



Operasi pada List, Set, dan String

Pada modul berikut, Anda akan belajar tentang contoh-contoh operasi pada *list*, *set*, dan *string*. Beberapa contoh operasi tidak dapat sekaligus dilakukan pada list, set, dan string. Hal ini karena perbedaan karakteristik dari masing-masing tipe data. Penjelasan mengenai apa itu list, set, dan string sudah dijelaskan di Modul Tipe Data. Anda bisa mengulang kembali modul tersebut untuk lebih memahaminya ya.

len()

Sebelumnya, Anda sudah mengetahui bahwa slicing digunakan untuk urutan. Salah satu fungsi yang paling bermanfaat untuk List atau String adalah *len()* yang akan menghitung panjang atau banyaknya elemen dari List (untuk String menjadi menghitung jumlah karakternya).



```
1. contoh_list = [1, 3, 3, 5, 5, 5, 7, 7, 9]
2. print(contoh_list)
3. print(len(contoh_list))
4.
5. contoh_set = set([1, 3, 3, 5, 5, 5, 7, 7, 9])
6. print(contoh_set)
7. print(len(contoh_set))
8.
9. contoh_string = "Belajar Python"
10. print(contoh_string)
11. print(len(contoh_string))
```



1. **Output:**

[1, 3, 3, 5, 5, 5, 7, 7, 9]

9

{1, 3, 5, 7, 9}

5

Belajar Python

14



DIBANTU



min() dan max()

Selain menghitung panjang atau banyaknya elemen, Anda juga dapat mengetahui berapa nilai minimum dan maksimum dari suatu list menggunakan fungsi min() dan max(). Berikut contohnya.

```
1. angka = [13, 7, 24, 5, 96, 84, 71, 11, 38]
2. print(min(angka))
3. print(max(angka))
```

Output:

```
5
96
```

Count

Untuk mengetahui berapa kali suatu objek muncul dalam list, Anda dapat menggunakan fungsi count().

```
1. genap = [2, 4, 4, 6, 6, 6, 8, 10, 10]
2. print(genap.count(6))
3. string = "Belajar Python di Dicoding sangat menyenangkan"
4. substring = "a"
5. print(string.count(substring))
```

Output:

```
3
6
```

Penggabungan dan Replikasi

Pada List juga dimungkinkan adanya penggabungan (+) dan replikasi (*).



DIBANTU



```
2. huruf = ['P', 'Y', 'T', 'H', 'O', 'N']
3. gabung = angka + huruf
4. print(gabung)
```

Output:

```
[2, 4, 6, 8, 'P', 'Y', 'T', 'H', 'O', 'N']
```

Contoh replikasi:

```
1. learn = ['P', 'Y', 'T', 'H', 'O', 'N']
2. replikasi = learn * 2
3. print(replikasi)
```

Output:

```
['P', 'Y', 'T', 'H', 'O', 'N', 'P', 'Y', 'T', 'H', 'O', 'N']
```

Fungsi pengali juga dapat Anda manfaatkan untuk inisialisasi List.

```
1. tujuh = [7]*7
2. print(len(tujuh))
3. print(tujuh)
```

Output:

```
7

[7, 7, 7, 7, 7, 7, 7]
```

Range

Fungsi `range()` memberikan deret bilangan dengan pola tertentu. Untuk melakukan perulangan (misalnya `for`) dalam mengakses elemen list, Anda dapat menggunakan fungsi `range()` pada Python. Lebih detail mengenai operasi perulangan akan dibahas pada modul Perulangan dan Kontrol Perulangan.



DIBANTU



1. Range dengan 1 parameter n: membuat deret bilangan yang dimulai dari 0, sebanyak n bilangan.

```
1. for i in range(5):  
2.     print(i)
```

Output:

```
0  
  
1  
  
2  
  
3  
  
4
```

2. Range dengan 2 parameter n,p: membuat deret bilangan yang dimulai dari n, hingga sebelum p (bilangan p tidak ikut). Sering disebut sebagai inklusif n (deret dimulai bilangan n) dan eksklusif p (deret tidak menyertakan bilangan p).

```
1. for i in range(1, 11):  
2.     print(i)
```

Output:





2

3

4

5

6

7

8

9

10

3. Range dengan 3 parameter n,p,q: membuat deret bilangan yang dimulai dari n, hingga sebelum p (bilangan p tidak ikut), dengan setiap elemennya memiliki selisih q.

```
1. print([_ for _ in range(0,20,5)])
```

Output:

```
[0, 5, 10, 15]
```

in dan not in

Untuk mengetahui sebuah nilai atau objek ada dalam list, Anda dapat menggunakan operator *in* dan *not in*. Fungsi ini akan mengembalikan nilai boolean *True* atau *False*. Contohnya adalah sebagai berikut:

```
1. kalimat = "Belajar Python di Dicoding sangat menyenangkan"
2. print('Dicoding' in kalimat)
3. print('tidak' in kalimat)
4. print('Dicoding' not in kalimat)
5. print('tidak' not in kalimat)
```

**DIBANTU**



True

False

False

True

Memberikan nilai untuk multiple variable

Anda kadang memiliki beberapa variabel dan perlu memberikan nilai pada variabel-variabel tersebut. Secara konvensional, Anda bisa menandai variabel-variabel dengan nilai yang Anda inginkan, satu per satu. Python memiliki cara yang lebih praktis! Anda dapat memberikan nilai untuk beberapa variabel sekaligus dari elemen *List* atau *Tuple* tanpa perlu menandai satu per satu seperti:

```
1. data = ['shirt', 'white', 'L']
2. apparel = data[0]
3. color = data[1]
4. size = data[2]
```

Anda dapat melakukannya dengan cara:

```
1. data = ['shirt', 'white', 'L'] # From List
2. apparel, color, size = data
3. data = ('shirt', 'white', 'L') # From Tuple
4. apparel, color, size = data
```

Tentu saja, jumlah variabel dan jumlah item pada *List/Tuple* harus sama. Jika tidak, Python akan memberikan *ValueError*. Bagian ini hanya memberikan gambaran tentang operasi pada *List/Tuple*, Anda akan mempelajari lebih jauh tentang kesalahan (Error) di modul Penanganan Kesalahan.

```
1. data = ['shirt', 'white', 'L'] # From List
2. apparel, color, size, price = data
```

Output:



DIBANTU



```
<ipython-input-60-de8a162c18fd> in <module>
```

```
1 list = ['shirt', 'white', 'L'] # From List
```

```
----> 2 apparel, color, size, price = list
```

```
ValueError: not enough values to unpack (expected 4, got 3)
```

Catatan : Penggunaan assignment pada multi variabel ini dapat Anda gunakan untuk menukar nilai dua atau lebih variabel:

```
1. apparel, color = 'shirt', 'white'
2. apparel, color = color, apparel
3. print(apparel)
4. print(color)
```

Output:

```
white
```

```
shirt
```

Sort

Jika Anda ingin mengurutkan angka atau urutan huruf, maka Anda bisa menggunakan metode `sort()`. Metode `sort()` ini bisa mengurutkan List. Perhatikan contoh di bawah:

Contoh:

```
1. angka = [100, 1000, 500, 200, 5]
2. angka.sort()
3. print(angka)
```

Output:

```
[5, 100, 200, 500, 1000]
```

Contoh lain:



DIBANTU



```
3. print(kendaraan)
```

Output:

```
['helikopter', 'mobil', 'motor', 'pesawat']
```

Anda juga dapat memasukkan keyword `reverse=True` di dalam parameter untuk menjadikan urutannya terbalik.

```
1. kendaraan = ['motor', 'mobil', 'helikopter', 'pesawat']
2. kendaraan.sort(reverse=True)
3. print(kendaraan)
```

Output:

```
['pesawat', 'motor', 'mobil', 'helikopter']
```

Beberapa hal yang perlu Anda ketahui:

1. Metode `sort` langsung melakukan pengurutan pada variabel yang dioperasikannya, sehingga Anda tidak perlu melakukan operasi assignment (=).
2. Metode `sort` tidak dapat mengurutkan list yang memiliki angka dan string sekaligus di dalamnya.

Contoh:

```
1. urutan = ['Dicoding', 1, 2, 'Indonesia', 3]
2. urutan.sort()
3. print(urutan)
```

Output:

Anda akan mengalami error seperti berikut:

Traceback (most recent call last):

File "<pyshell#102>", line 1, in <module>

urutan.sort()

TypeError: '<' not supported between instances of 'int' and 'str'

Bagian ini hanya memberikan gambaran tentang operasi `sort`, Anda akan mempelajari lebih jauh tentang kesalahan (Error) di modul Penanganan Kesalahan.



DIBANTU



Contoh:

```
1. kendaraan = ['motor', 'mobil', 'helikopter', 'Pesawat']  
2. kendaraan.sort()  
3. print(kendaraan)
```

Output:

```
['Pesawat', 'helikopter', 'mobil', 'motor']
```

Untuk mengatasi kendala ini, Anda dapat memasukkan keyword **str.lower** pada parameter. Hal ini akan membuat metode sort menganggap semua objek menggunakan huruf kecil, tanpa mengubah kondisi asli dari objek tersebut.

Contoh:

```
1. kendaraan = ['Motor', 'Mobil', 'helikopter', 'pesawat']  
2. kendaraan.sort(key=str.lower)  
3. print(kendaraan)
```

Output:

```
['helikopter', 'Mobil', 'Motor', 'pesawat']
```



DIBANTU



Berikut ini surat dari CEO kami Narenda Wicaksono:

[Surat CEO Dicoding](#)

[< Sebelumnya](#)

[Selanjutnya >](#)



Dicoding Space
Jl. Batik Kumeli No.50, Sukaluyu,
Kec. Cibeunying Kaler, Kota Bandung
Jawa Barat 40123



Decode Ideas
Discover Potential

[Tentang Kami](#)

[Blog](#)
[Reward](#)
[Showcase](#)
[Hubungi Kami](#)
[FAQ](#)

Penghargaan

