Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230997**
Nama Lengkap	**Christ Jevicto Ajimas Kirana**
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%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

1.1 Pengenalan dan Mendefinisikan Set

Set atau bisa disebut juga dengan istilah himpunan, yang digunakan untuk menyimpan sekumpulan data yang isi dari set tersebut di anggap unik. Beberapa sifat set adalah:

- Isi dari set disebut anggota (member).
- Anggota dari set bersifat immutable(intger, float, string, tuple, dll). Dengan demikian list dan dictionary yang bersifat mutable tidak bisa dimasukan kedalam set.
- Set bersifat mutable, yang berarti kita dapat menambah dan mengurangi isi set, oleh karena itu set tidak bisa dimasukan kedalam set.

Untuk mendefinisikan set ada beberapa cara seperti notasi {} dan fungsi set() contoh =

```
bilangan_genap = {2, 4, 6, 8, 10, 12}
bilangan_ganjil = {1, 3, 5, 7, 9, 11}
pernah_ke_bulan = set('Neil Armstrong', 'Buzz Aldrin')

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

Untuk mendefinisikan Set kosong kita tidak dapat menggunakan notasi {}, tapi harus menggunakan fungsi set(). Jika menggunakan notasi {} maka akan menjadi dictionary kosong

1.2 Pengaksesan Set

Set tidak memiliki indeks, makanya kita tidak bisa mengakses isi elemen set secara langsung, contoh:

```
1  nim = {'71200120', '71200195', '71200214'}
2  jumlah_nim = len(nim)
3  print(jumlah_nim)
4
5  for n in nim:
6      print(n)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\PRALPRO\71230997_minggu13> & C:/Users/chris/Ap
3
71200120
71200195
71200214
```

output yang dihasilkan urutannya berbeda, hal ini karena set tidak memiliki indeks, maka posisi tiap anggota dalam set tidak penting.

Set adalah mutable, artinya isinya bisa bertambah atau berkurang. Conoth menambahkan anggota pada sebuah set dengan fungsi **add()**:

```
1   plat_nomor = set()
2   plat_nomor.add('AB 1890 XA')
3   plat_nomor.add('AD 6810 MT')
4   print(len(plat_nomor))
5   plat_nomor.add('AB 8312 XA')
6   plat_nomor.add('AD 6810 MT')
7   for plat in plat_nomor:
8       print(plat)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERM

PS D:\PRALPRO\71230997_minggu13> & C:/
2
AD 6810 MT
AB 1890 XA
AB 8312 XA
```

Set memiliki mekanisme untuk memeriksa duplikasi saat menambahkan anggota baru. Jika anggota baru belum ada dalam Set, anggota tersebut akan ditambahkan. Jika sudah ada, pemanggilan fungsi add() tidak akan menambah anggota tersebut. Pengecekan ini otomatis dilakukan oleh fungsi add(), jadi Anda tidak perlu melakukannya sendiri.

Kita bisa menghapus isi dari set dengan beberapa cara yaitu dengan fungsi **discard()**, **remove()**, **pop()**, dan **clear().** Perbedaan dari fungsi[2] tersebut ialah:

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

Contoh pemakain fungsi[2] untuk menghapus isi set:

```
1   bilangan_prima = {13, 23, 7, 29, 11, 5}
2   # hapus 5 dari set tersebut
3   bilangan_prima.remove(5)
4   print(bilangan_prima)
5   # hapus 97 (tidak ada)
6   bilangan_prima.discard(97)
7   print(bilangan_prima)
8   # ambil dan hapus salah satu
9   bilangan = bilangan_prima.pop()
10  print(bilangan)
11  print(bilangan_prima)
12  # kosongkan set
13  bilangan_prima.clear()
14  print(bilangan_prima)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\PRALPRO\71230997_minggu13> & C:/Users/chris/
{29, 23, 7, 11, 13}
{29, 23, 7, 11, 13}
29
{23, 7, 11, 13}
set()
```

Fungsi discard() tidak menghasilkan error jika anggota yang ingin dihapus tidak ada dalam Set. Fungsi pop() akan mengeluarkan salah satu anggota secara acak dari Set dan berguna untuk memproses isi Set satu per satu tanpa memperhatikan urutan. Untuk mengubah nilai salah satu anggota dalam Set, Anda tidak bisa mengubahnya langsung. Anda harus menghapus anggota yang ingin diubah dan menambahkan anggota baru dengan nilai yang diinginkan. Contoh :

```
1   # Buat Set dari List
2   ikan = set(['koi', 'koki', 'kembung', 'salmon'])
3   print(ikan)
4   # ganti koi menjadi teri
5   ikan.remove('koi')
6   ikan.add('teri')
7   print(ikan)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTE

PS D:\PRALPRO\71230997_minggu13> & C:/Users/chris/AppData/L
{'salmon', 'kembung', 'koki', 'koi'}
{'kembung', 'koki', 'teri', 'salmon'}
```

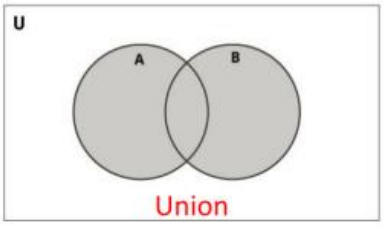
Untuk mengubah nilai 'koi' menjadi 'teri' dalam Set, hapus 'koi' lalu tambahkan 'teri'. Set tidak memiliki urutan data, sehingga setelah operasi ini, 'koi' tidak ada lagi dan digantikan oleh 'teri'. Urutan anggota dalam Set dapat berubah setiap kali ada penambahan atau penghapusan.

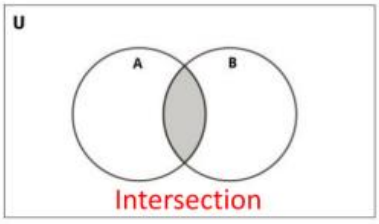
1.3 Operasi-Operasi pada Set

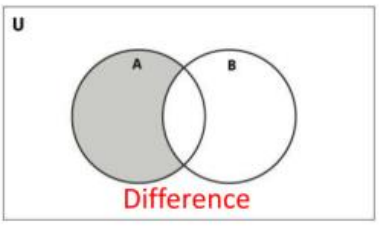
Ada beberapa opreasi pada set di Python yaitu:

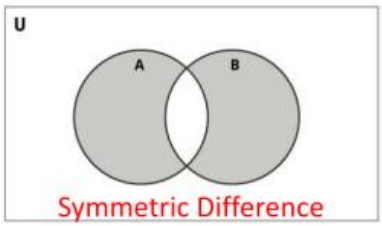
- Union: Menggabungkan dua Set menjadi satu dengan operator | atau fungsi union().
- Intersection: Menghasilkan irisan dua Set dengan operator & atau fungsi intersection().
- Difference: Menghasilkan selisih dua Set dengan operator - atau fungsi difference().
- Symmetric Difference: Menghasilkan Set baru dari dua Set kecuali irisannya dengan operator ^ atau fungsi symmetric_difference().

Ilustrasi :









Operator Union

Contoh :

```
1   merek_hp = {'Samsung', 'Apple', 'Xiaomi', 'Sony'}
2   merek_ac = {'LG', 'Samsung', 'Panasonic', 'Daikin', 'Sony'}
3   gabungan = merek_hp | merek_ac
4   print (gabungan)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER  COMMENTS

PS D:\PRALPRO\71230997_minggu13> & C:/Users/chris/AppData/Local/Program
{'Xiaomi', 'Daikin', 'LG', 'Sony', 'Apple', 'Panasonic', 'Samsung'}
```

Output dari program tersebut adalah penggabungan Set merek_hp dengan Set merek_ac, menghasilkan Set baru bernama gabungan. Anggota yang sama, seperti 'Samsung' dan 'Sony', hanya muncul sekali karena Set tidak mengizinkan duplikasi.

Operator Intersection

Contoh :

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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER

PS D:\PRALPRO\71230997_minggu13> & C:/Users/chris/AppData/Local
{'upin', 'ipin', 'mail'}
```

Program tersebut menghasilkan output dari operasi intersection antara Set renang dan Set tenis, yaitu anggota-anggota yang ada di kedua Set: mail, upin, dan ipin.

Operator Difference

Contoh :

```
1   english = {'desi', 'tono', 'evan', 'miko', 'takashi', 'chaewon'}
2   korean = {'chaewon', 'yeona', 'erika', 'miko'}
3   only_korean = korean - english
4   print(only_korean)
5   only_english = english - korean
6   print(only_english)
```

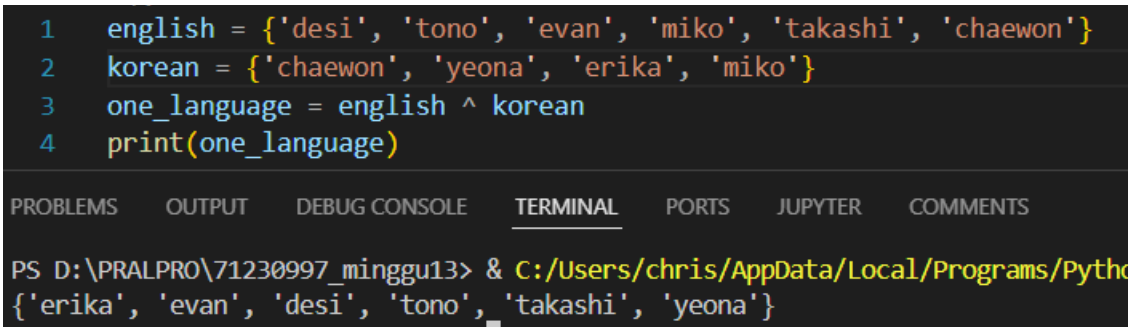
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER  COMMENTS

PS D:\PRALPRO\71230997_minggu13> & C:/Users/chris/AppData/Local/Programs/Pyt
{'erika', 'yeona'}
{'evan', 'tono', 'takashi', 'desi'}
```

Operator difference menghasilkan Set berisi anggota yang merupakan selisih dari dua Set. Pada contoh, digunakan untuk menemukan anggota yang hanya bisa berbahasa Korea dengan mencari selisih antara Set korean dan Set english. Sebaliknya, untuk mengetahui siapa yang hanya bisa berbahasa Inggris, cari selisih antara Set english dan Set korean.

Operator Symmetric Difference

Contoh :



Operator symmetric difference menghasilkan set baru yang berisi elemen-elemen unik yang hanya ada di salah satu dari dua set awal, tidak termasuk elemen-elemen yang ada di kedua set tersebut. Misalnya, dalam kasus bahasa, operator ini memungkinkan identifikasi orang yang hanya bisa berbicara dalam satu bahasa, tanpa memasukkan mereka yang bisa berbicara dalam dua bahasa.

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

```python
n = int(input('Masukkan jumlah kategori: '))

data_aplikasi = {}

for i in range(n):
    nama_kategori = input(f"Masukkan nama kategori ke-{i+1}: ")
    aplikasi = []
    print(f"Masukkan 5 nama aplikasi untuk kategori '{nama_kategori}':")
    for j in range(5):
        nama_aplikasi = input(f"Aplikasi ke-{j+1}: ")
        aplikasi.append(nama_aplikasi)
    data_aplikasi[nama_kategori] = aplikasi

print("\nDaftar nama aplikasi di setiap kategori:")
for kategori, aplikasi in data_aplikasi.items():
    print(f"{kategori}: {', '.join(aplikasi)}")

daftar_aplikasi_list = []

for aplikasi in data_aplikasi.values():
    daftar_aplikasi_list.append(set(aplikasi))


hasil = daftar_aplikasi_list[0]
for i in range(1, len(daftar_aplikasi_list)):
    hasil = hasil&daftar_aplikasi_list[i]
print(f"aplikasi yang muncul disemua kategori: {hasil}")

hasill = daftar_aplikasi_list[0]
for i in range(1, len(daftar_aplikasi_list)):
    hasill = hasill^daftar_aplikasi_list[i]
```

```
anu = daftar_aplikasi_list[0]
for i in range(1,len(daftar_aplikasi_list)):
    anu.update(anu&daftar_aplikasi_list[i])
for i in anu:
    if i in hasill:
        hasill.remove(i)
print(hasill)

print("Aplikasi yang hanya muncul di satu kategori saja:", hasill)

apl = list()
dua_kategori=set()

if n > 2:
    for i in data_aplikasi.values():
        apl+=i
    for i in apl:
        if apl.count(i)==2:
            dua_kategori.add(i)
    print("Aplikasi yang muncul tepat di dua kategori:", dua_kategori)
```

Input Jumlah Kategori:

Program meminta pengguna untuk memasukkan jumlah kategori.

Input Data Kategori dan Aplikasi:

Untuk setiap kategori, pengguna diminta memasukkan nama kategori dan lima aplikasi yang terkait. Data disimpan dalam struktur dictionary.

Tampilkan Aplikasi per Kategori:

Program mencetak daftar nama aplikasi untuk setiap kategori yang telah diinput oleh pengguna.

Temukan Aplikasi di Semua Kategori:

Program menggunakan operasi **intersection** pada set aplikasi untuk menemukan aplikasi yang muncul di semua kategori.

Temukan Aplikasi yang Hanya Muncul di Satu Kategori:

Program menggunakan operasi **XOR** pada set aplikasi untuk menemukan aplikasi yang unik di setiap kategori. Aplikasi yang muncul lebih dari satu kali dihapus dari hasil unik tersebut.

Temukan Aplikasi yang Muncul di Tepat Dua Kategori (Jika n > 2):

Program menghitung kemunculan setiap aplikasi di seluruh kategori.Aplikasi yang muncul tepat dua kali disimpan dan ditampilkan.

SOAL 2

```python
def demo():
    l = [1, 2, 2, 3, 4, 4, 5]
    print(f'list sebelum di konversi : {l}')
    s = set(l)
    print(f'list setelah di konversi menjadi set: {s}')
    print(f'set sebelum di konversi : {s}')
    l2 = set(s)
    print(f'set setelah di konversi menjadi list : {l2}')

    t = [1, 2, 2, 3, 4, 4, 5]
    print(f"tuple sebelum di konversi : {t}")
    s2 = set(t)
    print(f"tuple setelah di konversi menjadi set: {s2}")
    print(f'set sebelum di konversi : {s2}')
    t2 = tuple(s2)
    print(f"tuple setelah di konversi menjadi tuple : {t2}")

demo()
```

Konversi List ke Set dan Kembali ke List:

Dimulai dengan sebuah list yang berisi elemen-elemen, termasuk duplikat. List tersebut dikonversi menjadi set untuk menghilangkan elemen duplikat. Set kemudian dikonversi kembali menjadi list.

Konversi Tuple ke Set dan Kembali ke Tuple:

Dimulai dengan sebuah tuple yang berisi elemen-elemen, termasuk duplikat. Tuple tersebut dikonversi menjadi set untuk menghilangkan elemen duplikat. Set kemudian dikonversi kembali menjadi tuple.

SOAL 3

```
tabnine: test | explain | document | ask
def membacafile(namafile):
    try:
        with open(namafile, 'r', encoding='UTF-8') as file:
            return file.read().lower().split()
    except FileNotFoundError:
        print(f"file {namafile} tidak ditemukan")
        return[]

tabnine: test | explain | document | ask
def main():
    file1 = input('masukan file 1 = ')
    file2 = input('masukan file 2 = ')
    baca_file1 = membacafile(file1)
    baca_file2 = membacafile(file2)
    kata_sama = set(baca_file1) & (set(baca_file2))

    if kata_sama:
        print ("kata sama adalah: ")
        for i in sorted(kata_sama):
            print(i)
    else:
        ('tidak ada kata yang sama')

if __name__ == '__main__':
    main()
```

1. **Fungsi membacafile**:

   Mencoba membuka file yang diberikan dengan nama file (**namafile**). Membaca isi file, mengubahnya menjadi huruf kecil, dan memecahnya menjadi daftar kata. Jika file tidak ditemukan, mencetak pesan kesalahan dan mengembalikan daftar kosong.

2. **Fungsi main**:

   Meminta pengguna untuk memasukkan nama dua file. Membaca isi kedua file menggunakan fungsi **membacafile**. Mengonversi daftar kata dari kedua file menjadi set, lalu mencari irisan (kata yang sama) dari kedua set tersebut. Jika ada kata yang sama, menampilkan kata-kata tersebut secara berurutan. Jika tidak ada kata yang sama, mencetak pesan bahwa tidak ada kata yang sama.

LINK GITHUB = https://github.com/Yeeemeki/praktikum-algoritma-dan-pemograman.git