



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
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

Link Github: <https://github.com/Tohrubo/tugaspralpro12.git>

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Sets

Set adalah sebuah himpunan, mirip seperti List dan Tuple. Set memiliki sifat yang hanya menyimpan unique value, artinya value yang duplikat akan dihapus dan hanya 1 saja yang disimpan. Contoh set adalah seperti di bawah ini:

```
bilangan_genap = {2, 4, 6, 8, 10, 12}
bilangan_ganjil = {1, 3, 5, 7, 9, 11}
```

Sebuah set menggunakan {} untuk menandakan bahwa variabel menyimpan sebuah set. Tetapi karena dictionary juga menggunakan {}, maka untuk mendefinisikan sebuah set digunakan cara lain seperti:

```
pernah_ke_bulan = set('Neil Armstrong', 'Buzz Aldrin')
```

Sama seperti tipe data lain yang bisa didefinisikan dengan TIPE_DATA(), set juga bisa didefinisikan menggunakan set(). Sintaks ini juga bisa digunakan untuk membuat sebuah set kosong seperti di bawah ini:

```
pernah_ke_mars = set()
```

Karena yang digunakan {}, maka jika mencoba membuat set kosong hanya menggunakan {} untuk mendefinisikan set, hasilnya bukan menjadi set kosong tetapi menjadi dictionary kosong

```
x data = {} x
```

MATERI 2: Pengaksesan Set

Mengakses sebuah set sama seperti mengakses sebuah list dan tuple. Sebuah set yang berisi 3 value jika digunakan `len()` akan menampilkan jumlah 3. Contoh seperti di bawah ini:

```
nim = {'71200120', '71200195', '71200214'}
jumlah_nim = len(nim)
print(jumlah_nim)           # akan menghasilkan output 3
```

Untuk mengakses isi dari set dapat digunakan for loop untuk mengeluarkan isi dari sebuah set. Seperti contoh di bawah ini:

```
for n in nim:
    print(n)

71200214
71200195
71200120
```

Hasil output yang keluar dapat bervariasi urutannya, hal ini karena Set tidak memiliki angka indeks dan hanya menyimpan isi tanpa urutan. For loop yang digunakan menggunakan "in" juga karena set tidak mempunyai angka indeks, jadi for loop yang menggunakan range tidak bisa digunakan untuk mengeluarkan isi set dengan angka indeks.

Ada beberapa Set methods yang dapat diperhatikan, contohnya adalah:

- `.add()` → Digunakan untuk menambahkan sebuah value ke dalam set
- `.discard()` → Digunakan untuk menghapus sebuah value dari dalam set
- `.remove()` → Digunakan seperti `discard()`, tetapi akan error jika value yang ingin dihapus tidak ada di dalam set
- `.pop()` → Mengambil 1 value random dari dalam set dan menghapusnya. Akan error jika set adalah empty set
- `.clear()` → Menghapus semua elemen dari dalam set

Contoh penggunaan set methods `.add()` adalah sebagai berikut:

```
# definisikan sebuah set kosong
plat_nomor = set()

# tambahkan plat nomor 'AB 1890 XA'
plat_nomor.add('AB 1890 XA')

# tambahkan plat nomor 'AD 6810 MT'
plat_nomor.add('AD 6810 MT')

# jumlah anggota di dalam Set
print(len(plat_nomor))

# tambahkan plat yang sama sekali lagi
plat_nomor.add('AB 1890 XA')

# tampilkan semua plat nomor
for plat in plat_nomor:
    print(plat)
```

Output yang keluar akan seperti ini:

```
2
AD 6810 MT
AB 1890 XA
```

Contoh berikutnya:

```
bilangan_prima = {13, 23, 7, 29, 11, 5}
```

```
# hapus 5 dari set tersebut
```

```
bilangan_prima.remove(5)
```

```
print(bilangan_prima)
```

```
# hapus 97 (tidak ada)
```

```
bilangan_prima.discard(97)
```

```
print(bilangan_prima)
```

```
# ambil dan hapus salah satu
```

```
bilangan = bilangan_prima.pop()
```

```
print(bilangan)
```

```
print(bilangan_prima)
```

```
# kosongkan set
```

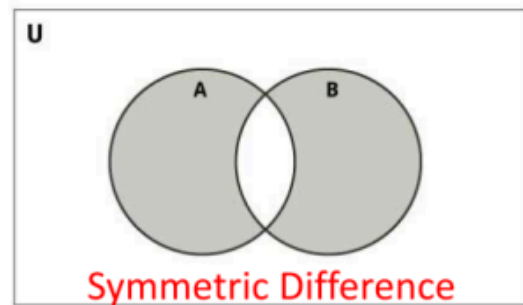
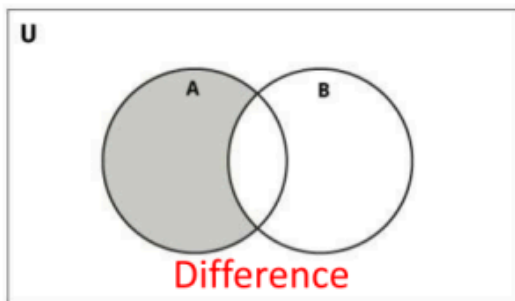
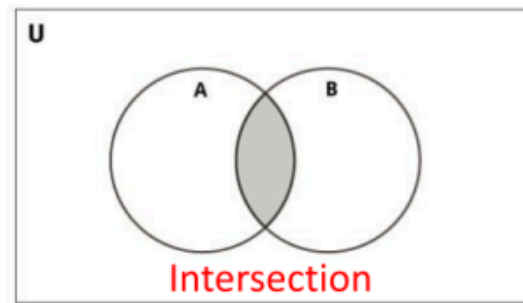
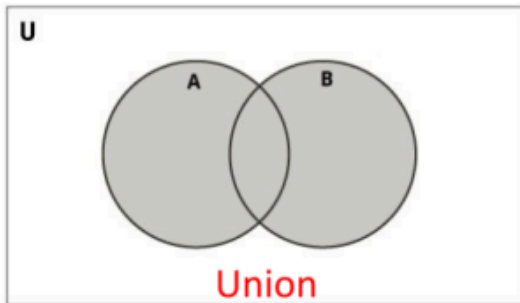
```
bilangan_prima.clear()
```

```
print(bilangan_prima)
```

Jika kode benar, maka output akan keluar seperti:

```
{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
```

MATERI 3: Set dan Operand



Set menggunakan logic juga untuk mengeluarkan hasil tertentu. Berikut adalah 4 logic yang paling berguna dan paling sering digunakan untuk mengeluarkan output tertentu:

1. Union → Untuk menggabungkan beberapa set menjadi satu.
Menggunakan operand " | "
2. Intersection → Untuk mencari value yang berada DI SEMUA set
Menggunakan operand " & "
3. Difference → Untuk mencari value yang ada di 1 set saja
Menggunakan operand " - "
4. Symmetric Difference → Untuk mencari value yang ada di kedua set NAMUN tidak ada di kedua set
Menggunakan operand " ^ "

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
Question1.py U × Question2.py U Question3.py U clone.txt U r
Question1.py > ...
1  n = int(input("Input Category amount: "))
2
3  categ = {}
4
5  for i in range(n):
6      cname = input("Input Category name: ")
7
8      apps = []
9      for j in range(5):
10         appname = input("App name: ")
11         apps.append(appname)
12
13         categ[cname] = apps
14
15  applst = []
16
17  for app in categ.values():
18      applst.append(set(app))
19
20  inres = applst[0]
21  for i in range(1, len(applst)):
22      inres = inres & applst[i]
23
24  print("Exist in all: ",inres)
25
26  dcount = {}
27  for key, value in categ.items():
28      for item in value:
29          dcount[item] = dcount.get(item, 0) + 1
30
31  symdiffres = set()
32  for key, value in dcount.items():
33      if value == 1:
34          symdiffres.add(key)
35
36  print("Exist in Only 1: ",symdiffres)
37
38  twoinres = set()
39  for key,value in dcount.items():
40      if value == 2:
41          twoinres.add(key)
42
43  print("Exist in only 2: ",twoinres)
```

B. Output Result

```
categ = {  
    "RPG" : ["Genshin", "Blue Archive", "FGO", "HSR", "WuWa"],  
    "Anime" : ["Genshin", "FGO", "DAL", "Trails", "Blue Archive"],  
    "Story" : ["Trails", "Ys", "Rance", "DAL", "FGO"],  
}
```

```
Exist in all: {'FGO'}  
Exist in Only 1: {'Rance', 'HSR', 'WuWa', 'Ys'}  
Exist in only 2: {'DAL', 'Blue Archive', 'Trails', 'Genshin'}
```

C. Explanation

Kode di atas menggunakan basis dari kode contoh 12.3. Tambahan yang diberikan adalah mencari aplikasi yang berada di hanya 1 kategori dan 2 kategori. Kode menggunakan for loop untuk mengambil key dan value dari dictionary categ dan setiap aplikasi akan dimasukan ke dictionary dcount untuk menghitung berapa kali aplikasi tersebut muncul di dictionary. Untuk yang muncul di 1 kategori hanya perlu mencari aplikasi yang muncul 1 kali saja. Untuk yang muncul di 2 kategori hanya perlu mencari aplikasi yang muncul 2 kali.

SOAL 2

A. Source Code

```
Question1.py U Question2.py U x Question3.py U clone.txt U
Question2.py > ...
1 lst = [1, 1, 2, 3, 4, 4, 5]
2 print(f"List = {lst}")
3
4 ls = set(lst)
5 print(f"List --> Set = {ls}")
6
7 lsl = list(ls)
8 print(f"List --> Set --> List = {lsl}")
9
10 tup = (1, 2, 3, 3, 3, 4, 5)
11 print(f"Tuple = {tup}")
12
13 ts = set(tup)
14 print(f"Tuple --> Set = {ts}")
15
16 tst = tuple(ts)
17 print(f"Tuple --> Set --> Tuple = {tst}")
```

B. Output Result

```
List = [1, 1, 2, 3, 4, 4, 5]
List --> Set = {1, 2, 3, 4, 5}
List --> Set --> List = [1, 2, 3, 4, 5]
Tuple = (1, 2, 3, 3, 3, 4, 5)
Tuple --> Set = {1, 2, 3, 4, 5}
Tuple --> Set --> Tuple = (1, 2, 3, 4, 5)
```

C. Explanation

Kode di atas hanya menggunakan konversi set untuk mengubah list dan tuple ke set dan *vice versa*. List dan tuple akan menyimpan semua value, Set hanya menyimpan unique value (Duplikat akan dihapus). Ketika diubah dari list/tuple ke set kembali menjadi list/tuple, ada value yang hilang karena set hanya menyimpan unique value, maka duplikat yang hilang tidak bisa dikembalikan.

SOAL 3

A. Source Code

```
Question1.py U Question2.py U Question3.py U × clone.txt U
Question3.py > ...
1 file1 = input("Insert Name for File 1: ")
2 file2 = input("Insert Name for File 2: ")
3
4 try:
5     handle1 = open(file1, "r")
6     handle2 = open(file2, "r")
7 except:
8     print("Cannot find file")
9     exit()
10
11 wf1s = set()
12 for lines1 in handle1:
13     sen1 = lines1.split()
14     for word1 in sen1:
15         lw1 = word1.lower()
16         wf1s.add(lw1)
17 wf2s = set()
18 for lines2 in handle2:
19     sen2 = lines2.split()
20     for word2 in sen2:
21         lw2 = word2.lower()
22         wf2s.add(lw2)
23
24 wordint = wf1s & wf2s
25 print(wordint)
```

B. Output Result

```
≡ clone.txt
1 Window breaks what happened
2 Juliet is the sun no she is not
3 Arise sun and kill what why
4 Who is sick pale grief miserable
```

```
≡ romeo.txt
1 But soft what light through yonder window breaks
2 It is the east and Juliet is the sun
3 Arise fair sun and kill the envious moon
4 Who is already sick and pale with grief
```

```
{'what', 'the', 'grief', 'and', 'window', 'kill', 'is', 'who', 'sick', 'arise', 'juliet', 'sun', 'breaks', 'pale'}
```

C. Explanation

Kode di atas menggunakan open dan 2 set untuk menemukan kata kata yang muncul di kedua file txt. User akan memasukkan nama 2 file txt, kemudian kedua file akan di open dengan handle1 dan handle2. Try except digunakan untuk mengatasi user input error. Kemudian ada 2 for loop untuk setiap file yang akan mencatat semua kata yang sudah di lowercase yang ada di dalam file dalam bentuk set. Karena dalam bentuk set, maka kata duplikat akan dihapus. Kemudian dicari intersection (AND) dari kedua set tersebut yang kemudian ditampilkan hasilnya.