



Python Programming

Tipe Data: Set dan Dictionary



01

Data Type: Set

Tipe Data: Set

- Set digunakan untuk menyimpan beberapa item dalam suatu variable. Bersifat *immutable*, *unordered*, *unindexed* dan tidak ada duplikasi.
- Walaupun *immutable* berbeda dengan *tuple*, pada set kita masih bisa menambah dan menghapus item
- Set dibuat dengan tanda kurung kurawal atau *curly bracket* “{ }”
- Karena bersifat *unindexed* maka kita tidak bisa mengakses item di set menggunakan index
- Set bersifat *unordered* dimana setiap item tidak bisa memiliki urutan yang ditentukan, setiap item dapat muncul dengan urutan yang berbeda

Tipe Data: Set

- Set bersifat *unordered*, jadi setiap item tidak selalu berurutan sesuai dengan saat pertama dibuat

```
1 a = {"apel", "jeruk", "ceri", "durian"}  
2  
3 print(a)
```

```
{'ceri', 'jeruk', 'apel', 'durian'}
```

- Set tidak bisa memiliki item yg sama

```
1 a = {"apel", "jeruk", "ceri", "durian", "apel"}  
2  
3 print(a)
```

```
{'apel', 'jeruk', 'durian', 'ceri'}
```

Tipe Data: Set

- Untuk melihat jumlah item dalam set dapat menggunakan fungsi len()

```
1 a = {"apel", "jeruk", "ceri", "durian"}  
2  
3 print(len(a))  
4
```

- Set juga bisa terdiri dari beberapa tipe data yang berbeda

```
1 a = {"apel", 20, True, "durian", 8.7}  
2  
3 print(a)  
  
{True, 'durian', 20, 8.7, 'apel'}
```

Tipe Data: Set

- Karena set tidak memiliki index, maka untuk mengakses item dalam set harus menggunakan loop atau if

```
1 setA = {"apel", "jeruk", "ceri", "durian"}  
2  
3 for item in setA:  
4     print(item)
```

```
apel  
jeruk  
durian  
ceri
```

```
1 setA = {"apel", "jeruk", "ceri", "durian"}  
2  
3 print("jeruk" in setA)  
4 print("pisang" in setA)
```

```
True  
False
```

Tipe Data: Set

- Setelah set dibuat, item tidak bisa diganti tetapi dapat ditambah dan dihapus
- Menambah item di set menggunakan add() method

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 a.add("semangka")
3
4 print(a)
```

{'durian', 'apel', 'semangka', 'jeruk', 'ceri'}

- Menambah item di set dari set lainnya

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = {"strawberry", "blueberry"}
3
4 a.update(b)
5
6 print(a)
```

{'jeruk', 'durian', 'blueberry', 'strawberry', 'apel', 'ceri'}

Tipe Data: Set

- Menambahkan dua buah set dengan union() method

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = {"strawberry", "blueberry", "apel"}
3
4 c = a.union(b)
5
6 print(c)
```

```
{'jeruk', 'durian', 'blueberry', 'strawberry', 'apel', 'ceri'}
```

- Berbeda dengan update() bila menggunakan union(), diperlukan deklarasi set baru untuk menyimpan kedua set yang telah ditambahkan
- Baik union() maupun update() tidak akan menyimpan item yang sama

Tipe Data: set

- Hanya menyimpan item yang sama dari dua buah set menggunakan metode `intersection_update()`

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = {"strawberry", "blueberry", "apel", "jeruk"}
3
4 a.intersection_update(b)
5
6 print(a)
```

`{'apel', 'jeruk'}`

- Hanya menyimpan item yang sama dari dua buah set menggunakan metode `intersection()` (harus disimpan dalam set yang baru)

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = {"strawberry", "blueberry", "apel", "jeruk"}
3
4 c = a.intersection(b)
5
6 print(c)
```

`{'apel', 'jeruk'}`

Type Data: set

- Hanya menyimpan item yang berbeda dari dua buah set menggunakan metode `symmetric_difference_update()`

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = {"strawberry", "blueberry", "apel", "jeruk"}
3
4 a.symmetric_difference_update(b)
5
6 print(a)
```

```
{'durian', 'blueberry', 'strawberry', 'ceri'}
```

- Hanya menyimpan item yang berbeda dari dua buah set menggunakan metode `symmetric_difference()` (harus disimpan dalam set yang baru)

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = {"strawberry", "blueberry", "apel", "jeruk"}
3
4 c = a.symmetric_difference(b)
5
6 print(c)
```

```
{'durian', 'blueberry', 'strawberry', 'ceri'}
```

Tipe Data: Set

- Menambah item di *set* dari sebuah *list*

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = ["strawberry", "blueberry"]
3
4 a.update(b)
5
6 print(a)
```

{'durian', 'apel', 'jeruk', 'blueberry', 'strawberry', 'ceri'}

- Menambah item di *set* dari sebuah *tuple*

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2 b = ("strawberry", "blueberry")
3
4 a.update(b)
5
6 print(a)
```

{'durian', 'apel', 'jeruk', 'blueberry', 'strawberry', 'ceri'}

Type Data: Set

- Menghapus item paling terakhir di set dengan `pop()` *method*

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2
3 a.pop()
4
5 print(a)
```

`{'jeruk', 'durian', 'ceri'}`

- Karena sifat *set* adalah *unordered* maka kita tidak akan bisa tahu item mana yang terhapus apabila menggunakan metode `pop()`
- Menghapus seluruh item secara sekaligus pada *set*

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2
3 a.clear()
4
5 print(a)
```

`set()`

Type Data: Set

- Menghapus item paling terakhir di set dengan `pop()` *method*

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2
3 a.pop()
4
5 print(a)
```

{'jeruk', 'durian', 'ceri'}

- Karena sifat *set* adalah *unordered* maka kita tidak akan bisa tahu item mana yang terhapus apabila menggunakan metode `pop()`
- Menghapus seluruh item secara sekaligus pada *set*

```
1 a = {"apel", "jeruk", "ceri", "durian"}
2
3 a.clear()
4
5 print(a)
```

set()

Quiz

1. Manakah diantara pilihan berikut yang merupakan sifat set di Python?
 - a. Mutable
 - b. Allows Duplicate
 - c. Ordered
 - d. Different data types not allowed
 - e. Unindexed
2. Manakah fungsi untuk menambahkan dua buah set hanya yang berbeda tanpa membuat variable baru?
 - a. `union()`
 - b. `intersection_update()`
 - c. `Intersection()`
 - d. `symmetric_difference_update()`
 - e. `symmetric_difference()`



03

Data Type: Dictionary

Type Data: Dictionary

- *Dictionary* digunakan untuk menyimpan data dengan format key: value
- Bersifat *ordered*, *changeable*, dan *do not allow duplicated*
- Untuk pembuatan *dictionary* diawali dengan kurung kurawal dan harus memiliki keys dan values
- Karena bersifat *ordered* (setelah Python versi 3.6) maka setiap keys dan values akan berurutan

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(kucing)
```

```
{'nama': 'Kuro', 'umur': 2, 'ras': 'Persian', 'jantan': True, 'hobi': ['makan',  
'tidur']}
```


Type Data: Dictionary

- Karena bersifat tidak bisa ada duplikat data, maka yang disimpan hanya value terakhir dari seluruh key yang sama. Namun, values bisa memiliki data yang sama

```
1 kucing = {  
2     "nama": "Takuro",  
3     "umur": 2,  
4     "nama": "Kuro",  
5     "ras": "Persian",  
6     "jantan": True,  
7     "lucu": True,  
8     "hobi": ["makan", "tidur"]  
9 }  
10  
11 print(kucing)
```

```
{'nama': 'Kuro', 'umur': 2, 'ras': 'Persian', 'jantan': True, 'lucu': True, 'hobi': ['makan', 'tidur']}
```

Type Data: Dictionary

- Untuk mengecek jumlah items dalam *dictionary* bisa menggunakan method `len()`

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(len(kucing))
```

5

- Membuat *dictionary* baru dengan method `dict()`

```
1 buah = dict(nama = "Apel", warna = "Merah", manis = True)  
2  
3 print(buah)
```

```
{'nama': 'Apel', 'warna': 'Merah', 'manis': True}
```

Type Data: Dictionary

- Mengakses item di *dictionary* dengan nama key

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(kucing['nama'])
```

Kuro

- Mengakses item di *dictionary* dengan method `get()`

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(kucing.get("ras"))
```

Persian

Type Data: Dictionary

- Mengambil seluruh *keys* dari *dictionary*

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(kucing.keys())
```

dict_keys(['nama', 'umur', 'ras', 'jantan', 'hobi'])

- Mengambil seluruh *values* dari *dictionary*

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(kucing.values())
```

dict_values(['Kuro', 2, 'Persian', True, ['makan', 'tidur']])

Type Data: Dictionary

- *Update values* dari sebuah *keys* dan menambahkan *key* di *dictionary*

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 print(kucing) # print dict sebelum update  
10  
11 kucing["umur"] = 1.5 # update value dari key "umur"  
12  
13 print(kucing) # print dict setelah update  
14  
15 kucing["lucu"] = True # tambah key "lucu" dan value-nya True  
16  
17 print(kucing) # print dict setelah ditambah key dan value-nya
```

{'nama': 'Kuro', 'umur': 2, 'ras': 'Persian', 'jantan': True, 'hobi': ['makan', 'tidur']}

{'nama': 'Kuro', 'umur': 1.5, 'ras': 'Persian', 'jantan': True, 'hobi': ['makan', 'tidur']}

{'nama': 'Kuro', 'umur': 1.5, 'ras': 'Persian', 'jantan': True, 'hobi': ['makan', 'tidur'], 'lucu': True}

Type Data: Dictionary

- *Update values* dari sebuah *keys* dan menambah item dengan menggunakan method `update()`

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 kucing.update({"umur": 1.5}) # update value dari key umur  
10 print(kucing) # print dict setelah update value  
11  
12 kucing.update({"warna": ["putih", "hitam"]}) # menambah item pada dict  
13 print(kucing) # print dict setelah menambah item
```

```
{'nama': 'Kuro', 'umur': 1.5, 'ras': 'Persian', 'jantan': True, 'hobi': ['makan', 'tidur']}
```

```
{'nama': 'Kuro', 'umur': 1.5, 'ras': 'Persian', 'jantan': True, 'hobi': ['makan', 'tidur'], 'warna': ['putih', 'hitam']}
```

Type Data: Dictionary

- Mengubah item pada *dictionary* menjadi sebuah *tuple* dalam *list* menggunakan method `items()` dan mengecek apakah terdapat suatu *key* dalam *dictionary*

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 x = kucing.items()  
10 print(x)  
11  
12 if "nama" in kucing:  
13     print("Ada key 'nama' pada dictionary")
```

```
dict_items([('nama', 'Kuro'), ('umur', 2), ('ras', 'Persian'), ('jantan', True), ('hobi', ['makan', 'tidur'])])  
Ada key 'nama' pada dictionary
```

Type Data: Dictionary

- Menghapus item pada *dictionary* pada key tertentu dengan metode pop()

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 kucing.pop("jantan")  
10 print(kucing)
```

```
{'nama': 'Kuro', 'umur': 2, 'ras': 'Persian', 'hobi': ['makan', 'tidur']}
```


Terimakasih

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution