



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231052
Nama Lengkap	GABRIEL SACHIO ATMADJAJA
Minggu ke / Materi	13 / Tipe Data Set

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Sumber: Modul Pratikum Algoritma dan Pemrograman

MATERI 1

Set adalah tempat menyimpan sekumpulan data yang semuanya unik. Set disebut juga sebagai himpunan. Beberapa sifat yang dimiliki set yaitu:

- Isi dari set disebut anggota
- Anggota dari set harus bersifat *immutable*
- Set bersifat *mutable*

Ada dua cara dalam membuat set yaitu dengan notasi { } dan satunya dengan menggunakan fungsi set(). Berikut contohnya:

```
setpy > ...
1  iniset = {'a', 'b', 'c', 'c'}
2
3  print(type(iniset))
4  print(iniset)
5
6  nama = set()
7  print(type(nama))
```

Gambar 1.1: Cara membuat set

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
<class 'set'>
{'c', 'a', 'b'}
<class 'set'>
```

Gambar 1.2: Output

Hal penting yang perlu diketahui lainnya adalah set tidak bisa diakses dengan index karena tidak mempunyai index. Jika di perhatikan di gambar 1.1 , pada variabel iniset urutannya dalam kodenya adalah 'a', 'b', 'c', dan 'c'. Tapi ketika kode di baris 4 yang print(iniset) dijalankan, outputnya memunculkan anggotanya secara random. Selain itu, duplikasi dalam akan dianggap satu anggota (contohnya c). Lalu dalam set, 1 dengan True itu dianggap sama dan 0 dengan False dianggap sama. Berikut maksudnya:

```
8
9  a = {1, True, 0, False}           # True dan 1 dianggap sama, False dan 0 dianggap sama
10 print(a)
11
```

Gambar 1.3: Contoh duplikasi lain yang dianggap sama dalam set

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
{0, 1}
```

Gambar 1.4: Output

Fungsi pada set

1) ADD

```
12 # ADD
13 a = {"1", "2", "3"}
14 a.add(7)
15 a.add(8)
16 a.add(9)
17 print(a)
```

Gambar 1.5: Fungsi add

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
{'3', 7, 8, 9, '1', '2'}
```

Gambar 1.6: Output

2) POP

```
19 # POP --> bisa return nilai
20 b = a.pop()
21 print(b)
```

Gambar 1.7: Fungsi Pop

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
3
```

Gambar 1.8: Output

3) DISCARD

```
23 # DISCARD
24 a.discard("3")
25 print(a)
```

Gambar 1.9: Fungsi Discard

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
{'1', '2'}
```

Gambar 1.10: Output

4) REMOVE

```
27 # REMOVE
28 a.remove("3")
29 print(a)
```

Gambar 1.11: Fungsi remove

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
{'1', '2'}
```

Gambar 1.12: Output

5) CLEAR

```
31 # CLEAR
32 a.clear()
33 print(a)
```

Gambar 1.13: Fungsi clear

```
PS D:\Sachio\Kampus\PrakAlPro\Pert13> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
set()
```

Gambar 1.14: Output

Operasi pada set

1) Union → kata kuncinya menggabungkan anggota

Ada 3 cara membuat union:

```
39 Union
40 1) set1.update(set2)    --> akan merubah set1
41 2) set1.union(set2)    --> tdk akan merubah set1
42 3) set3 = set1 | set2
```

Gambar 1.15: Cara membuat union

2) Intersection → kata kuncinya elemen yang sama

Ada 2 cara membuat intersection:

```
44 Intersection
45 1) set1.intersection(set2)
46 2) set3 = set1 & set2
```

Gambar 1.16: Cara membuat intersection

3) Difference → kata kuncinya elemen yang beda di salah satu himpunan/set

Ada 2 cara membuat operasi difference:

```
48 Difference
49 1) set1.difference(set2)
50 2) set3 = set1 - set2
```

Gambar 1.17: Cara membuat operasi difference

4) Symmetric Difference → kata kuncinya semua elemen yang berbeda dalam set

Ada 2 cara membuat operasi symmetric difference:

```
52 Symmetric difference
53 1) set1.symmetric_difference(set2)
54 2) set3 = set1 ^ set2
```

Gambar 1.18: Cara membuat operasi symmetric difference

BAGIAN 2: LATIHAN MANDIRI (60%)

Source: <https://github.com/gabrielsachioa/PrakAIPro12.git>

SOAL 1

Berikut tambahan kemampuan pada kasus kategori google play store:

```
lat12-1.py > ...
1  n = int(input("Masukkan jumlah kategori: "))
2
3  aplikasi = {}
4  for i in range(n):
5
6      kategori = str(input(f"Masukkan nama kategori ke-{i+1}: "))
7      set_app = set()
8
9      for j in range(5):
10         inputan_app = str(input(f"\tMasukkan nama aplikasi ke-{j+1}: "))
11         set_app.add(inputan_app)
12
13     aplikasi[kategori] = set_app
14
15 list_kategori = []
16 for kategori in aplikasi:
17     list_kategori.append(aplikasi[kategori])
18
19 hasil = list_kategori[0]
20 for k in range(1, len(list_kategori)):
21     hasil = hasil.intersection(list_kategori[k])
22
23 hasil2 = list_kategori[0]
24 for l in range(1, len(list_kategori)):
25     hasil2 = hasil2.symmetric_difference(list_kategori[l])
26
27 print(f"Kategori yang muncul semua: {hasil}")
28 print(f"Kategori yang muncul 1 kali: {hasil2}")
29 if n > 2:
30     hasil3 = list_kategori[0]
31     tmp = list_kategori[1]
32     for m in range(1, len(list_kategori)):
33         hasil3 = hasil3.symmetric_difference(list_kategori[m])
34     print(f"Kategori yang muncul 2 kali: {hasil3}")
35 else:
36     print("Kategori yang muncul 2 kali tidak ada")
```

Gambar 2.1: Menambahkan kemampuan di kasus kategori google play store

```
PS D:\Sachio\Tugas\prakalpro12> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Tugas/prakalpro12/lat12-1.py
Masukkan jumlah kategori: 2
Masukkan nama kategori ke-1: Finance
    Masukkan nama aplikasi ke-1: RTI Saham
    Masukkan nama aplikasi ke-2: Mirae
    Masukkan nama aplikasi ke-3: IPOT
    Masukkan nama aplikasi ke-4: Poems
    Masukkan nama aplikasi ke-5: Calculator
Masukkan nama kategori ke-2: Utilities
    Masukkan nama aplikasi ke-1: Photos
    Masukkan nama aplikasi ke-2: Weather
    Masukkan nama aplikasi ke-3: Calculator
    Masukkan nama aplikasi ke-4: Camera
    Masukkan nama aplikasi ke-5: Notes
Kategori yang muncul semua: {'Calculator'}
Kategori yang muncul 1 kali: {'Poems', 'Photos', 'Mirae', 'Camera', 'Weather', 'IPOT', 'RTI Saham', 'Notes'}
Kategori yang muncul 2 kali tidak ada
```

Gambar 2.2: Output

SOAL 2

```
lat12-2.py > ...
1  data = [1, 2, 3, 4, 5]
2
3  print("List menjadi Set: ")
4  print(f"Before = {data} \t, \t After = {set(data)}\n")
5
6  print("Set menjadi List: ")
7  print(f"Before = {set(data)} \t, \t After = {list(set(data))}\n")
8
9  print("Tuple menjadi Set: ")
10 print(f"Before = {tuple(data)} \t, \t After = {set(tuple(data))}\n")
11
12 print("Set menjadi Tuple: ")
13 print(f"Before = {set(data)} \t, \t After = {tuple(set(data))}\n")
```

Gambar 2.3: Program konversi suatu tipe data

Keterangan: Intinya cara termudah untuk mengkonversi suatu tipe data adalah dengan menggunakan fungsi dari python tersebut.

Output:

```
PS D:\Sachio\Tugas\prakalpro12> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
List menjadi Set:
Before = [1, 2, 3, 4, 5]          ,          After = {1, 2, 3, 4, 5}

Set menjadi List:
Before = {1, 2, 3, 4, 5}        ,          After = [1, 2, 3, 4, 5]

Tuple menjadi Set:
Before = (1, 2, 3, 4, 5)        ,          After = {1, 2, 3, 4, 5}

Set menjadi Tuple:
Before = {1, 2, 3, 4, 5}        ,          After = (1, 2, 3, 4, 5)
```

Gambar 2.4: Output

SOAL 3

```
lat12-3.py > ...
1 input1 = input("Masukkan file pertama: ")
2 input2 = input("Masukkan file kedua: ")
3
4 try:
5     with open(input1, 'r') as f1, open(input2, 'r') as f2:
6         file1 = f1.read().lower()
7         file2 = f2.read().lower()
8
9         file1 = file1.split()
10        file2 = file2.split()
11
12        file1 = set(file1)
13        file2 = set(file2)
14
15        hasil = file1.intersection(file2)
16        print(f"Kata yang muncul di kedua file: {hasil}")
17 except:
18     print("File tidak ditemukan")
```

Gambar 2.5: Program mencari

Keterangan: Intinya buat variabel untuk meminta inputan file seperti kode di baris 1 dan 2. Lalu dalam kondisi try (jika file ditemukan), buka kedua file tersebut dengan mode read dan simpan di variabel yang berbeda. Buat variabel untuk membaca isi file tersebut dan mengubah seluruh isinya menjadi lowercase (huruf non-kapital/huruf kecil) seperti di baris 6 dan 7. Selanjutnya, masing-masing file di split atau dibagi menjadi token-token berdasarkan spasi seperti di baris 9 dan 10 agar bisa diubah ke tipe data set nantinya. Setelah itu ubah masing-masing variable itu menjadi tipe data set dengan cara menggunakan fungsi set diikuti variabelnya, ini bertujuan supaya bisa menggunakan fungsi intersection. Setelah itu buat variabel hasil untuk menampung intersection dari kedua variable file1 dan file2. Setelah itu print output intersection tersebut. Terakhir, di dalam kondisi excep (jika file tidak ada/ tidak ditemukan/nama file salah), print “File tidak ditemukan”

Output:

```
PS D:\Sachio\Tugas\prakalpro12> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Tugas/prakalpro12/lat12-3.py
Masukkan file pertama: teks1.txt
Masukkan file kedua: teks2.txt
Kata yang muncul di kedua file: {'sayapku', 'potong', 'melayang', 'ku', 'syalala', 'andai'}
```

Gambar 2.6: Output