

Dictionaries

Perkuliahan Algoritma Pemrograman pada Semester Ganjil 2020

Tipe data string dan list, index ditunjukkan dengan bilangan integer yang dimulai dari nol, misalkan

In [1]:

```
dataList=[5,8,1,4,2,3]
print('ukuran list=',len(dataList))
for i in range (len(dataList)):
    print('data List-',i,':',dataList[i])
```

```
ukuran list= 6
data List- 0 : 5
data List- 1 : 8
data List- 2 : 1
data List- 3 : 4
data List- 4 : 2
data List- 5 : 3
```

In []:

```
dataStr='Algoritma'
print('ukuran list=',len(dataStr))
for i in range (len(dataStr)):
    print('data -',i,':',dataStr[i])
```

Sedangkan tipe data **dictionaries** adalah tipe data sepertinya halnya list yang memiliki beberapa anggota (dengan tipe data yang berbeda-beda), hanya saja index dari dictionaries ini tidak hanya berupa integer, dapat berupa float, string, character, dan lain-lain, seperti contoh berikut:

In []:

```
dataDict={}
dataDict['apel']='apple'
dataDict['semangka']='watermelon'
dataDict['anggur']='grape'
dataDict['stroberi']='strawberry'
dataDict['nanas']='pineapple'
dataDict['pisang']='banana'
dataDict['mangga']='mango'
for fruit in dataDict:
    print(fruit, ' : ',dataDict[fruit])
print(dataDict)
```

In []:



```
data=[]  
data.append(5)
```

In []:



```
a={}
```

In []:



```
a['test']=7
```

In []:



```
print(dataDict)
```

Pada variabel *a* dengan tipe data **dictionaries** tersebut, dapat diperhatikan beberapa hal sebagai berikut :

- setiap anggota dipisahkan oleh koma
- setiap anggota terdiri dari index dan nilai dari anggota tersebut, dimana antara index dan nilai dipisahkan dengan :
- index disebut dengan **keys**, dan nilai disebut dengan **value**, sehingga anggota dari elemen tersebut disebut dengan **key value pairs**

Cara lain untuk mendefinisikan suatu variabel dengan tipe data dictionaries tersebut adalah :

In []:



```
b={'one':'satu','two':'dua','three':'tiga'}
```

In []:



```
print(b)
```

In [1]:



```
dataDict3={10:'apple', 55:'watermelon', 7 : 'grape', 0:'strawberry' }  
print (dataDict3)
```

```
{10: 'apple', 55: 'watermelon', 7: 'grape', 0: 'strawberry'}
```

Berikut adalah variabel bertipe data dictionaries, dengan index berupa integer.

In []:



```
data={}  
data[1]='satu'  
data[3]='empat'
```

Operasi Dictionaries

del

Untuk menghapus suatu **key value pairs** dari suatu dictionaries, dapat dilakukan dengan menggunakan operasi **del**. Misalkan terdapat kantin yang memiliki data stok jajanan, dengan key adalah jenis jajanan dan value yang terdapat pada masing-masing key tersebut adalah jumlah stok jajanan yang terdapat pada kantin, adalah sebagai berikut :

In [2]:



```
snacks={}
snacks['chocolate']=20
snacks['cracker']=15
snacks['candy']=38
snacks['biscuit']=5
```

In [3]:



```
print(snacks)
```

```
{'chocolate': 20, 'cracker': 15, 'candy': 38, 'biscuit': 5}
```

Misalkan mahasiswa membeli semua candy yang terdapat pada kantin, tersebut, maka admin dapat menghapus anggota snacks yang memiliki key candy, dengan perintah berikut

In [4]:



```
del snacks['candy']
```

In [5]:



```
print(snacks)
```

```
{'chocolate': 20, 'cracker': 15, 'biscuit': 5}
```

Atau, ketika admin menginginkan hanya jumlah stok yang habis, akan tetapi keberadaan key candy tetap dipertahankan, dapat dilakukan perintah sebagai berikut

In [6]:

```
snacks={}
snacks['chocolate']=20
snacks['cracker']=15
snacks['candy']=38
snacks['biscuit']=5
print (snacks)
snacks['candy']=0
print(snacks)
```

```
{'chocolate': 20, 'cracker': 15, 'candy': 38, 'biscuit': 5}
{'chocolate': 20, 'cracker': 15, 'candy': 0, 'biscuit': 5}
```

len

Seperti halnya tipe data list, perintah **len** pada dictionaries menghasilkan jumlah anggota yang terdapat pada dictionaries

In []:

```
snacks={}
snacks['chocolate']=20
snacks['cracker']=15
snacks['candy']=38
snacks['biscuit']=5
```

In []:

```
a=[5,6,3,2]
print(len(a))
b='surabaya'
print(len(b))
c=(6,7,1)
print(len(c))
```

Methods

Methods memiliki fungsi yang sama dengan **function**, yaitu suatu sub code dengan argumen dan nilai balik tertentu, hanya saja perintah untuk eksekusi **method** ini berbeda dengan **function**.

Misalkan terdapat suatu function **keys** dengan argumen berupa dictionaries, dan nilai balik adalah daftar **keys** yang terdapat pada dictionaries tersebut, maka perintah eksekusi function ini adalah :

keys(snacks)

Akan tetapi jika **keys** tersebut berupa method, maka eksekusi method tersebut adalah

snacks.keys()

Terdapat beberapa method yang terdapat pada dictionaries ini, antara lain:

- keys

- items
- values, dll

In [7]:



```
snacks.keys()
```

Out[7]:

```
dict_keys(['chocolate', 'cracker', 'candy', 'biscuit'])
```

In []:



```
snacks.
```

In [10]:



```
for i in snacks.keys():  
    print(i)
```

```
chocolate  
cracker  
candy  
biscuit
```

In [8]:



```
snacks.items()
```

Out[8]:

```
dict_items([('chocolate', 20), ('cracker', 15), ('candy', 0), ('biscuit',  
5)])
```

In [9]:



```
snacks.values()
```

Out[9]:

```
dict_values([20, 15, 0, 5])
```

In []:



```
snacks.get('candy2',0)
```

Aliasing, Copying

Jika terdapat dua buah variabel yang menunjuk pada obyek yang sama, maka perubahan terhadap satu variabel, akan berakibat perubahan pada variabel yang lainnya.

Misalkan terdapat suatu list aList, dan bList sebagai berikut

In [11]:



```
aList=[5,8,1,4,2,3]
print(aList)
```

[5, 8, 1, 4, 2, 3]

In [12]:



```
bList=aList
print(bList)
```

[5, 8, 1, 4, 2, 3]

In [13]:



```
bList[3]=100
print(bList)
```

[5, 8, 1, 100, 2, 3]

In []:



```
print(aList)
```

In []:



```
aList[5]=1000
print(aList)
```

In []:



```
print(bList)
```

Oleh karena itu, jika ingin memiliki dua buah variabel dengan data yang sama (inisialisasi), dan menginginkan agar perubahan terhadap suatu variabel tidak merubah data dari variabel yang lain, gunakanlah method **copy**. Hal ini dapat dilakukan pada tipe data list maupun dictionary.

In [14]:



```
aList=[5,8,1,4,2,3]
bList=aList.copy()
print(aList)
print(bList)
```

[5, 8, 1, 4, 2, 3]
[5, 8, 1, 4, 2, 3]

In [15]:



```
bList[3]=100
print(bList)
```

[5, 8, 1, 100, 2, 3]

In []:



```
print(aList)
```

In []:



```
aList[0]=300
print(aList)
```

In []:



```
print(bList)
```

Misalkan terdapat variabel *fruit* dengan tipe data **dictionaries**, yang berisi nama buah dan warnanya, sebagai berikut:

In []:



```
fruit={}
fruit['apple']='red'
fruit['pineapple']='yellow'
fruit['cucumber']='green'
fruit['orange']='orange'
```

In []:



```
fruit
```

In []:



```
alias=fruit
alias
```

In []:



```
alias['apple']='pink'
print(alias)
print(fruit)
alias['grape']='purple'
print(alias)
print(fruit)
```

In []:



```
fruit['apple']='red'  
print(fruit)  
print(alias)
```

In []:



```
alias=fruit.copy()  
print(fruit)  
print(alias)
```

In []:



```
alias['strawberry']='red'
```

In []:



```
print(fruit)  
print(alias)
```

In []:



```
fruit['apple']='green'  
print(fruit)  
print(alias)
```