

# Definisi Dictionary



Dictionary adalah collection yang tidak runtut, dapat diubah, dan berindeks serta tidak mengijinkan adanya member ganda.

## Dictionary Python

Dictionary Python berbeda dengan List ataupun Tuple.

Setiap urutanya berisi key dan value.

Setiap key dipisahkan dari value-nya oleh titik dua (:), item dipisahkan oleh koma, dan semuanya tertutup dalam kurung kurawal.

Dictionary kosong tanpa barang ditulis hanya dengan dua kurung kurawal, seperti ini: {}.

Nilai kamus bisa berupa tipe apa pun, namun key harus berupa tipe data yang tidak berubah seperti string, angka, atau tupel.

# Akses Nilai Dalam Dictionary Python

Untuk mengakses elemen Dictionary, Anda dapat menggunakan tanda kurung siku yang sudah dikenal bersama dengan key untuk mendapatkan nilainya. Berikut adalah contoh sederhananya :

```
dict = {'Nama': 'Adir', 'Umur': 20, 'Asal': 'Bangkalan'}
print ("dict['Nama']: ", dict['Nama'])
print ("dict['Umur']: ", dict['Umur'])
print ("dict['Asal']: ", dict['Asal'])
```

```
[Running] python -u "c:\Us
dict['Nama']: Adir
dict['Umur']: 20
dict['Asal']: Bangkalan
```

Activate Windows  
Go to Settings to activate Windows.

## Update Nilai Dalam Dictionary Python

Dictionary dapat update dengan menambahkan entri baru atau pasangan nilai kunci, memodifikasi entri yang ada, atau menghapus entri yang ada seperti ditunjukkan pada contoh sederhana yang diberikan di bawah ini.

```
dict = {'Nama': 'Adir', 'Umur': 20, 'Asal': 'Bangkalan'}
dict['Umur'] = 22 # Mengubah entri yang sudah ada
dict['Kuliah'] = "UTM" # Menambah entri baru
print ("dict['Nama']: ", dict['Nama'])
print ("dict['Umur']: ", dict['Umur'])
print ("dict['Asal']: ", dict['Asal'])
print ("dict['Kuliah']: ", dict['Kuliah'])
print(dict)
```

```
[Running] python -u "c:\Users\Ydpm\Documents\Py\Dictionary.py"
dict['Nama']: Adir
dict['Umur']: 22
dict['Asal']: Bangkalan
dict['Kuliah']: UTM
{'Nama': 'Adir', 'Umur': 22, 'Asal': 'Bangkalan', 'Kuliah': 'UTM'}
```

## Hapus Elemen Dictionary Python

Dictionary terdapat fungsi untuk menghapus elemen individual atau menghapus keseluruhan isi Dictionary. Anda juga dapat menghapus seluruh Dictionary dalam satu operasi.

```
dict = {'Nama': 'Adir', 'Umur': '20', 'Asal': 'Bangkalan'}
print(dict)
del dict['Nama'] # hapus entri dengan key 'Name'
print(dict)
dict.clear()     # hapus semua entri di dict
print(dict)
del dict         # hapus dictionary yang sudah ada
print(dict)
```

```
[Running] python -u "c:\Users\Ydpn\Documents\Py\Dictionary.py"
{'Nama': 'Adir', 'Umur': '20', 'Asal': 'Bangkalan'}
{'Umur': '20', 'Asal': 'Bangkalan'}
{}
<class 'dict'>
```

Activate Windows  
Go to Settings to activate Windows.

## Fungsi Build-in Pada Dictionary Python

Fungsi Python	Penjelasan
cmp(dict1, dict2)	Membandingkan unsur keduanya.
len(dict)	Memberikan panjang total Dictionary. Ini sama dengan jumlah item dalam Dictionary.
str(dict)	Menghasilkan representasi string yang dapat dicetak dari Dictionary
type(variable)	Mengembalikan tipe variabel yang lulus. Jika variabel yang dilewatkan adalah Dictionary, maka akan mengembalikan tipe Dictionary.

## Method Build-in Pada Dictionary Python

Method Python	Penjelasan
<code>dict.clear()</code>	Menghapus semua elemen Dictionary
<code>dict.copy()</code>	Mengembalikan salinan Dictionary
<code>dict.fromkeys()</code>	Buat Dictionary baru dengan kunci dari seq dan nilai yang disetel ke nilai.
<code>dict.get(key, default=None)</code>	For key, nilai pengembalian atau default jika tombol tidak ada dalam Dictionary
<code>dict.has_key(key)</code>	Mengembalikan true jika key dalam Dictionary, false sebaliknya
<code>dict.items()</code>	Mengembalikan daftar dari pasangan tuple dictionary (key, value)
<code>dict.keys()</code>	Mengembalikan daftar key dictionary
<code>dict.setdefault(key, default=None)</code>	Mirip dengan get (), tapi akan mengatur <code>dict [key] = default</code> jika kunci belum ada di dict
<code>dict.update(dict2)</code>	Menambahkan pasangan kunci kata kunci dict2 ke dict
<code>dict.values()</code>	Mengembalikan daftar nilai dictionary

## Himpunan (set)

- Set adalah salah satu tipe data di Python yang tidak berurut (*unordered*).
- Set memiliki anggota yang unik (tidak ada duplikasi). Jadi misalnya kalau kita meletakkan dua anggota yang sama di dalam set, maka otomatis set akan menghilangkan yang salah satunya.
- Set bisa digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan lain – lain.

## Membuat Set

Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal { }, dipisahkan menggunakan tanda koma.

Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi **set()**

Set bisa berisi data campuran, baik integer, float, string, dan lain sebagainya. Akan tetapi set tidak bisa berisi list, set, dan dictionary.



## Contoh

```
1 # set integer |
2 my_set = {1,2,3}
3 print(my_set)
4
5 # set dengan menggunakan fungsi set()
6 my_set = set([1,2,3])
7 print(my_set)
8
9 # set data campuran
10 my_set = {1, 2.0, "Python", (3,4,5)}
11 print(my_set)
12
13 # bila kita mengisi duplikasi, set akan menghilangkan salah satu
14 # output: {1,2,3}
15 my_set = {1,2,2,3,3,3}
16 print(my_set)
17
18 # set tidak bisa berisi anggota list
19 # contoh berikut akan muncul error TypeError
20 my_set = {1,2,[3,4,5]}
```

```
{1, 2, 3}
{1, 2, 3}
Traceback (most recent call last):
  File "c:\Users\Ydpm\Documents\Py\set.py", line 20, in <module>
    {1, 2.0, (3, 4, 5), 'Python'}
    {1, 2, 3}
    my_set = {1,2,[3,4,5]}
TypeError: unhashable type: 'list'
```

Activate Windows  
Go to Settings to activate Windows.

## Membuat Set

Untuk membuat set kosong, kita tidak bisa menggunakan { }, karena itu akan dianggap sebagai dictionary. Kita harus menggunakan fungsi set() tanpa argumen untuk membuat set kosong.

```
# membuat variabel a dengan {}
a = {}
print(type(a))

# harus menggunakan fungsi set()
a = set()
print(type(a))
```

```
<class 'dict'>
<class 'set'>
```

Activate Windows

# Mengubah Anggota Set

- Set bersifat *mutable*. Tapi, karena set adalah tipe data tidak berurut (unordered), maka kita tidak bisa menggunakan indeks. Set tidak mendukung indeks ataupun slicing.
- Untuk menambah satu anggota ke dalam set, kita bisa menggunakan fungsi `add()`,
- untuk menambahkan beberapa anggota sekaligus kita bisa menggunakan fungsi `update()`.
- List, tuple, maupun string bisa digunakan sebagai masukan dari fungsi `update()`.

Activate Windows  
Go to Settings to activate Windows

## Contoh

```
1 # membuat set baru
2 my_set = {1,2,3}
3 print(my_set)
4
5 # bila kita hilangkan tanda # dari baris 6 akan muncul error TypeError
6 #my_set[0]
7
8 # menambah satu anggota
9 # output: {1,2,3,4}
10 my_set.add(4)
11 print(my_set)
12
13 # menambah beberapa anggota |
14 # set akan menghilangkan duplikasi
15 # output: {1,2,3,4,5,6}
16 my_set.update([3,4,5,6])
17 print(my_set)
```

```
[Running] python -u
{1, 2, 3}
{1, 2, 3, 4}
{1, 2, 3, 4, 5, 6}
```

Activate Windows  
Go to Settings to activate Windows

## Menghapus Anggota Set

- Kita bisa menghapus anggota set dengan menggunakan fungsi `discard()` dan `remove()`.
- Perbedaanannya, fungsi `discard()` tidak akan memunculkan error bila anggota yang ingin dihapus ternyata tidak ada di dalam set, sedangkan `remove()` sebaliknya.

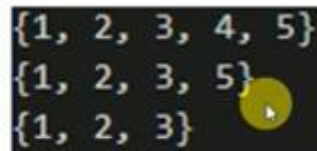
## Contoh

```
# membuat set baru |
my_set = {1, 2, 3, 4, 5}
print(my_set)

# menghapus 4 dengan discard
# output: {1, 2, 3, 5}
my_set.discard(4)
print(my_set)

# menghapus 5 dengan remove
# output : {1, 2, 3}
my_set.remove(5)
print(my_set)

# anggota yang mau dihapus tidak ada dalam set
# discard tidak akan memunculkan error
# output: {1, 2, 3}
my_set.discard(6)
```



```
{1, 2, 3, 4, 5}
{1, 2, 3, 5}
{1, 2, 3}
```

## Menghapus Anggota Set

Selain `discard()` dan `remove()`, kita bisa menghapus anggota set dengan menggunakan fungsi `pop()`. Dengan menggunakan fungsi `pop()`, kita menghapus salah satu anggota secara acak (random).

Untuk mengosongkan atau menghapus seluruh anggota set, kita bisa menggunakan fungsi `clear()`.

## Contoh

```
1 # membuat set baru
2 # output: set berisi anggota yang unik
3 my_set = set("HelloPython")
4 print(my_set)
5
6 # pop anggota
7 # output: anggota acak
8 print(my_set.pop())
9
10 # pop anggota lainnya
11 # output: anggota acak
12 print(my_set.pop())
13
14 # mengosongkan set
15 # output: set()
16 my_set.clear()
17 print(my_set) |
```

```
{'P', 'y', 'e', 't', 'n', 'H', 'h', 'o', 'l'}
P
y
set()
```

Activate Windows  
Go to Settings to activate Windows.

40

## Operasi Set di Python

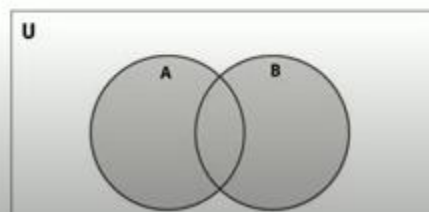
Set dapat digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan komplement.

Mari kita ambil dua contoh set berikut:

```
>>> A = {1, 2, 3, 4, 5}
>>> B = {4, 5, 6, 7, 8}
```

## Operasi Gabungan (Union)

- Gabungan (union) dari A dan B adalah himpunan atau set anggota yang ada di A dan B.
- Gabungan dapat dibuat dengan menggunakan operator palang (|). Selain itu juga bisa dilakukan dengan menggunakan fungsi union().



Activate Windows

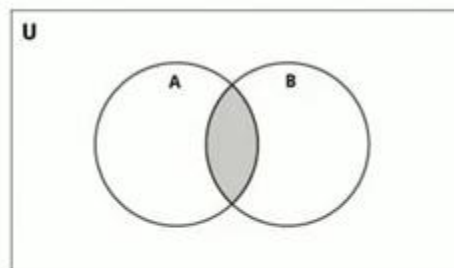


## Contoh

```
1 # Membuat set A and B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Gabungan menggunakan operator |
6 # output: {1, 2, 3, 4, 5, 6, 7, 8}
7 print(A | B)
8
9 # Menggunakan fungsi union()
10 # output: {1, 2, 3, 4, 5, 6, 7, 8}
11 A.union(B)
12
13 # output: {1, 2, 3, 4, 5, 6, 7, 8}
14 B.union(A)
```

```
[Running] python -u "c:\
{1, 2, 3, 4, 5, 6, 7, 8}
```

## Operasi Irisan (Intersection)



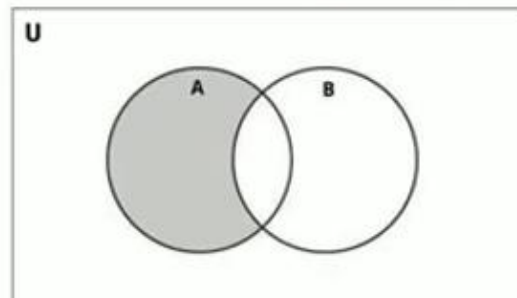
- Irisan (intersection) dari A dan B adalah himpunan atau set anggota yang sama di A dan B.
- Irisan dilakukan dengan menggunakan operator jangkar (&). Irisan juga bisa dilakukan dengan menggunakan fungsi intersection().

## Contoh

```
1 # Membuat set A and B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Irisan menggunakan operator &
6 # output: {4,5}
7 print(A & B)
8 # Menggunakan fungsi intersection()
9 # output: {4,5}
10 A.intersection(B)
11
12 # output: {4,5}
13 B.intersection(A)
```

```
[Running] python
{4, 5}
```

## Operasi Selisih (Difference)



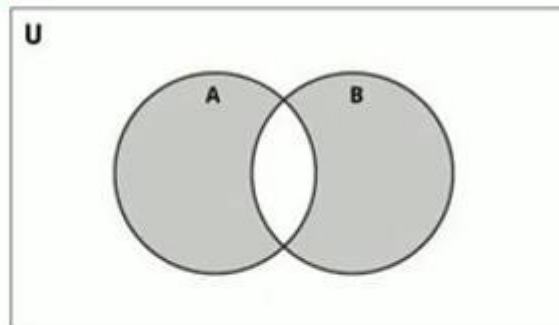
- Selisih (intersection) dari A dan B adalah himpunan atau set anggota yang hanya ada di A dan tidak ada di B. Begitu juga sebaliknya, ada di B tapi tidak ada di A.
- Selisih dilakukan dengan menggunakan operator kurang (-). Bisa juga dengan menggunakan fungsi difference().

## Contoh

```
1 # membuat A and B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Menggunakan operator - pada A
6 # Output: {1, 2, 3}
7 print(A - B)
8
9 # Output: {1, 2, 3}
10 A.difference(B)
11
12 # Menggunakan operator - pada B
13 # Output: {8, 6, 7}
14 print(B - A)
15
16 # Output: {8, 6, 7}
17 B.difference(A)
```

```
[Running] python
{1, 2, 3}
{8, 6, 7}
```

## Operasi Komplemen (Symmetric Difference)



- Operasi komplemen (symmetric difference) dari A dan B adalah himpunan atau set anggota yang ada di A dan di B, tapi tidak di keduanya.
- Komplemen dilakukan dengan menggunakan operator  $\wedge$ . Bisa juga dengan menggunakan fungsi `symmetric_difference()`.

Activate Windows  
Go to Settings to act

## Contoh

```
1 # membuat A and B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Menggunakan operator ^ pada A
6 # Output: {1, 2, 3, 6, 7, 8}
7 print(A ^ B)
8
9 # Output: {1, 2, 3, 6, 7, 8}
10 A.symmetric_difference(B)
11
12 # Menggunakan operator ^ pada B
13 # Output: {1, 2, 3, 6, 7, 8}
14 print(B ^ A)
15
16 # Output: {1, 2, 3, 6, 7, 8}
17 B.symmetric_difference(A)
```

```
[Running] python -u
{1, 2, 3, 6, 7, 8}
{1, 2, 3, 6, 7, 8}
```

## Metode (Fungsi) Set

Metode	Deskripsi
<code>add()</code>	Menambahkan satu anggota ke set
<code>clear()</code>	Menghapus semua anggota set
<code>copy()</code>	Mengembalikan shallow copy dari set
<code>difference()</code>	Mengembalikan set baru berisi selisih dua atau lebih set
<code>difference_update()</code>	Menghapus semua anggota set lain yang ada di set ini
<code>discard()</code>	Menghapus satu anggota dari set
<code>intersection()</code>	Mengembalikan set baru berisi irisan antara dua atau lebih set
<code>intersection_update()</code>	Mengupdate set dengan irisan set bersangkutan dan set lainnya
<code>isdisjoint()</code>	Mengembalikan True jika dua set tidak memiliki irisan
<code>issubset()</code>	Mengembalikan True jika set lain berisi set ini
<code>issuperset()</code>	Mengembalikan True jika set ini berisi set lain
<code>pop()</code>	Menghapus dan mengembalikan anggota acak dari sebuah set
<code>remove()</code>	Menghapus satu anggota dari set
<code>symmetric_difference()</code>	Mengembalikan set baru berisi komplemen dari dua set
<code>symmetric_difference_update()</code>	Mengupdate set dengan komplemen dari set ini dan set lainnya
<code>union()</code>	Mengembalikan set baru berisi gabungan dua atau lebih set
<code>update()</code>	Mengupdate set dengan gabungan set ini dan set lainnya



