



DASAR PEMROGRAMAN

INF1004AA

LAPORAN PRAKTIKUM 8:

Set & Dictionary

Oleh :

INDRIAN

1911102441003

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur
Samarinda, 30 November 2019

Laporan Praktikum 8: Set & Dictionary

Pokok Bahasan:

- ❖ Tipe Data Set & Dictionary

Tujuan Pembelajaran:

- ✓ Memahami tipe data List dan Tuple Python 3

1. Tipe Data Tuple

Set merupakan salah satu tipe data collection yang bersifat unordered dan unindexed, yang artinya suatu set tidak dapat diurutkan dan tidak memiliki index/key. Set atau dalam bahasa Indonesia disebut juga dengan himpunan merupakan suatu struktur data yang mirip dengan himpunan dalam matematika.

Percobaan & Latihan 8.1:

Code:

```
praktikum8 > code > latihan8_1.py > ...
1  name = {"INDRIAN", "RIAN", 1911102441003}
2
3  for n in name:
4      print(n)
5
6  print("Count Index: ", len(name))
```

Output:

```
RIAN
INDRIAN
1911102441003
Count Index: 3
```

Penjelasan:

- Pada line 3-4: Melakukan loop data Set dan mengeluarkan data didalamnya
- Pada line 6: Menghitung jumlah data Set

2. Method pada Set

Suatu set memiliki method bawaan (built-in) yang digunakan untuk bermacam keperluan, seperti menambahkan dan menghapus elemen; mengosongkan set dan lain sebagainya.

Percobaan & Latihan 8.2:

Code:

```
praktikum8 > code > latihan8_2.py > ...
1  name = {"INDRIAN", "RIAN", 1911102441003}
2  newName = {}
3  print(name, newName)
4
5  # Method add(<ELEMENT>)
6  name.add("Android Developer")
7
8  # Method copy()
9  newName = name.copy()
10 print(newName)
11
12 # Method clear()
13 newName.clear()
14 print(newName)
15
16 # Method difference(<SET>)
17 newName = newName.copy()
18 name.add("Flutter Developer")
19 print(name.difference(newName))
20
21 # Method discard(<ELEMEN>)
22 name.discard("Flutter Developer")
23 newName.clear()
24 newName.add("Android Developer")
25 print(name)
26
27 # Method intersection(<SET>)
28 print(name.intersection(newName))
29
30 # Method intersection_update ()
31 newName.intersection_update(name)
32 print(newName)
```

```
33
34 # Method isdisjoint()
35 print(name.isdisjoint(newName))
36
37 # Method issuperset(<SET>)
38 print(name.issuperset(newName))
39
40 # Method pop()
41 name.pop()
42 print(name)
43
44 # Method remove(<ELEMEN>)
45 name.remove("RIAN")
46 print(name)
47
48 # Method symmetric_difference(<SET>)
49 newName.symmetric_difference(name)
50 print(newName)
51
52 # Method symmetric_difference_update(<SET>)
53 name.symmetric_difference_update(newName)
54 print(name)
55
56 # Method union(<SET>)
57 name.union(newName)
58 print(name)
59
60 # Method update(<SET>)
61 name.update(newName)
62 print(name)
```

Output:

```
{1911102441003, 'RIAN', 'INDRIAN'} {}  
{'Android Developer', 1911102441003, 'RIAN', 'INDRIAN'}  
set()  
{1911102441003, 'RIAN', 'Android Developer', 'INDRIAN', 'Flutter'}  
{1911102441003, 'INDRIAN', 'RIAN', 'Android Developer'}  
{'Android Developer'}  
{'Android Developer'}  
False  
True  
{'INDRIAN', 'RIAN', 'Android Developer'}  
{'INDRIAN', 'Android Developer'}  
{'Android Developer'}  
{'INDRIAN'}  
{'INDRIAN'}  
{'INDRIAN', 'Android Developer'}  
PS F:\UMKT\PRAKTIKUM\PRAKTIKUM_DASAR_PEMROGRAMAN_PYTHON> []
```

Penjelasan:

- **add()** Menambahkan element pada Set
- **clear()** Menghapus semua element pada Set
- **copy()** Menyalin isi dari Set ke variable Set lain
- **difference(<SET>)** Mencari item-item atau himpunan dari dua buah himpunan x dan y, dimana elemen pada x yang tidak berada pada set y
- **discard(<ELEMEN>)** Menghapus item pada Set
- **intersection(<SET>)** Menghasilkan suatu set baru yang berisi elemen-elemen yang sama pada semua set. Misalkan kita memiliki set x dan set y yang masing-masing berisi elemen-elemen, fungsi ini dapat digunakan untuk menghasilkan set baru misalkan set c yang berisi elemen-elemen yang sama diantara kedua set.
- **intersection_update ()** Fungsinya mirip dengan method intersection, namun digunakan untuk memperbaharui elemen pada sebuah set berdasarkan elemen yang sama dengan semua set. Misalkan kita memiliki set x dan set y yang masing-masing berisi elemen-elemen, fungsi ini dapat digunakan untuk memperbaharui elemen pada set x menjadi elemen-elemen yang sama diantara kedua set
- **isdisjoint()** Mengecek apakah dua buah set saling tidak terkait. Method ini akan menghasilkan True jika dua buah set tidak saling terkait dan False jika sebaliknya
- **issuperset(<SET>)** Mengecek apakah sebuah set merupakan superset dari set yang lainnya. Method ini akan mengembalikan nilai True jika suatu set merupakan superset dari set yang lainnya dan mengembalikan nilai False jika sebaliknya
- **pop()** Menghapus sebarang element pada Set

- **remove(<ELEMEN>)** Menghapus element spesifik pada Set
- **symmetric_difference(<SET>)** Menghasilkan suatu set baru yang berisi elemen-elemen yang bukan merupakan irisan dari dua buah set
- **symmetric_difference_update(<SET>)** Untuk sebuah set yang berisi elemen-elemen yang bukan merupakan irisan dari set yang lain
- **union(<SET>)** Menghasilkan set baru hasil penggabungan dua atau lebih set
- **update(<SET>)** Menambahkan atau meng-update elemen suatu set dengan elemen dari set yang lain.

3. Tipe Data Dictionary

Tipe data dictionary ialah salah satu tipe data collection yang bersifat unordered dan indexed. Tipe data ini yang mirip dengan tipe data List, namun tipe data List indexnya merupakan angka, tipe data ini memiliki index string. Tipe data jenis ini juga disebut dengan Associative Array.

Percobaan & Latihan 8.2:

Code:

```
praktikum8 > code > latihan8_3.py > ...
1  name = {"INDRIAN", "RIAN", 1911102441003}
2  print(name)
3
4  name = dict({"name": "INDRIAN", "nim": 1911102441003})
5  print(name)
6
7  name = dict({1: "INDRIAN", 2: 1911102441003})
8  print(name)
9
10 name = {}
11 print(name)
```

Output:

```
{1911102441003, 'RIAN', 'INDRIAN'}
{'name': 'INDRIAN', 'nim': 1911102441003}
{1: 'INDRIAN', 2: 1911102441003}
{}
PS F:\UMKT\PRAKTIKUM\PRAKTIKUM_DASAR_PEMROGRAMAN_PYTHON>
```

Penjelasan:

- Line 4-5: Membuat Set dengan index key
- Line 7-8: Membuat Set dengan index Integer
- Line 10-11: Membuat Set yang kosong

4. Operasi pada Dictionary

Dictionary memiliki Operasi didalamnya berupa Menambahkan suatu element pada dictionary, Mengakses data pada dictionary, dan mengecek index pada dictionary menggunakan perintah for.

Percobaan & Latihan 8.4:

Code:

```
praktikum8 > code > latihan8_4.py > ...
1  myBio = {
2      "nama": "INDRIAN",
3      "prodi": "Teknik Informatika",
4      "nim": 1911102441003,
5      "angkatan": "2019"
6  }
7
8  print("Nama: ", myBio["nama"])
9  print("Prodi: ", myBio["prodi"])
10 print("Nim: ", myBio["nim"])
11 print("Angkatan: ", myBio["angkatan"])
```

Output:

```
Nama: INDRIAN
Prodi: Teknik Informatika
Nim: 1911102441003
Angkatan: 2019
```

Penjelasan:

- Line 1-6 Membuat variable Set dengan index key
- Line 8-11 Mengeluarkan element-element yang sudah di buat

5. Perulangan pada Dictionary

Dictionary dapat melakukan perulangan didalamnya, ada perbedaan perulangan pada dictionary dengan perulangan pada list, tuple, dan set.

Percobaan & Latihan 8.5:

Code:

```
praktikum8 > code > latihan8_5.py > ...
1  bio = {
2      "name": "INDRIAN",
3      "nim": 1911102441003
4  }
5
6  for key in bio:
7      print(bio[key])
8
9  for key, value in bio.items():
10     print("%s: %s" % (key, value))
```

Output:

```
INDRIAN
1911102441003
name: INDRIAN
nim: 1911102441003
```

Penjelasan:

- Line 6-7 Looping Set mengeluarkan key
- Line 9-10 Looping Set mengeluarkan Key, dan Value

6. Method pada Dictionary

Percobaan & Latihan 8.6:

Code:

```
praktikum8 > code > latihan8_6.py > ...
1  bio = {
2      "name": "INDRIAN",
3      "nim": 1911102441003
4  }
5  backup_bio = bio.copy()
6  print(bio)
7
8
9  # Method clear()
10 bio.clear()
11 print(bio)
12
13 # Method copy()
14 bio = backup_bio.copy()
15 print(bio)
16
17 # Method fromkeys(<KEY>[, <VALUES>])
18 x = {"key1", "key2", "key3"}
19 print(dict.fromkeys(x, 16))
20
21 # Method get(<INDEX>)
22 print(bio.get("nim"))
23
24 # Method items()
25 print(bio.items())
26
27 # Method keys()
28 print(bio.keys())
29
30 # Method pop()
31 bio.pop("name")
32 print(bio)
```

```
33 |
34 # Method setdefault()
35 x = bio.setdefault("prodi", "Teknik Informatika")
36 print(x)
37
38 # Method update()
39 bio.update({"name": "INDRIAN"})
40 print(bio)
41
42 # Method values()
43 print(bio.values())
```

Output:

```
{'name': 'INDRIAN', 'nim': 1911102441003}
{}
{'name': 'INDRIAN', 'nim': 1911102441003}
{'key2': 16, 'key1': 16, 'key3': 16}
1911102441003
dict_items([('name', 'INDRIAN'), ('nim', 1911102441003)])
dict_keys(['name', 'nim'])
{'nim': 1911102441003}
Teknik Informatika
{'nim': 1911102441003, 'prodi': 'Teknik Informatika', 'name': 'INDRIAN'}
dict_values([1911102441003, 'Teknik Informatika', 'INDRIAN'])
```

Penjelasan:

- **clear()** Menghapus semua elemen
- **copy()** Menyalin suatu variabel dictionary ke variabel lainnya
- **formkeys(<KEY>[, <VALUES>])** Untuk menciptakan suatu dictionary dengan index dan value yang spesifik.
- **get(<INDEX>)** Mengambil suatu value berdasarkan index yang diberikan.
- **items()** Menampilkan suatu dictionary dalam bentuk objek. Suatu objek akan berisi suatu index dan value suatu dictionary secara berpasangan seperti tuple dalam list.
- **keys()** Mengembalikan suatu objek dictionary yang berisi index dari suatu dictionary
- **pop()** Mengembalikan suatu elemen yang dihapus berdasarkan spesifik index.
- **setdefault()** Mengambil suatu nilai berdasarkan index tertentu dan dapat mengembalikan suatu nilai default jika index tersebut tidak ada.
- **update()** Memasukan spesifik elemen pada dictionary. Elemen-elemen yang dapat disisipkan dapat berupa dictionary atau iterable object.
- **values()** Mengambil suatu nilai dari dictionary dalam bentuk list().

Laporan resmi:

- **Latihan 8.1**
Penggunaan tipe data set suatu tipe data yang digunakan membuat suatu kelompok data tanpa ada data yang sama
- **Latihan 8.2**
Method pada set mempermudah untuk mengoperasikan suatu elemen tertentu menggunakan add, copy, clear, difference, discard, intersection, intersection_update, Isdisjoint, Issuperset, Pop, Remove, Symmetric_Difference, Symmetric_Difference_Update, Union, Update.
- **Latihan 8.3**
Tipe data Dictionary mirip dengan tipe data list, namun tipe data dict menggunakan string
- **Latihan 8.4**
Operasi data dictionary memiliki Operasi didalamnya berupa Menambahkan suatu element pada dictionary, Mengakses data pada dictionary, dan mengecek index pada dictionary menggunakan perintah for
- **Latihan 8.5**
Mengulang atau mencetak ulang elemen yang terdapat pada suatu variabel
- **Latihan 8.6**
Method pada dictionary memiliki banyak operasi seperti Clear, Copy, Fromkeys, Get, Items, Keys, Pop, Setdefault, Update, Values. Operasi tersebut memudahkan untuk membuat suatu program tanpa menambahkan statement baru.

Kesimpulan:

Set & Dictionary berbeda dengan list ataupun tuple karena setiap urutannya berisi key dan value yang dipisahkan oleh titik dua (:), item dipisahkan oleh koma (,), dict kosong dapat menggunakan kurung kurawal {}.

Repository:

https://github.com/indrian16/PRAKTIKUM_DASAR_PEMROGRAMAN_PYTHON