- Contoh: daftar = [2, 3, 4, 5]
- tanda kurung [] menandakan objek list, sehingga daftar akan bertipe list dengan isi 2, 3, 4, dan 5.
- Karena bersifat objek maka list menyediakan member function dalam ADT-nya antara lain:

- **append(x)** : menambahkan satu item di ujung kanan dari list
- **clear()** : mengosongkan isi list
- **count(x)** : menghitung frekuensi x dalam list
- **copy()** : menyalin isi list
- **extend(L)** : meng-eksetensi suatu list dengan menambahkan L
- **insert(i,x)** : menyisipkan item x pada posisi i
- **remove(x)** : membuang item pertama bernilai x dari list
- **pop(i)** : membuang item yang berada pada posisi i,
- **pop()** : membuang item terakhir
- **index(x)** : menemukan index dari item x
- **sort()** : mengurutkan item dalam list
- **reverse()** : membalikkan urutan item

```
>>> daftar=[2, 3, 4, 5]
>>> daftar.append(6)
>>> daftar.append(7)
>>> daftar
[2, 3, 4, 5, 6, 7]
>>> daftar.insert(4,8)
>>> daftar
[2, 3, 4, 5, 8, 6, 7]
>>> daftar.pop()
7
>>> daftar
```

```
[2, 3, 4, 5, 8, 6]
>>> daftar.sort()
>>> daftar
[2, 3, 4, 5, 6, 8]
>>> daftar.reverse()
>>> daftar
[8, 6, 5, 4, 3, 2]
>>> daftar.remove(4)
>>> daftar
[8, 6, 5, 3, 2]
>>>
```

tuple

- hampir sama dengan list, tetapi tuple tidak memerlukan tanda kurung atau memakai tanda kurung biasa ()

```
>>> t = 12345, 54321, 'hello!'
```

```
>>> t[0]
```

```
12345
```

```
>>> t
```

```
(12345, 54321, 'hello!')
```

```
>>> u = t,(1, 2, 3, 4, 5)
```

```
>>> u
```

```
((12345, 54321, 'hello!'), (1, 2, 3, 4, 5))
```

```
>>>
```

dictionary (dict)

- dictionary merupakan daftar dimana setiap item memerlukan dua elemen, yaitu: key dan data
- dictionary memerlukan tanda kurung { } untuk menciptakannya.

```
>>> phone = {'Amir':3456, 'Bandu':7349, 'Chomas':5436}
>>> phone['Amir']
3456
>>> phone['Daud']=2645
>>> phone
{'Bandu': 7349, 'Chomas': 5436, 'Daud': 2645, 'Amir':
3456}
>>> phone.keys()
dict_keys(['Bandu', 'Chomas', 'Daud', 'Amir'])
>>> del phone['Daud']
>>> phone
{'Bandu': 7349, 'Chomas': 5436, 'Amir': 3456}
>>>
```


Class / Object Python

- **Sifat-sifat Class:**
 - **attribut**
 - **method**
 - **constructor**
 - **encapsulation**
 - **inheritance**
 - **polymorphism**

- **Suatu class digunakan untuk mendefinisikan suatu model object**
- **Object diciptakan melalui constructor class**
- **sifat-object dinyatakan sebagai atribut didalam class**
- **kelakuan/fungsi object dinyatakan sebagai method di dalam class**
- **atribut dan method suatu class terlindung dari class lain (encapsulation)**
- **atribut dan method suatu class dapat diwariskan (inheritance) ke class lainnya**
- **beberapa method boleh memakai nama yang sama (polymorphism) asalkan atribut-nya berbeda**

membuat Class

Bentuk umum class:

```
class Nama-class:
    def __init__(self, data, ..):
        self.atribut = data
        ...
    def method(self, parameter):
        ...
```

contoh: class Circle

```
#kelas Circle, Circle.py
from math import pi, hypot
class myCircle:
    #constructor
    def __init__(self, x=0, y=0, radius=1):
        self.r = radius
        self.x = x
        self.y = y

    def keliling(self):
        return 2.0*pi*(self.r)
```

```

def luas(self):
    return pi*(self.r ** 2)

def pusat(self):
    print("Lingkaran pusat: (%.2f, %.2f)" % (self.x, self.y))
    print("Jari-jari : %.2f" % self.r)

def distFromOrigin(self):
    return hypot(self.x, self.y)

def main():
    C = myCircle(37.0, 43.0, 2.50)
    C.pusat()
    area = C.luas()
    kel   = C.keliling()
    jarak = C.distFromOrigin()
    print("Luasnya %f" % area)
    print("Keliling %f" % kel)
    print("Jarak lingkaran dari titik pusat-nya %f" %jarak)

main()

```

hasil RUN

```
>>> %Run Circle.py  
Lingkaran pusat: (37.00, 43.00)  
Jari-jari : 2.50  
Luasnya 19.634954  
Keliling 15.707963  
Jarak lingkaran dari titik pusat-nya 56.727418
```

Kuiz

1. Bahasa Python termasuk kategori Compiler atau Interpreter?
2. Sebutkan objek dasar dari Python beserta contoh-nya
3. Kalau mau meng-install Python kemana anda mencari installer-nya dan apa nama file installer-nya?
4. Berapa nilai dari ekspresi Python berikut ini:
 - a. $2 * (3 + 4) ** 2$
 - b. $2 * 3 + 4 ** 2$
 - c. $2 + 3 * 4 ** 2$
5. Berapa nilai ekspresi Python berikut ini:
 - a. $5 * 4 ** 2 / 2 - 4 * 5 ** 2$
 - b. $(5 * 4) ** 2 / (2 - 4 * 5) ** 2$
 - c. $(5 * 4 ** 2 / 2) - 4 * 5 ** 2$