



PRAKTIKUM DASAR PEMROGRAMAN

INF1005

LAPORAN PRAKTIKUM 8 : Set & Dictionary

Oleh :

Akhmad Qasim

2211102441237

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2022

Laporan Praktikum 8:

Set & Dictionary

Latihan :

8.1. Tipe Data Tuple

- a.
- b. Output

```
Akhmad
Qasim
2211102441237

Jumlah index pada variable nama adalah 3
```

Tipe data set merupakan tipe data yang tidak dapat diindex dan diurutkan.
Tipe data set menampilkan output secara acak, artinya tidak ada urutannya.

8.2. Method pada Set

- a.
- b. Syntax

```
1  nama = {"Akhmad", "Qasim", 2211102441237}
2  nama.add('Teknik Informatika')
3  print(nama)
4
5  nama = {"Akhmad", "Qasim", 2211102441237}
6  nama.clear()
7  print(nama)
8
9  nama = {"Akhmad", "Qasim", 2211102441237}
10 nama_baru = nama.copy()
11 print(nama_baru)
12
13 nama = {"Akhmad", "Qasim", 2211102441237}
14 nama_2 = {"Akhmad", "Qasim", 2211102441239}
15 print(nama.difference(nama_2))
```

```
16
17  nama = {"Akhmad", "Qasim", 2211102441237}
18  nama.discard("Qasim")
19  print(nama)
20
21  nama = {"Akhmad", "Qasim", 2211102441237}
22  nama_2 = {"Akhmad", "Qasim", 2211102441212}
23  print(nama.intersection(nama_2))
24
25  nama = {"Akhmad", "Qasim", 2211102441237}
26  nama_2 = {"Akhmad", "Qasim", 2211102441212}
27  nama.intersection_update(nama_2)
28  print(nama)
29
30  nama = {"Akhmad", "Qasim", 2211102441237}
31  nama_2 = {"Akhmad", "Qasim", 2211102441212}
32  print(nama.isdisjoint(nama_2))
33
34  nama = {"Akhmad", "Qasim", 2211102441237}
35  nama_2 = {"Akhmad", "Qasim", 2211102441212}
36  print(nama.issuperset(nama_2))
37
38  nama = {"Akhmad", "Qasim", 2211102441237}
39  print(nama.pop())
40  print(nama)
41
42  nama = {"Akhmad", "Qasim", 2211102441237}
43  nama.remove("Akhmad")
44  print(nama)
45
46  nama = {"Akhmad", "Qasim", 2211102441237}
47  nama_2 = {"Akhmad", "Qasim", 2211102441212}
48  print(nama.symmetric_difference(nama_2))
49
50  nama = {"Akhmad", "Qasim", 2211102441237}
51  nama_2 = {"Akhmad", "Qasim", 2211102441212}
52  nama.symmetric_difference_update(nama_2)
53  print(nama)
54
55  nama = {"Akhmad", "Qasim", 2211102441237}
56  nama_2 = {"Akhmad", "Qasim", 2211102441212}
57  print(nama.union(nama_2))
58
59  nama = {"Akhmad", "Qasim", 2211102441237}
60  nama_2 = {"Akhmad", "Qasim", 2211102441212}
61  nama.update(nama_2)
62  print(nama)
```

Output

```
{'Akhmad', 2211102441237, 'Teknik Informatika', 'Qasim'}
set()
{'Akhmad', 2211102441237, 'Qasim'}
{2211102441237}
{'Akhmad', 2211102441237}
{'Akhmad', 'Qasim'}
{'Akhmad', 'Qasim'}
False
False
Akhmad
{2211102441237, 'Qasim'}
{2211102441237, 'Qasim'}
{2211102441212, 2211102441237}
{2211102441212, 2211102441237}
{'Akhmad', 2211102441237, 2211102441212, 'Qasim'}
{'Akhmad', 2211102441237, 2211102441212, 'Qasim'}
```

Penjelasan :

- i. Add
Method yang digunakan untuk menambahkan elemen pada set.
- ii. Clear
Method yang digunakan untuk mengosongkan semua elemen set.
- iii. Copy
Method ini digunakan untuk menyalin semua elemen pada variable set yang ada ke variable baru sudah ditentukan.
- iv. Difference
Method ini digunakan untuk membandingkan 2 variable yang bertipe data set. Method ini akan mengeluarkan output jika variable y tidak memiliki elemen yang sama dengan variable x.
- v. Discard
Method yang digunakan untuk menghapus sebuah elemen dari set.
- vi. Intersection
Method ini digunakan untuk membandingkan 2 elemen dan mengeluarkan output apabila elemen bernilai sama diantara variable.
- vii. Intersection_update
Method yang berfungsi untuk membandingkan 2 variable dan mencari

elemen yang sama, setelah itu mengubah variable dari hasil perbandingan yang dilakukan.

viii. Isdisjoint

Method yang digunakan untuk mengecek isi elemen dari variable apakah sama atau tidak. Jika berbeda akan mengeluarkan output true dan false jika terdapat keterkaitan terhadap semua elemen.

ix. Issuperset

Method yang digunakan untuk mengecek persamaan dari variable kedua ke yang pertama. Jika terdapat kesamaan semua elemen pada variable pertama maka akan mengeluarkan nilai true dan apabila variable pertama tidak memiliki semua elemen yang ada di variable kedua maka akan bernilai false.

x. Pop

Method ini digunakan untuk menghapus sebuah elemen secara acak dan mengembalikan sebuah nilai dari elemen yang telah dihapus.

xi. Remove

Method ini digunakan untuk menghapus elemen secara spesifik.

xii. Symmetric_difference

Method ini digunakan untuk menggabungkan 2 variable dan mengembalikan nilai berupa tipe data set serta elemen-elemen dari gabungan 2 elemen. Elemen yang sama tidak dimasukkan ke dalam set yang baru.

xiii. Symmetric_difference_update

Method yang digunakan untuk merubah isi elemen pada variable dari hasil gabungan variable kedua.

xiv. Union

Method yang menggabungkan seluruh isi set dan mengembalikan tipe data set baru.

xv. Update

Method ini digunakan untuk mengupdate suatu set dengan set lainnya.

8.3. Tipe Data Dictionary

a.

b. Syntax

```
1 mahasiswaA = {"nama": "Akhmad Qasim", "nim": "2211102441237"}
2
3 user = "Akhmad Qasim" # Di input user
4
5 if user == mahasiswaA["nama"]:
6     print("Anda mahasiswa kelas A, dengan NIM :", mahasiswaA["nim"])
7 else:
8     print("Anda bukan mahasiswa kelas A")
9 |
```

Output

```
Anda mahasiswa kelas A, dengan NIM : 2211102441237
```

Dictionary digunakan sebagai penyimpanan data terhadap mahasiswa, kita dapat menggunakan struktur kontrol untuk mengecek apakah mahasiswa tersebut adalah kelas A atau bukan.

8.4. Operasi pada Dictionary

a.

b.

c. Syntax

```
1 formatData = ["nama", "prodi", "nim", "angkatan"]
2 data = ["Akhmad Qasim", "S1 Teknik Informatika", 2211102441237, 2022]
3 dic = {}
4
5 for x, y in zip(formatData, data):
6     dic[x] = y
7
8 nama = dic["nama"]
9 prodi = dic["prodi"]
10 nim = dic["nim"]
11 angkatan = dic["angkatan"]
12
13 userInput = "Akhmad Qasim"
14 nimInput = 2211102441237
15
16 if userInput == nama and nimInput == nim:
17     print("Nama :", nama)
18     print("\nProdi :", prodi)
19     print("\nNIM :", nim)
20     print("\nAngkatan :", angkatan)
21 else:
22     print("Anda belum terdaftar")
23
```

Output

```
Nama : Akhmad Qasim
Prodi : S1 Teknik Informatika
NIM : 2211102441237
Angkatan : 2022
```

Format data digabungkan dengan data user menggunakan for ke dalam dictionary di variable dic. If digunakan sebagai pengecek, apakah userInput dan nimInput ada pada dictionary, apabila data nama dan nim sesuai maka akan menampilkan nama, prodi, nim, dan angkatan., sebaliknya jika tidak ada maka akan menampilkan output “Anda belum terdaftar”.

8.5. Perulangan pada Dictionary

- a.
- b.
- c. Syntax

```
1 bio = {
2     "nama": "Akhmad Qasim",
3     "nim": 2211102441237
4 }
5
6 print("Perulangan pada Index :")
7 for x in bio:
8     print(x)
9
10 print("\nAkses Value pada Perulangan :")
11 for x in bio:
12     print(bio[x])
13
14 print("\nPerulangan pada Value :")
15 for x in bio.values():
16     print(x)
17
18 print("\nPerulangan pada Item/Elemen :")
19 for x, y in bio.items():
20     print(x, y)
21
```

Output

```
Perulangan pada Index :
nama
nim

Akses Value pada Perulangan :
Akhmad Qasim
2211102441237
```

```
Perulangan pada Value :  
Akhmad Qasim  
2211102441237  
  
Perulangan pada Item/Elemen :  
nama Akhmad Qasim  
nim 2211102441237
```

Hasil dari perulangan disimpan ke variable x dan menampilkan keynya. Pada baris 11 digunakan perulangan yang sama tetapi mengakses value menggunakan bio[0]. Pada baris 15 dilakukan perulangan dan mengambil value yang dimasukkan ke variable lokal x. Pada perulangan terakhir dilakukan pengambilan data key dan value yang dimasukkan ke dalam variable x untuk key dan y sebagai value.

8.6. Method pada Dictionary

- a.
- b.
- c. Syntax

```
1  print("Clear :")  
2  bio = {  
3      "nama": "Akhmad Qasim",  
4      "nim": 2211102441237  
5  }  
6  bio.clear()  
7  print(bio)  
8  
9  print("\nCopy :")  
10 bio = {  
11     "nama": "Akhmad Qasim",  
12     "nim": 2211102441237  
13 }  
14 a2 = bio.copy()  
15 print(a2)  
16  
17 print("\nFormkeys :")  
18 a3 = ("nama", "nim", "kelas")  
19 b3 = "null"  
20 c3 = dict.fromkeys(a3, b3)  
21 print(c3)  
22  
23 print("\nGet :")  
24 a4 = bio.get("nama")  
25 print(a4)
```



```
26
27     print("\n Items :")
28     a5 = bio.items()
29     print(a5)
30
31     print("\n Keys :")
32     a6 = bio.keys()
33     print(a6)
34
35     print("\nPop :")
36     print(bio.pop("nim"))
37     print(bio)
38
39     print("\nSetdefault :")
40     a8 = bio.setdefault("nim", 2211)
41     print(a8)
42
43     print("\nUpdate :")
44     bio.update({"nim": 2211102441237})
45     print(bio)
46
47     print("\nValues :")
48     a10 = bio.values()
49     print(a10)
50
```

Output

```
Clear :
{}

Copy :
{'nama': 'Akhmad Qasim', 'nim': 2211102441237}

Formkeys :
{'nama': 'null', 'nim': 'null', 'kelas': 'null'}

Get :
Akhmad Qasim

Items :
dict_items([('nama', 'Akhmad Qasim'), ('nim', 2211102441237)])

Keys :
dict_keys(['nama', 'nim'])
```

```
Pop :
2211102441237
{'nama': 'Akhmad Qasim'}

Setdefault :
2211

Update :
{'nama': 'Akhmad Qasim', 'nim': 2211102441237}

Values :
dict_values(['Akhmad Qasim', 2211102441237])
```

Penjelasan method yang digunakan :

- i. Method clear digunakan pada baris ke-6, berfungsi untuk menghapus semua isi dari dictionary bio.
- ii. Method copy digunakan untuk menyalin isi dari dictionary dan disimpan ke dalam variable baru.
- iii. Pada baris ke-20, digunakan method fromkeys yang bertujuan membuat dictionary baru dengan index dan value yang unik.
- iv. Method get digunakan untuk mengambil value dari key yang dimasukkan ke parameter.
- v. Method items digunakan untuk mengambil semua data key dan value yang ada pada dictionary.
- vi. Method keys digunakan untuk mengambil semua key dari sebuah dictionary.
- vii. Method pop digunakan untuk menghapus sebuah elemen berdasarkan dari nilai key nya dan mengembalikan nilai tersebut.
- viii. Method setdefault digunakan untuk mengembalikan sebuah nilai default jika nilai yang di ambil tidak ada.
- ix. Method update digunakan mengubah sebuah value yang ada di dictionary dengan menggunakan key nya.

- x. Method values digunakan untuk mengambil semua data value yang ada pada sebuah dictionary.

Hasil dan analisa dari Percobaan & Latihan 8.1 sampai dengan 8.6 :

- 8.1. Tipe data set merupakan sebuah tipe data yang tidak dapat di indeks dan tipe data ini tidak memiliki urutan terhadap elemennya.
- 8.2. Tipe data set mempunyai berbagai macam method yang dapat digunakan untuk membandingkan, menghitung, menambahkan, mengosongkan set, dan menghapus.
- 8.3. Tipe data dictionary merupakan sebuah tipe data yang dapat di indeks dengan key tapi tidak memiliki urutan. Tipe data dictionary hampir sama dengan tipe data list, hanya saja jika list menggunakan angka sebagai indeksnya sedangkan dictionary menggunakan string sebagai indeksnya.
- 8.4. Tipe data dictionary memiliki operasi untuk menambahkan elemen, mengakses elemen, dan melakukan pengecekan pada index dan value.
- 8.5. Tipe data dictionary dapat diiterasi menggunakan for untuk mendapatkan nilai key dari dictionary dan juga value dari dictionary.
- 8.6. Tipe data dictionary method yang dapat digunakan untuk menghapus semua elemen, menyalin semua elemen, menciptakan dictionary baru dari index dan value yang spesifik, mendapatkan sebuah elemen, dan masih banyak lagi method yang dapat digunakan.

Kesimpulan :

Tipe data set merupakan tipe data yang tidak dapat di indeks dan tidakurut, digunakan dalam beberapa studi kasus seperti himpunan dalam matematika yang tidak memerlukan indeks serta dapat melakukan operasi gabungan, irisan, selisih, dan komplemen. Sedangkan Tipe data dictionary sebuah tipe data yang memiliki key sebagai indeksnya tetapi tidak berurutan, digunakan untuk memetakan sebuah value ke dalam key. Key dan value pada tipe data dictionary sangat fleksibel, karena dapat menggunakan string, number, array, dan lainnya.