



KONSEP PEMROGRAMAN

[Python 3.6.2]

[Abstract](#)

Data Structure Lanjut (List)

Puji Winar Cahyo
github.com/pwcahyo

A. Tujuan Pembelajaran.

Mahasiswa diharapkan dapat :

- 1) Memahami konsep list
- 2) Dapat melakukan implementasi list

B. Pengantar

Pembahasan pada module sebelumnya telah menjelaskan apa saja yang dimaksud dengan immutable object dan mutable object. Keduanya memiliki sifat tersendiri, sehingga penggunaan object tersebut perlu penyesuaian. Seperti contoh apabila object tersebut memang akan dinamis (dapat dilakukan pergantian) maka lebih tepat kita menggunakan mutable object. Berbeda dengan object yang tidak memerlukan pergantian maka bisa menggunakan immutable object.

Selain tuple dan set, data structure yang sering digunakan diantara lain ada list dan dictionary, keduanya merupakan mutable object yang tentunya memiliki sifat fleksibilitas dalam editing. Pembahasan dan Implementasi list dan dictionary dapat dilihat pada bagian C.

C. Implementasi

1. list

List merupakan data structure yang tersedia dalam bahasa pemrograman python. Data di dalam list dapat berbentuk immutable object. seperti : string, number maupun None. Isi data di dalam list tersebut juga tidak harus beraturan, dalam 1 list bisa berisi beberapa macam variasi immutable object. Tidak hanya itu, di dalam list bisa terdiri beberapa duplicate data, berbeda dengan set yang data didalamnya harus menggunakan format secara unique. Method yang tersedia untuk pemrosesan list dapat dijabarkan sebagai berikut.

a) mencetak data list

data list dapat dicetak menggunakan pemanggilan index list, seperti contoh berikut mencetak data 4, yang berada pada index list ke 3.

```
>>> list = [1,2,3,4,5]
>>> print(list[3])
```

mencetak isi list menggunakan perulangan

```
>>> list = [1,2,3,4,5]
>>> for x in list:
>>>     print(x)
```

b. `list.count(element)`

menghitung banyaknya elemen sejenis yang telah didefinisikan, dengan contoh mencari banyaknya data dengan nilai 5.

```
>>> list = [1,1,3,1,5]
>>> list.count(1)
>>> 3
```

c. `list.index(element)`

mencari index element data 5. Dengan hasil berada pada index ke-4.

```
>>> list = [1,2,3,4,5]
>>> list.index(5)
>>> 4
```

mencari index element data 1. Dengan pencarian dimulai dari index ke-2.

```
>>> list = [1,1,3,4,1]
>>> list.index(1, 2)
>>> 4
```

d. `len(list)`

mengetahui jumlah panjang atau banyaknya isi list.

```
>>> list = [1,2,3,4,5]
>>> len(list)
>>> 5
```

e. `membership`

list dapat dicari nilai keanggotaannya, seperti contoh apakah 3 termasuk dalam anggota list.

```
>>> list = [1,2,3,4,5]
>>> member = 3 in list
>>> print(member)
True
```

f. not membership

not membership adalah kebalikan dari membership, yaitu mencari nilai bukan anggota dari list, seperti contoh berikut, apakah 3 bukan anggota list.

```
>>> list = [1,2,3,4,5]
>>> member = 3 not in list
>>> print(member)
False
```

g. max & min

mencari nilai maksimal didalam list.

```
>>> list = [1,2,3,4,5]
>>> print(max(list))
```

mencari nilai minimal didalam list.

```
>>> list = [1,2,3,4,5]
>>> print(min(list))
```

h. penambahan data

✓ concatenation list

penambahan data list dapat dilakukan seperti concatenation pada set, seperti contoh menambahkan data 6 pada tuple [1,2,3,4,5].

```
>>> list = [1,2,3,4,5]
>>> list+= [6,]
```

✓ penambahan data list pada index tertentu

penambahan data 6 pada index ke-2 dapat dilihat seperti contoh berikut.

```
>>> list = [1,2,3,4,5]
>>> list.insert(2,6)
>>> print(list)
[1,2,6,3,4,5]
```

✓ penambahan data pada list di index paling akhir

seperti contoh melakukan penambahan data 7 pada index paling akhir, atau dibelakang index ke-4 dapat menggunakan list.append(data)

```
>>> list = [1,2,3,4,5]
>>> list.append(7)
>>> print(list)
[1,2,3,4,5,6,7]
```

i. penggabungan data

- ✓ penggabungan dua list atau lebih.

```
>>> list1 = [1,2,3,4,5]
>>> list2 = [5,6,7,8,9]
>>> list3 = list1 + list2
>>> print(list3)
```

- ✓ penggabungan dua list atau lebih dengan melakukan perubahan list sebelumnya.

```
>>> list1 = [1,2,3,4,5]
>>> list2 = [5,6,7,8,9]
>>> list1.extend(list2)
>>> print(list1)
[1,2,3,4,5,6,7,8,9]
>>> list1.extend("slow")
>>> print(list1)
[1,2,3,4,5,6,7,8,9,'s','l','o','w']
```

j. convert data tuple kedalam variable

data list dapat dimasukan kedalam variable python, dengan contoh memasukan data 2 kedalam variable y.

```
>>> x,y,z = [1,2,3]
>>> print(y)
```

k. merubah data didalam list

isi dari list dapat dilakukan perubahan, seperti contoh **melakukan perubahan nilai 2 menjadi 0 pada index ke-1**, kemudian **merubah nilai 3 menjadi 6 pada index ke [2][0]**.

```
>>> list = [1, 2, [3, 4, 5]]
>>> list[1] = 0
>>> print(list)
[1, 0, [3, 4, 5]]
>>> list[2][0] = 6
>>> print(list)
[1, 0, [6, 4, 5]]
```

l. repetition

nilai didalam list dapat dilakukan operasi repetition (duplikasi berulang). Seperti contoh string "Hae PWCahyo" diulang menjadi 5 kali

```
>>> list = ["Hae PWCahyo"]
>>> repetition = list*5
>>> print(repetition)
```

m. join string list

nilai string list dapat digabungkan menggunakan fungsi join. Dapat dilihat seperti contoh berikut.

```
>>> tuple = ["saya","makan","nasi"]
>>> join_list = " ".join(tuple)
>>> print(join_list)
```

n. convert list ke tuple

data structure berbentuk list dapat diconvert kedalam tuple. Seperti contoh berikut. List [1,2,3,4,5] dilakukan convert kedalam tuple

```
>>> list = [1,2,3,4,5]
>>> tuple = tuple(list)
>>> print(tuple)
```

o. operator perbandingan

list dapat dilakukan perbandingan (>) (<) (<=) (>=) (!=) (==) sehingga akan menghasilkan nilai boolean. Sebagai contoh berikut melakukan perbandingan sama dengan.

```
>>> list1 = [1,2,3,4,5]
>>> list2 = [1,2,3,4,5]
>>> print(list == list2)
```

p. Menghapus element/data list

nilai didalam list dapat dihapus dengan menggunakan fungsi list.remove(data). dengan menggunakan list.remove(data) tersebut maka data akan dihapus berdasarkan index paling awal Seperti contoh berikut, yaitu menghapus nilai 3.

```
>>> list = [1,3,2,3,4,5,3]
>>> list.remove(3)
>>> print(list)
[1,2,3,4,5,3]
```

q. Mengambil element/data list

Selain menggunakan pengambilan data dengan menggunakan list[index] dengan prinsip data dipanggil akan tetapi data akan tetap exist berada didalam list. Sebaliknya apabila menggunakan list.pop() data pada index paling akhir akan diambil kemudian data dan index tersebut menjadi tidak ada. Apabila index pop diisi maka akan mengambil data dengan index yang telah didefinisikan. Seperti contoh list.pop(2) berarti akan mengambil data pada index ke-2, kemudian data dibelakangnya (pada yang dimulai pada index ke-3 dan seterusnya) akan maju mengisi index ke-2.

```
>>> list = [1,2,3,4,5]
>>> list.pop()
5
>>> print(list)
[1,2,3,4,5]
>>> a = list.pop(2)
>>> print(a)
3
>>> print(list)
[1,2,4,5]
```

r. Pengurutan data list

Data list dapat diurutkan dengan menggunakan fungsi sorted(list, reverse=False), reverse merupakan pengurutan data berdasarkan descending. Data yang diurutkan harus seragam, berbentuk string atau number kesemuanya. Seperti contoh berikut.

```
>>> list = [1,4,3,5,2]
>>> sorted(list)
[1,2,3,4,5]
>>> sorted(list, reverse=True)
[5,4,3,2,1]
```

s. Copy list

List dapat dilakukan copy, sehingga akan menciptakan object baru dengan nilai yang sama, `list.copy()` berbeda dengan `list_new = list`. `list.copy` akan menciptakan object baru dengan data dan alamat memory yang baru (independent/berdiri sendiri), sedangkan `list_new = list` hanya akan melakukan copy data akan tetapi tetap akan merujuk pada alamat memory yang sama. seperti

```
>>> list = [1,4,3,5,2]
>>> id(list)
4333302344
>>> list_new = list
>>> id(list_new)
4333302344
>>> list_copy = list.copy()
>>> id(list_copy)
4333302356
>>> print(list)
[1,4,3,5,2]
>>> print(list_new)
[1,4,3,5,2]
>>> print(list_copy)
[1,4,3,5,2]
```

t. Menghapus semua data list

Isi didalam list dapat dihapus secara keseluruhan dengan menggunakan perintah `list.clear()`

```
>>> list = [1,4,3,5,2]
>>> list.clear()
>>> print(list)
[]
```


Latihan

1. Pada saat libur akhir semester William, Kate, Anderson, Jame dan Mady berencana pergi berlibur ke yogyakarta. Mereka berencana membeli tiket kereta dari jakarta ke yogyakarta pada saat memesan ternyata mereka mendapat tempat duduk sebagai berikut.

- 1) Nomor pertama : William
- 2) Nomor kedua : Kate
- 3) Nomor ketiga : Anderson
- 4) Nomor keempat : Jame
- 5) Nomor kelima : Mady

Rute perjalanan dari Jakarta ke Yogyakarta :

Jakarta → Ciamis → Kebumen → Yogyakarta

Soal dikerjakan denganurut menggunakan list python:

- a. Buatlah satu buah object list untuk menampung keseluruhan orang yang ingin berlibur tersebut, sesuaikan tempat duduk dengan penempatan data dan index data. (Buat List)
- b. Sebelum berangkat ternyata ada orang baru yang ingin ikut berlibur ke yogyakarta orang tersebut bernama “Jones”. Tambahkan Jones pada posisi tempat duduk terakhir (nomor 6). (Tambahkan pada list)
- c. Ternyata sampai Ciamis Anderson sangat ingin bertemu keluarganya, sehingga dia memutuskan untuk tidak melanjutkan liburannya ke Yogyakarta, untuk itu Anderson memberikan tempat duduknya ke pada temannya dari Ciamis yang ingin ikut berlibur ke Yogyakarta. Nama teman Anderson adalah Grace. Tempatkan Grace pada nomor tempat duduk Anderson. (Ubah Data List nama Anderson dengan Grace)
- d. Sampai Kebumen ada pemeriksaan tiket untuk 6 orang penumpang tersebut, petugas memeriksa apakah ada orang yang bernama “Thompson” dari ke enam orang tersebut ? (Check Menggunakan fungsi Membership list adakah “Thompson”) (True/False)
- e. Setelah dari kebumen ternyata tempat duduk kereta harusurut sesuai abjad nama orang. (Urutkan list sesuai nama orang)
- f. 6 Orang tersebut telah sampai di yogyakarta. Tampilkan siapa saja 6 orang tersebut menggunakan perulangan (for/while)
- g. Kereta dari Yogyakarta kembali lagi ke jakarta setelah ke enam orang tersebut turun. (kosongkan list secara keseluruhan)