



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231042
Nama Lengkap	Revaldo Fransisco Hohary
Minggu ke / Materi	12 / 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%)

MATERI 1

A. Pengenalan dan Mendefinisikan Set

Set merupakan tipe data python yang digunakan untuk menyimpan sekumpulan data yang memiliki keunikan. Ini juga biasa dikenal istilah himpunan. Beberapa sifat set dalam python adalah:

- Isi dari Set disebut sebagai anggota (member).
- Anggota dari Set harus bersifat immutable. Beberapa tipe data immutable pada Python: integer, float, string, tuple dan lain-lain. Dengan demikian list dan dictionary (mutable) tidak dapat dimasukkan ke dalam Set.
- Set sendiri bersifat mutable, artinya anda dapat menambah atau mengurangi isi dari sebuah Set. Karena itu Set tidak dapat dimasukkan ke dalam Set.

Untuk definisi set sendiri, beberapa cara dapat dilakukan yaitu dengan menggunakan notasi {} dan fungsi set() seperti berikut :

```
# dengan menggunakan {}
genap = {2, 4, 6, 8, 10, 12}
ganjil = {1, 3, 5, 7, 9, 11}
# dengan menggunakan fungsi set()
mhs_ukdw = set('Revaldo', 'Soter')
```

Anda harus menggunakan fungsi set(), seperti yang ditunjukkan di bawah ini, untuk mendefinisikan Set kosong, tetapi notasi {} tidak dapat digunakan.

```
# dengan fungsi set()
mhs_ukdw = set() # menghasilkan set kosong
data = {} # ini akan menghasilkan dictionary kosong
```

B. Pengaksesan Set

Karena set tidak memiliki indeks, Anda tidak dapat secara langsung mengakses anggota set. Lihat contoh program berikut:

```
nim = {'71231042', '7231050', '71200214'}
jumlah_nim = len(nim)
print(jumlah_nim) # akan menghasilkan output 3
# tampilkan isi set satu-persatu
for n in nim:
    print(n)
```

Output yang dihasilkan dari program diatas adalah :

```
3
71231042
71231050
71200214
```

Karena Set tidak memiliki indeks, sehingga tidak ada urutan posisi anggota, output yang dihasilkan urutannya berbeda dengan deklarasi Set sebelumnya. Ini karena posisi anggota tidak penting untuk tipe data Set.

Set adalah jenis data yang dapat diubah, yang berarti isinya dapat bertambah atau berkurang. Cara menggunakan fungsi add untuk menambah anggota ke set ditunjukkan dalam program berikut:

```
# definisikan sebuah set kosong
plat = set()
# tambahkan plat nomor 'AB 1890 XA'
plat.add('AB 2006 XA')
# tambahkan plat nomor 'AD 6810 MT'
plat.add('PA 0101 MT')
# jumlah anggota di dalam Set
print(len(plat))
# tambahkan plat yang sama sekali lagi
plat.add('AB 2006 XA')
# tampilkan semua plat nomor
for nomor in plat:
    print(nomor)
```

Output yang dihasilkan :

```
2
PA 0101 MT
AB 2006 XA
```

Set memiliki mekanisme untuk melihat apakah anggota baru yang akan ditambahkan sudah ada di dalamnya (cek duplikasi). Jika tidak ada, anggota tersebut dapat dimasukkan. Namun, jika ada anggota dengan nilai yang sama di dalam Set, maka

Penggunaan fungsi add() tidak akan menambah anggota ke Set karena pengecekan duplikasi sudah ada di dalamnya. Ada beberapa cara untuk menghapus anggota dari set, seperti menggunakan fungsi discard(), remove(), pop(), dan clear(). Gambar 12.1 menunjukkan perbedaan antara empat fungsi tersebut. Program yang menunjukkan penghapusan anggota dari sebuah set dicontohkan di sini:

discard()	remove()	pop()	clear()
Menghapus satu elemen yang disebutkan	Menghapus satu elemen yang disebutkan	Mengambil salah satu dan menghapusnya dari set (tidak tentu)	Menghapus seluruh elemen di dalam set
Tidak ada error	Muncul error jika elemen yang dihapus tidak ada	Error jika set kosong	Tidak ada error

Gambar 12.1 : Fungsi untuk menghapus Set

```
prima = {13, 23, 7, 29, 11, 5}
# hapus 5 dari set tersebut
prima.remove(5)
print(prima)
# hapus 97 (tidak ada)
prima.discard(97)
print(prima)
# ambil dan hapus salah satu
bilangan = prima.pop()
print(bilangan)
print(prima)
# kosongkan set
prima.clear()
print(prima)
```

Output dari program diatas adalah :

```
{5, 7, 11, 13, 23, 29} # awal
{7, 11, 13, 23, 29} # setelah 5 dihapus. Perhatikan urutan
berubah
{7, 11, 13, 23, 29} # 97 tidak ada, sehingga Set tidak berubah
7 # fungsi pop() mengeluarkan 7
{11, 13, 23, 29} # setelah 7 keluar dari Set
set() # setelah isi dari Set dihapus semua
```

Jika anggota yang ingin dihapus tidak ada di dalam Set, fungsi discard() tidak akan menghasilkan error, tetapi fungsi pop() akan mengambil salah satu anggota secara acak dan mengeluarkannya dari Set. Fungsi pop() akan berguna jika kita ingin memproses isi dari Set satu-satunya tanpa mengalami error.

Jika kita mempertimbangkan urutan dan posisi setiap anggota dalam Set, bagaimana jika kita ingin mengubah nilai salah satu anggota? Nilai anggota dalam set tidak dapat diubah secara langsung. Satu-satunya cara untuk melakukannya adalah melakukan operasi penggantian, atau penggantian, dengan menghapus anggota yang ingin diubah, kemudian memasukkan anggota baru dengan nilai yang diinginkan. Program berikut menunjukkan contohnya:

```
# Buat Set dari List
ikan = set(['koi', 'koki', 'kembung', 'salmon'])
print(ikan)
# ganti koi menjadi teri
ikan.remove('koi')
ikan.add('teri')
print(ikan)
```

Output yang dihasilkan:

```
{'koi', 'koki', 'salmon', 'kembung'}
{'teri', 'salmon', 'kembung', 'koki'}
```

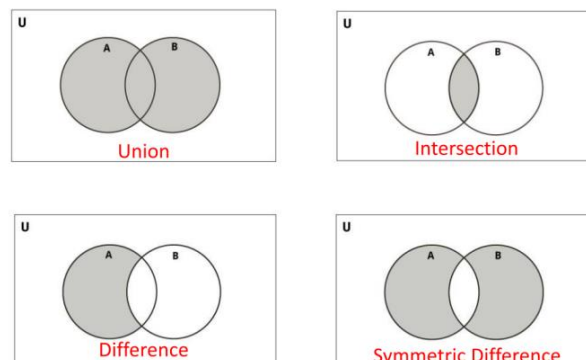
Untuk mengubah nilai "koi" menjadi "teri", hapus anggota "koi" dan kemudian masukkan anggota baru dengan nilai "teri". Sekali lagi, penting untuk menekankan bahwa set tidak mengenal urutan atau posisi data. sehingga output program

Nilai "koi" telah diganti dengan "teri". Setelah itu, urutan isi Set ikan saat ditampilkan berubah. Urutan anggota dalam Set biasanya akan berubah setiap kali ada penambahan atau penghapusan.

C. Operasi-Operasi pada Set

Operasi-operasi pada Set adalah operasi-operasi pada himpunan. Berikut ini adalah daftar operasi-operasi Set pada Python:

- Operator Union.** Menggabungkan dua Set menjadi satu. Dapat menggunakan operator `|` maupun fungsi `union()`.
- Operator Intersection.** Menghasilkan irisan dari dua Set. Dapat menggunakan operator `&` maupun fungsi `intersection()`.
- Operator Difference.** Menghasilkan Set baru yang merupakan selisih dari dua Set yang dibandingkan. Dapat menggunakan operator `-` maupun fungsi `difference()`.
- Operator Symmetric Difference.** Menghasilkan Set baru yang merupakan jumlah dari dua Set kecuali irisannya. Dapat menggunakan operator maupun fungsi `symmetric_difference()`.



Gambar 12.2 : Operasi pada Set

1. Operator Union

Contoh penggunaan operator union sebagai berikut:

```
merek_hp = {'Samsung', 'Apple', 'Xiaomi', 'Sony'}
merek_ac = {'LG', 'Samsung', 'Panasonic', 'Daikin', 'Sony'}
# union dari merek_hp dan merek_ac
gabungan = merek_hp | merek_ac
# bisa juga menggunakan gabungan = merek_hp.union(merek_ac)
print(gabungan)
```

Anggota dari Set `merek_hp` dan anggota dari Set `merek_ac` digabungkan oleh program, menghasilkan Set baru yang disebut `gabungan`. "Samsung" dan "Sony" adalah duplikasi (anggota yang sama), jadi mereka ada di Set.

Karena anggota set harus unik, gabungan hanya muncul sekali. Hasilnya adalah sebagai berikut:

```
{'Xiaomi', 'Sony', 'LG', 'Daikin', 'Samsung', 'Apple',  
'Panasonic'}
```

2. Operator Intersection

Contoh dari penggunaan operator Intersection dapat dilihat pada program berikut:

```
renang = {'siti', 'mail', 'ikhsan', 'upin', 'ipin'}  
tenis = {'joko', 'mail', 'ipin', 'upin', 'tejo'}  
# suka renang dan tenis  
renang_tenis = renang & tenis  
print(renang_tenis)
```

Program akan menghasilkan output dari operasi intersection dari Set Renang dan Tennis, yang berarti anggota yang berada di kedua set, yaitu mail, upin, dan ipin, sekaligus. Operasi intersection menghasilkan output seperti:

```
{'upin', 'ipin', 'mail'}
```

3. Operator Difference

Contoh penggunaan operator difference sebagai berikut:

```
# bisa berbahasa English  
english = {'desi', 'tono', 'evan', 'miko', 'takashi',  
'chaewon'}  
# bisa berbahasa korea  
korean = {'chaewon', 'yeona', 'erika', 'miko'}  
# siapa yang hanya bisa bahasa korea?  
only_korean = korean - english  
print(only_korean)  
# siapa yang hanya bisa bahasa english?  
only_english = english - korean  
print(only_english)
```

Set yang anggotanya adalah selisih dari dua set akan dihasilkan oleh operator perbedaan.

yang dibandingkan dengan Dalam kasus ini, digunakan untuk mendapatkan anggota yang hanya bisa berbahasa Korea dengan mencari selisih antara set bahasa Korea dan set bahasa Inggris. Namun, jika ingin mengetahui siapa saja yang hanya bisa berbicara bahasa Inggris, harus mencari antara set Korea dan set Inggris. Program tersebut menghasilkan output sebagai berikut:

```
{'erika', 'yeona'} # hanya bisa Korea  
{'takashi', 'desi', 'evan', 'tono'} # hanya bisa English
```

4. Operator Symmetric Difference

Contoh penggunaan operator ini sebagai berikut:

```
english = {'desi', 'tono', 'evan', 'miko', 'takashi', 'chaewon'}  
korean =
```

```
{'chaewon', 'yeona', 'erika', 'miko'} #  
hanya bisa bicara satu bahasa saja  
one_language = english ^ Korean  
print(one_language)
```

Operator perbedaan simetrik akan membuat set baru yang menggabungkan dua set tetapi tidak memasukkan irisannya. Contoh berikut menggunakan operator perbedaan simetri untuk mengidentifikasi seseorang yang hanya dapat berbicara dalam satu bahasa (artinya tidak melibatkan irisannya). Selain itu, biasanya dapat dihasilkan dari operasi berikut:

```
one_language = english.union(korean) -  
english.intersection(korean)
```

Outputnya sebagai berikut:

```
{'yeona', 'tono', 'desi', 'erika', 'evan', 'takashi'}
```

D. Link Github

<https://github.com/Frealy0901/Tugas-PrakAIPro13.git>

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1

```
latihan1.py > ...
1  jmlh_ktgri = int(input("masukan jumlah kategori : ")) #pada bagian ini kita meminta inputan kategori dari use
2  data_aplikasi = {} #buat set kosong
3
4  for i in range(jmlh_ktgri): # mengulang-ngulang i yang berisi hasil inputan user
5      nama = input("masukan nama katgori : ") # meminta user memasukan nama dari kategori tersebut
6      print("masukan 5 nama aplikasi di kategori ", nama) # memunculkan nama kategori dari inputan yqng dimasukan user
7      apk = list() #membuat list kosong
8
9      for j in range(5): # mengulang-ngulang sebanyak 5 kali
10         nama_apk = input("masukan nama aplikasi : ") # meminta inputan user untuk nama aplikasinya
11         apk.append(nama_apk) # memasukan nama aplikasi kedalam list apk diatas yang sudah kita buat
12         data_aplikasi[nama] = apk
13     print(data_aplikasi) # memunculkan variabel data aplikasi yang sudah dimasukan apk
14     print() #memunculkan
15
16     daftar_apk = list() # membuat list
17     for apk in data_aplikasi.values(): # jika apk ada di dalam data aplikasi
18         daftar_apk.append(set(apk)) # maka akan ditambahkan dan dirubah ke set
19     print(daftar_apk) # memunculkan daftar apk
20     print() # print kosong
21
22     hasil = daftar_apk # memasukan daftar aplikasi ke dalam hasil
23     for x in range(1,jmlh_ktgri): # mengulang-ulang x yang di mulai dari 1 sampe dengan jumlah kategori yang diinputkan pengguna
24         unik = daftar_apk[i] ^ daftar_apk[i-1] #membuat variabel unik yang berisi daftar aplikasi
25     else:
26         unik = daftar_apk[0] ^ daftar_apk[i]
27     print() # membuat print ksoong
28     print("applikasi yang hanya muncul di satu kategori saja: ", unik) # memunculkan kata unik yang sudah di lakukan
29
30     print()
31     if jmlh_ktgri > 2: # jika jumlah kategori lebih besar dari 2
32         for y in range(1,jmlh_ktgri): # akan mengulang y dari 1 sampai jumlah kategori dari inputan user
33             intr = daftar_apk[i] and daftar_apk[i-1]
34         else:
35             intr = daftar_apk[0] and daftar_apk[i]
36         print("aplikasi yang muncul di dua kategori sekaligus:", intr)
37
```

```
masukan jumlah kategori : 2
masukan nama katgori : barang
masukan 5 nama aplikasi di kategori barang
masukan nama aplikasi : bolpen
masukan nama aplikasi : pena
masukan nama aplikasi : mouse
masukan nama aplikasi : hp
masukan nama aplikasi : gunting
masukan nama katgori : jasa
masukan 5 nama aplikasi di kategori jasa
masukan nama aplikasi : keuangan
masukan nama aplikasi : kebersihan
masukan nama aplikasi : pariwisata
masukan nama aplikasi : travel
masukan nama aplikasi : dekor
{'barang': ['bolpen', 'pena', 'mouse', 'hp', 'gunting'], 'jasa': ['keuangan', 'kebersihan', 'pariwisata', 'travel', 'dekor']}

[{'bolpen', 'gunting', 'pena', 'hp', 'mouse'}, {'pariwisata', 'kebersihan', 'travel', 'keuangan', 'dekor'}]

applikasi yang hanya muncul di satu kategori saja: {'pariwisata', 'gunting', 'pena', 'hp', 'keuangan', 'mouse', 'bolpen', 'kebersihan', 'travel', 'dekor'}
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\Set>
```


SOAL 2

```
latihan2.py > ...
1 data = 'saya pergi ke pasar' # masukan kalimat
2 data_list = list(data) # rubah kalimat tersebut kedalam list
3
4 print("list ke set") # memunculkan list ke set
5 print("sebelum ->", data_list) # memunculkan data list
6 print("sesudah ->", set(data_list)) # memunculkan data list yang dirubah ke set
7 print() # memunculkan print kosong
8 data_set = set(data) # membuat variabel data set dan rubah kalimat yang sudah ada menjadi set
9 print("set ke list") # memunculkan set ke list
10 print("sebelum -> ", data_set) # memunculkan set sebelum dirubah ke list
11 print("sesudah -> ", list(data_set)) # memunculkan sesudah dirubah ke list
12 print() # memunculkan print kosong
13 data_tuple = tuple(data) # membuat variabel data tuple dan rubah kalimat yang sudah dimasukan ke tuple
14 print("tuple ke set") # memunculkan tuple ke set
15 print("sebelum -> ", data_tuple) # memunculkan data tuple yang belum di rubah ke set
16 print("sesudah -> ", set(data_tuple)) # memunculkan data tuple yang sudah dirubah ke set
17 print() # memunculkan print kosong
18 print("set ke tuple") # memunculkan set ke tuple
19 print("sebelum -> ", data_set) # sebelum di rubah ketuple
20 print("sesudah -> ", tuple(data_set)) # sesudah dirubah ke tuple
```

```
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\Set> & C:/Users/Asus/AppData/Local/Programs/Python/Python312/python.exe
list ke set
sebelum -> ['s', 'a', 'y', 'a', ' ', 'p', 'e', 'r', 'g', 'i', ' ', 'k', 'e', ' ', 'p', 'a', 's', 'a', 'r']
sesudah -> {'p', 'k', ' ', 'g', 'e', 'i', 'r', 'a', 'y', 's'}

set ke list
sebelum -> {'p', 'k', ' ', 'g', 'e', 'i', 'r', 'a', 'y', 's'}
sesudah -> ['p', 'k', ' ', 'g', 'e', 'i', 'r', 'a', 'y', 's']

tuple ke set
sebelum -> ('s', 'a', 'y', 'a', ' ', 'p', 'e', 'r', 'g', 'i', ' ', 'k', 'e', ' ', 'p', 'a', 's', 'a', 'r')
sesudah -> {'p', 'k', ' ', 'g', 'e', 'i', 'r', 'a', 'y', 's'}

set ke tuple
sebelum -> {'p', 'k', ' ', 'g', 'e', 'i', 'r', 'a', 'y', 's'}
sesudah -> ('p', 'k', ' ', 'g', 'e', 'i', 'r', 'a', 'y', 's')
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\Set>
```

SOAL 3

```
latihan3.py > ...
1 teks1 = input("Masukkan file teks 1: ") #meminta user memasukan inputan file teks
2 teks2 = input("Masukkan file teks 2: ")
3 try: # menggunakan try dan except jika ada salah masukan dari user
4     handle = open(teks1, 'r') # ini akan membuka file tersebut dan membaca teks tersebut
5     handle2 = open(teks2, 'r')
6 except:
7     print("File cannot be opened")
8     exit()
9 a = set() # membuat set yang kosong
10
11 for teks in handle: # mengulang teks disetiap handle
12     tekslow = teks.lower() # membuat semua teks kata menjadi huruf kecil
13     tekspl = tekslow.split() # split setiap kata
14     for b in tekspl: # untuk tekspl dimasukan kedalam b
15         a.add(b) # b ditambahkan kedalam variabel a
16
17 for teks in handle2: # mengulang teks disetiap handle
18     tekslow = teks.lower() # membuat semua teks kata menjadi huruf kecil
19     tekspl = tekslow.split() # split setiap kata
20     for b in tekspl: # untuk tekspl dimasukan kedalam b
21         a.add(b) # untuk tekspl dimasukan kedalam b
22 print(a) # memunculkan variabel a
```

PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Set> & C:/Users/Asus/AppData/Local/Programs/Python/Python312/python.exe c:/Users/Asus/Documents/UKDW/PrakAIPro/Set/latihan3.py

Masukkan file teks 1: file1.txt

Masukkan file teks 2: file2.txt

```
{'syamsuddin', 'orang', 'bakkan', 'dpr', 'selatan', 'pemeriksaan', 'ketua', 'rutan', 'ya', 'soal', 'di', 'dikonfirmasi', 'karena', 'tutur', 'liar', 'penyerahan', 'yang', 'ali', 'sekarang', 'pada', 'at
au', 'sudah', 'cabang', 'kembali', 'merah', '(21/5/2024)', 'syamsuddin', 'kepada', 'proses', 'bersangkutan', 'pidana', 'hukumannya', 'kali', 'azis', 'baca', 'korupsi', 'kpk', 'rutan', 'kuningan', 'o
knum', 'kasus', 'penyidikan', 'juga', 'uang', 'telah', 'artikel', 'banyak', 'memang', 'aclc', 'pimpinan', 'perlu', 'petugas', 'kan', 'baik', 'azis', 'diduga', 'kata', 'tersebut', 'diperiksa', 'gedung', 'per
nah', 'mengatakan', 'putih', 'masa', 'menghuni', 'kedua', 'dengan', 'https://apps.detik.com/detik/', 'selengkapnya', 'sedang', 'mendatangi', 'rabu', 'diserahkan', 'sebelumnya', '(pungli)', 'menjalani', 'in
i', 'terpidana', 'selasa', 'dpr', '66', 'dipecat', 'ali', 'syamsuddin', 'jakarta', 'ini', 'hari', 'konfirmasi', 'kemudian', 'itu', 'kpk', 'kami', 'download', 'wakil', 'pungutan', 'mungkin', 'dalam', 'war
tawan', 'napi-napi', 'untuk', 'detiknews', 'perihal', 'apps', 'merupakan', 'selesai', 'dan', 'periksa', 'https://news.detik.com/berita/d-7351748/kpk-kembali-periksa-azis-syamsuddin-terkait-pungli-rutan.',
'15', 'detikcom', 'pidana', 'terkait', 'mantan', 'kpk', 'pungli', 'kpk'}
```

PS C:\Users\Asus\Documents\UKDW\PrakAIPro\Set>