Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230978**
Nama Lengkap	**Jonathan Satriani Gracio Andrianto**
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%)

Pengenalan dan Mendefinisikan Set

Set adalah salah satu tipe data dalam Python yang digunakan untuk menyimpan sekumpulan data yang semuanya bersifat unik. Tipe data set memiliki sifat yang sama seperti himpunan, itulah kenapa tipe data ini juga dikenal dengan sebutan himpunan. Berikut karakteristik dari tipe data set :

- Isi dari set disebut anggota atau member
- Anggota dari set harus bersifat immutable, sehingga tipe data mutable seperti list dan dictionary tidak dapat dimasukkan ke dalam set.
- Set bersifat mutable, berarti kita dapat menambah atau menghapus elemen dari sebuah set. Dengan logika yang sama dari poin kedua berarti set tidak dapat dimasukkan ke dalam set lainnya.

Kita dapat mendefinisikan set dengan menggunakan notasi {} dan fungsi set() seperti contoh berikut ini:

```
#dengan notasi {}
bilangan = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
#dengan set()
namague = set("Jonathan", "Andrianto")
```

Untuk mendefinisikan set kosong :

```
#harus dengan set()
set1 = set() #menghasilkan set kosong
data = {} #menghasilkan dictionary kosong
```

Pengaksesan Set

Karena set tidak memiliki index, maka kita mengaksesnya dengan cara :

```
nama = {"jo", "noel"}
jumlah_nama = len(nama)
print(jumlah_nama) # akan menghasilkan output 2
# tampilkan isi set satu-persatu
for i in nama:
    print(i)

'''
output
2
noel
jo
'''
```

Jika diperhatikan, urutan output yang dihasilkan berbeda dengan urutan deklarasi set. Hal ini terjadi karena set tidak memiliki indeks, sehingga anggotanya tidak memiliki urutan tertentu. Dalam tipe data set, posisi anggota tidaklah penting.

Karena set mutable, artinya isinya dapat ditambah. Cara menambah anggota ke dalam set dengan fungsi add() :

```
# definisikan sebuah set kosong
nama = set()
# tambahkan plat nomor 'AB 1890 XA'
nama.add('Jo')
# tambahkan plat nomor 'AD 6810 MT'
nama.add('Noel')
# jumlah anggota di dalam Set
print(len(nama))
# tambahkan plat yang sama sekali lagi
nama.add('Jo')
# tampilkan semua plat nomor
for i in nama :
    print(i)
'''
output
2
Jo
Noel
'''
```

Set memiliki mekanisme untuk memeriksa apakah elemen baru yang akan ditambahkan sudah ada di dalam Set (cek duplikasi). Jika elemen tersebut belum ada, maka elemen tersebut dapat ditambahkan ke dalam Set. Namun, jika elemen dengan nilai yang sama sudah ada di dalam Set, pemanggilan fungsi add() tidak akan menambahkan elemen tersebut lagi. Pengecekan duplikasi ini sudah terintegrasi dalam fungsi add(), sehingga kita tidak perlu melakukannya secara manual.

Untuk menghapus anggota dari sebuah set, beberapa cara yang dapat digunakkan yaitu dengan fungsi discard(), remove(), pop() dan clear(). Perbedaan dari empat fungsi tersebut dapat dilihat pada Gambar 1.1.

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

1.1 Fungsi-fungsi untuk menghapus anggota dari Set.

Berikut program penghapusan anggota set :

```
bilangan_prima = {13, 23, 7, 29, 11, 5}
# hapus 5 dari set tersebut
bilangan_prima.remove(5)
print(bilangan_prima) #{7, 11, 13, 23, 29} setelah 5 dihapus. Perhatikan urutan
berubah
# hapus 97 (tidak ada)
bilangan_prima.discard(97)
print(bilangan_prima) #{7, 11, 13, 23, 29} 97 tidak ada, sehingga Set tidak
berubah
# ambil dan hapus salah satu
bilangan = bilangan_prima.pop()
print(bilangan) #7 fungsi pop() mengeluarkan 7
print(bilangan_prima) #{11, 13, 23, 29} setelah 7 keluar dari set
# kosongkan set
bilangan_prima.clear()
print(bilangan_prima) #set() setelah isi dari Set dihapus semua
```

Fungsi discard() tidak akan menghasilkan error jika elemen yang ingin dihapus tidak ada di dalam Set. Sebaliknya, fungsi pop() akan menghapus dan mengembalikan salah satu elemen dari set secara acak. Fungsi pop() berguna jika kita ingin memproses anggota-anggota dari set satu per satu tanpa memperhatikan urutan atau posisi setiap elemen di dalam set.

Pada Set kita tidak bisa mengubah langsung nilai dari salah satu anggota di dalam Set. Jika ingin mengganti nilai anggota set maka kita harus melakukan operasi penggantian (replace) dengan cara menghapus anggota yang mau diubah, kemudian memasukkan anggota baru yang kita inginkan. Contohnya :

```
# Buat Set dari List
goats = set(['ronaldo', 'pele', 'cristiano', 'messi'])
```

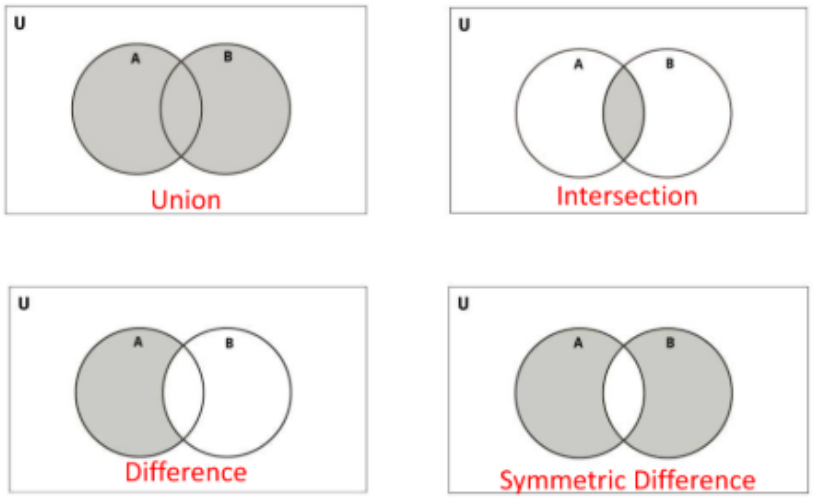
```
print(goats) #output = {'ronaldo', 'messi', 'pele', 'cristiano'}
# ganti ronaldo(botak) menjadi maradona
goats.remove('ronaldo')
goats.add('maradona')
print(goats) #output = {'pele', 'maradona', 'messi', 'cristiano'}
```

Operasi-Operasi pada Set

Operasi-operasi pada set sama dengan operasi-operasi pada himpunan. Berikut operasi-operasi pada tipe data set :

- **Operator Union** : Untuk menggabungkan dua set menjadi satu. Dapat menggunakan operator '|' atau fungsi union().
- **Operator Intersection** : Untuk mencari irisan dari dua set. Dapat menggunakan operator '&' atau fungsi intersection().
- **Operator Difference** : Untuk menghasilkan set baru yang merupkan selisih dari dua set. Dapat menggunakan operator '-' atau fungsi difference()
- **Operator Symmetric Difference** : Untuk menghasilkan set baru yang merupakan jumlah dari dua set kecuali irisannya. Dapat menggunakan operator '^' maupun fungsi symmetric_difference().

Ilustrasi dari operator-operator tipe data set dapat dilihat pada gambar 1.2 :



1.2 Operasi-operasi pada Set.

OPERATOR UNION

Contoh :

```
pemain_madrid = {'Figo', 'Modric', 'Cristiano'}
pemain_barca = {'Messi', 'Figo', 'Dani Alves'}
gabungan = pemain_madrid | pemain_barca
print(gabungan) #output = {'Messi', 'Figo', 'Modric', 'Dani Alves', 'Cristiano'}
```

‘Figo’ hanya muncul sekali karena anggota dari set harus unik.

OPERATOR INTERSECTION

Contoh :

```
game = {'Jo', 'Alven', 'Efrant'}
belajar = {'Nevan', 'Daniel', 'Jo'}
print(game & belajar) #output = {'Jo'}
```

Intersection berarti anggota-anggota yang berada di dalam set game dan set belajar sekaligus, yaitu Jo.

OPERATOR DIFFERENCE

Contoh :

```
indo = {'Jo', 'Nevan', 'Hosea', 'Alven', 'Daniel'}
china = {'Hosea', 'Vincent', 'Alven', 'Dex'}
#yang hanya bisa bahasa cina
only_china = china - indo
print(only_china) #output = {'Vincent', 'Dex'}
#yang hanya bisa bahasa indo
only_indo = indo - china
print(only_indo) #output = {'Nevan', 'Jo', 'Daniel'}
```

Operator difference akan menghasilkan Set yang anggotanya merupakan selisih dari kedua Set yang dibandingkan

OPERATOR SYMMETRIC DIFFERENCE

Contoh :

```
indo = {'Jo', 'Nevan', 'Hosea', 'Alven', 'Daniel'}
china = {'Hosea', 'Vincent', 'Alven', 'Dex'}
only_one_languange = indo ^ china
print(only_one_languange) #{'Vincent', 'Jo', 'Nevan', 'Daniel', 'Dex'}
```

Operator symmetric difference akan menghasilkan Set baru yang merupakan gabungan dari dua Set tetapi tidak termasuk irisannya.

Kegiatan Praktikum

Kasus 1. Kategori aplikasi di Play Store

Buatlah program yang meminta input n kategori aplikasi, lalu program meminta pengguna untuk memasukkan nama-nama aplikasi sebanyak 5 untuk masing-masing kategori. Setelah pengguna selesai memasukkan semua nama aplikasi, program akan menampilkan:

- Daftar nama aplikasi di setiap kategori
- Program yang muncul di semua kategori

Untuk membuat program ini, langkah-langkah yang perlu diimplementasikan adalah sebagai berikut:

- Minta jumlah kategori (n).
- Minta nama kategori, kemudian minta nama-nama aplikasi sebanyak 5 untuk kategori tersebut
- Setelah semua kategori selesai di-input, berikutnya tampilkan daftar nama aplikasi di setiap kategori
- Dengan operasi intersection, tampilkan nama aplikasi yang muncul di semua kategori yang ada.

Untuk permasalahan ini, kita akan menggunakan bantuan List, Dictionary dan Set. Solusi untuk masalah tersebut dapat dilihat pada program berikut ini:

```
# input n kategori
n = int(input('Masukkan jumlah kategori: '))
# siapkan dictionary kosong
data_aplikasi = {}
# input nama kategori dan aplikasi di dalamnya
for i in range(n):
    nama_kategori = input('Masukkan nama kategori:')
    print('Masukkan 5 nama aplikasi di kategori', nama_kategori)
# siapkan list kosong untuk nama-nama aplikasi
    aplikasi = []
    for j in range(5):
        nama_aplikasi = input('Nama aplikasi: ')
        aplikasi.append(nama_aplikasi)
    # masukkan dalam dictionary
    data_aplikasi[nama_kategori] = aplikasi
# tampilkan dictionary data_aplikasi
print(data_aplikasi)
daftar_aplikasi_list = []
# ambil semua daftar aplikasi dari setiap kategori
for aplikasi in data_aplikasi.values():
    daftar_aplikasi_list.append(set(aplikasi))
print(daftar_aplikasi_list)
# lakukan intersection ke semua set yang ada
hasil = daftar_aplikasi_list[0]
for i in range(1, len(daftar_aplikasi_list)):
    hasil = hasil.intersection(daftar_aplikasi_list[i])
```

```
print(hasil)
```

Outputnya :

```
Masukkan jumlah kategori: 2
Masukkan nama kategori:finance
Masukkan 5 nama aplikasi di kategori finance
Nama aplikasi: rti saham
Nama aplikasi: mirae
Nama aplikasi: ipot
Nama aplikasi: poems
Nama aplikasi: kalkulator
Masukkan nama kategori:utilities
Masukkan 5 nama aplikasi di kategori utilities
Nama aplikasi: photos
Nama aplikasi: weather
Nama aplikasi: kamera
Nama aplikasi: kalkulator
Nama aplikasi: notes
{'finance': ['rti saham', 'mirae', 'ipot', 'poems', 'kalkulator'], 'utilities': ['photos', 'weather', 'kamera', 'kalkulator', '
notes']}
[{'mirae', 'ipot', 'poems', 'rti saham', 'kalkulator'}, {'notes', 'weather', 'photos', 'kamera', 'kalkulator'}]
{'kalkulator'}
```

Dari kedua kategori tersebut, yang muncul pada keduanya sekaligus adalah aplikasi kalkulator.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

Source Code

```
# input n kategori
n = int(input('Masukkan jumlah kategori: '))
# siapkan dictionary kosong
data_aplikasi = {}
# input nama kategori dan aplikasi di dalamnya
for i in range(n):
    nama_kategori = input('Masukkan nama kategori:')
    print('Masukkan 5 nama aplikasi di kategori', nama_kategori)
# siapkan list kosong untuk nama-nama aplikasi
    aplikasi = []
    for j in range(5):
        nama_aplikasi = input('Nama aplikasi: ')
        aplikasi.append(nama_aplikasi)
    # masukkan dalam dictionary
    data_aplikasi[nama_kategori] = aplikasi
# tampilkan dictionary data_aplikasi
print(data_aplikasi)
daftar_aplikasi_list = []
# ambil semua daftar aplikasi dari setiap kategori
for aplikasi in data_aplikasi.values():
    daftar_aplikasi_list.append(set(aplikasi))
print(daftar_aplikasi_list)
```

```
hasil = daftar_aplikasi_list[0]
for i in range(1, len(daftar_aplikasi_list)) :
    hasil = hasil.intersection(daftar_aplikasi_list[i])
print(f"yang muncul disemuanya : {hasil}")

kamus_apk = dict()
for apk_set in daftar_aplikasi_list :
    for apk in apk_set :
        if apk not in kamus_apk :
            kamus_apk[apk] = 1
        elif apk in kamus_apk:
            kamus_apk[apk] += 1
hasil2 = set()
for key, value in kamus_apk.items() :
        if value == 1 :
            hasil2.add(key)
print(f"yang muncul hanya sekali : {hasil2}")

if n > 2 :
    hasil3 = set()
    for key, value in kamus_apk.items() :
        if value == 2 :
            hasil3.add(key)
    print(f"yang muncul tepat dua kali : {hasil3}")
```

Yang berubah :

```
kamus_apk = dict()
for apk_set in daftar_aplikasi_list :
    for apk in apk_set :
        if apk not in kamus_apk :
            kamus_apk[apk] = 1
        elif apk in kamus_apk:
            kamus_apk[apk] += 1
hasil2 = set()
for key, value in kamus_apk.items() :
        if value == 1 :
            hasil2.add(key)
print(f"yang muncul hanya sekali : {hasil2}")

if n > 2 :
    hasil3 = set()
    for key, value in kamus_apk.items() :
        if value == 2 :
```

```
            hasil3.add(key)
    print(f"yang muncul tepat dua kali : {hasil3}")
```

Output

```
Masukkan jumlah kategori: 3
Masukkan nama kategori:pertama
Masukkan 5 nama aplikasi di kategori pertama
Nama aplikasi: a
Nama aplikasi: b
Nama aplikasi: c
Nama aplikasi: d
Nama aplikasi: e
Masukkan nama kategori:kedua
Masukkan 5 nama aplikasi di kategori kedua
Nama aplikasi: a
Nama aplikasi: b
Nama aplikasi: f
Nama aplikasi: g
Nama aplikasi: h
Masukkan nama kategori:ketiga
Masukkan 5 nama aplikasi di kategori ketiga
Nama aplikasi: a
Nama aplikasi: f
Nama aplikasi: i
Nama aplikasi: j
Nama aplikasi: k
```

```
{'pertama': ['a', 'b', 'c', 'd', 'e'], 'kedua': ['a', 'b', 'f', 'g', 'h'], 'ketiga': ['a', 'f', 'i', 'j', 'k']}
[{'c', 'b', 'e', 'a', 'd'}, {'g', 'h', 'b', 'f', 'a'}, {'i', 'k', 'f', 'j', 'a'}]
yang muncul disemuanya : {'a'}
yang muncul hanya sekali : {'g', 'c', 'h', 'i', 'k', 'e', 'j', 'd'}
yang muncul tepat dua kali : {'b', 'f'}
```

Penjelasan

pertama kita buat dictionary baru yang akan memuat nama aplikasi serta berapa kali kemunculannya. Kemudian kuta buat perulangan untuk memasukan nilai dari daftar aplikasi ke dalam dictionary yang baru kita buat. Kemudian kita tinggal mengecek apakah aplikasinya muncul di satu kategori saja dan apakah aplikasinya muncul di dua kategori sekaligus.

SOAL 2

Source Code

```
def data_diri(nim, nama, alamat) :
def list_to_set(list1) :
    print(f"list awal = {list1}")
    hasil = set(list1)
    print(f"hasil konversi ke set = {hasil}")

def set_to_list(set1) :
    print(f"set awal = {set1}")
    hasil = list(set1)
    print(f"hasil konversi ke list = {hasil}")

def tuple_to_set(tuple1) :
    print(f"tuple awal = {tuple1}")
    hasil = set(tuple1)
    print(f"hasil konversi ke set = {hasil}")

def set_to_tuple(set2) :
    print(f"set awal = {set2}")
    hasil = tuple(set2)
    print(f"hasil konversi ke tuple = {hasil}")

list1 = [1, 2, 3]
list_to_set(list1)
set1 = {4, 5, 6}
set_to_list(set1)
tuple1 = (7, 8, 9)
tuple_to_set(tuple1)
set2 = {10, 11, 12}
set_to_tuple(set2)
```

Output

```
list awal = [1, 2, 3]
hasil konversi ke set = {1, 2, 3}
set awal = {4, 5, 6}
hasil konversi ke list = [4, 5, 6]
tuple awal = (7, 8, 9)
hasil konversi ke set = {8, 9, 7}
set awal = {10, 11, 12}
hasil konversi ke tuple = (10, 11, 12)
```

Penjelasan

Pertama saya membuat fungsi fungsinya terlebih dahulu agar memudahkan saya dalam melihat konversinya. Kemudian saya gunakan fungsi set() untuk mengubah menjadi tipe data set, fungsi list() untuk mengubah menjadi tipe data list, dan fungsi tuple() untuk mengubah menjadi tipe data tuple.

SOAL 3

Souce Code

```python
def teks_file(filename):
    try:
        with open(filename, 'r') as file:
            teks = file.read().lower()
            return set(teks.split())
    except FileNotFoundError:
        print(f"{filename}' tidak ditemukan")
        return set()

file1 = input("masukan nama file 1 :")
file2 = input("masukan nama file 2 :")
teks1 = teks_file(file1)
teks2 = teks_file(file2)

teks_sama = teks1 & teks2

if len(teks_sama) == 0 :
    print("tidak ada kata yang muncul pada kedua file")
else :
    print(f"kata-kata yang muncul di kedua teks adalah : {teks_sama}")
```

Output

```
masukan nama file 1 :uno.txt
masukan nama file 2 :dos.txt
kata-kata yang muncul di kedua teks adalah : {'teman', 'nama', 'halo', 'saya', 'andrianto'}
PS D:\kuliah\SEM 2\PR. ALPRO\PERTEMUAN 13> & C:/Users/ASUS/AppData/Local/Programs/Python/Pyt
PR. ALPRO/PERTEMUAN 13/nomor3.py"
masukan nama file 1 :wlee.txt
masukan nama file 2 :uno.txt
tidak ada kata yang muncul pada kedua file
```

Penjelasan

Pertama kita buat fungsi untuk membuka file teks dan memasukkan isi teks tersebut menjadi anggota set. Kemudian kita bandingkan kedua et isi teks file tersebut dan kita gunakkan operator ‘&’ untuk mencari kata yang sama.

Link Github

https://github.com/JonathanAndrianto123/laporan_alpro12.git