



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	71220829
Nama Lengkap	TIMOTIUS EKANA TEJO KESUMA
Minggu ke / Materi	14 / Tipe Data Set

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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 12.3.1

Set merupakan salah satu dari empat struktur data koleksi utama dalam bahasa pemrograman Python, selain list, tuple, dan dictionary. Berbeda dari tipe data koleksi lainnya, Set memiliki karakteristik utama yaitu hanya menyimpan elemen yang bersifat **unik** (tidak ada duplikasi) dan **tidak berurutan**.

Set sangat berguna dalam berbagai aplikasi pemrograman yang memerlukan penyaringan data duplikat atau operasi himpunan, seperti irisan (*intersection*), gabungan (*union*), selisih (*difference*), dan lain-lain.

Karakteristik Set:

1. Unik: Set tidak menyimpan elemen yang sama lebih dari satu kali. Jika kita mencoba memasukkan elemen yang sudah ada, Python akan mengabaikannya.
2. Tidak Berurutan: Elemen dalam Set tidak disimpan dalam urutan tertentu. Oleh karena itu, Set tidak dapat diakses melalui indeks seperti List atau Tuple.
3. Bersifat Mutable: Isi Set bisa diubah, yaitu elemen dapat ditambahkan atau dihapus.
4. Anggota Set harus Immutable: Elemen yang disimpan dalam Set harus bertipe data yang tidak dapat diubah seperti int, float, string, dan tuple. Tipe data yang mutable seperti list atau dictionary tidak dapat menjadi anggota Set.

Mendefinisikan Set:

Set dapat didefinisikan dengan dua cara:

1. Menggunakan tanda kurung kurawal {}
2. Menggunakan fungsi set()

Contoh:

```
# Menggunakan tanda kurung kurawal
bilangan_genap = {2, 4, 6, 8}

# Menggunakan fungsi set()
pionir_luar_angkasa = set(['Neil Armstrong', 'Buzz Aldrin'])
```

MATERI 12.3.2

Set tidak memiliki indeks, sehingga tidak dapat diakses seperti List atau Tuple. Untuk menampilkan semua isi dari Set, kita dapat menggunakan perulangan for.

```
nim = {'71200120', '71200195', '71200214'}

# Menampilkan jumlah elemen
print(len(nim))    # Output: 3

# Menampilkan elemen satu per satu
for n in nim:
    print(n)
```

Untuk menambahkan elemen ke dalam Set, gunakan fungsi .add().

Contoh:

```
plat_nomor = set()
plat_nomor.add('AB 1234 CD')
plat_nomor.add('AD 5678 EF')
plat_nomor.add('AB 1234 CD') # Tidak ditambahkan lagi karena duplikat
```

Python menyediakan beberapa metode untuk menghapus elemen dari Set, yaitu:

Fungsi	Deskripsi
<code>remove(x)</code>	Menghapus elemen x, akan error jika x tidak ada.
<code>discard(x)</code>	Menghapus elemen x, tidak error jika x tidak ditemukan.
<code>pop()</code>	Menghapus dan mengembalikan elemen secara acak dari Set.
<code>clear()</code>	Menghapus semua elemen dalam Set.

```
bilangan = {2, 3, 5, 7}
```

```
bilangan.remove(5)      # Menghapus 5
bilangan.discard(10)    # Tidak error walau 10 tidak ada
bilangan.pop()          # Menghapus elemen acak
bilangan.clear()        # Set kosong
```

MATERI 12.3.3

Python menyediakan operator-operator himpunan:

Operasi	Simbol / Fungsi
Gabungan (Union)	`
Irisan (Intersection)	& atau .intersection()
Selisih (Difference)	- atau .difference()
Selisih Simetris	^ atau .symmetric_difference()

Contoh Union:

```
a = {'Samsung', 'Sony'}  
b = {'Sony', 'Panasonic'}  
gabungan = a | b
```

Contoh Intersection:

```
renang = {'mail', 'ipin'}  
tenis = {'mail', 'ipin'}  
gabungan = renang & tenis
```

Contoh Difference:

```
english = {'desi', 'miko'}  
korean = {'miko', 'erika'}  
only_english = english - korean
```

Contoh Symetric Difference:

```
one_language = english ^ korean
```

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

Dari contoh kasus kategori kasus di Play Store, tambahkan kemampuan-kemampuan berikut ini:

- Tampilkan nama-nama aplikasi yang hanya muncul di satu kategori saja.
- Untuk input $n > 2$, tampilkan nama-nama aplikasi yang muncul tepat di dua kategori sekaligus

Jawab:

```
1  n = int(input('Masukkan jumlah kategori: '))
2  data_aplikasi = {}
3
4  for i in range(n):
5      nama_kategori = input('Masukkan nama kategori:')
6      print('Masukkan 5 nama aplikasi di kategori', nama_kategori)
7      aplikasi = []
8      for j in range(5):
9          nama_aplikasi = input('Nama aplikasi: ')
10         aplikasi.append(nama_aplikasi)
11         data_aplikasi[nama_kategori] = aplikasi
12
13     print('\nData Aplikasi per Kategori:')
14     print(data_aplikasi)
15
16     daftar_aplikasi_list = []
17     for aplikasi in data_aplikasi.values():
18         daftar_aplikasi_list.append(set(aplikasi))
19
20     print('\nSet Aplikasi per Kategori:')
21     print(daftar_aplikasi_list)
22
23     hasil = daftar_aplikasi_list[0]
24     for i in range(1, len(daftar_aplikasi_list)):
25         hasil = hasil.intersection(daftar_aplikasi_list[i])
26
27     print('\nAplikasi yang muncul di SEMUA kategori:')
28     print(hasil)
29
30     gabungan_semua = set()
31
32     from collections import Counter
33
34     semua_aplikasi_flat = []
35
36     for s in daftar_aplikasi_list:
37         gabungan_semua |= s
38         semua_aplikasi_flat += list(s)
39
40     frekuensi_aplikasi = Counter(semua_aplikasi_flat)
41
42     unik_satu_kategori = {app for app, freq in frekuensi_aplikasi.items() if freq == 1}
43     print('\nAplikasi yang hanya muncul di SATU kategori:')
44     print(unik_satu_kategori)
45
46     if n > 2:
47         tepat_dua_kategori = {app for app, freq in frekuensi_aplikasi.items() if freq == 2}
48         print('\nAplikasi yang muncul TEPAT di DUA kategori:')
49         print(tepat_dua_kategori)
50
```

Output:

```

Masukkan jumlah kategori: 3
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: Calculator
Masukkan nama kategori: Utilities
Masukkan 5 nama aplikasi di kategori Utilities
Nama aplikasi: Photos
Nama aplikasi: Weather
Nama aplikasi: Calculator
Nama aplikasi: Camera
Nama aplikasi: Notes
Masukkan nama kategori: Collages
Masukkan 5 nama aplikasi di kategori Collages
Nama aplikasi: Notes
Nama aplikasi: Calculator
Nama aplikasi: VSCode
Nama aplikasi: LMS
Nama aplikasi: EClass

Data Aplikasi per Kategori:
{'Finance': ['RTI Saham', 'Mirae', 'IPOT', 'Poems', 'Calculator'], 'Utilities': ['Photos', 'Weather', 'Calculator', 'Camera', 'Notes'], 'Collages': ['Notes', 'Calculator', 'VSCode', 'LMS', 'EClass']}

Set Aplikasi per Kategori:
[{'IPOT', 'Calculator', 'Mirae', 'RTI Saham', 'Poems'}, {'Photos', 'Camera', 'Calculator', 'Notes', 'Weather'}, {'LMS', 'Notes', 'Calculator', 'EClass', 'VSCode'}]

Aplikasi yang muncul di SEMUA kategori:
{'Calculator'}

Aplikasi yang hanya muncul di SATU kategori:
{'Photos', 'IPOT', 'Camera', 'LMS', 'Mirae', 'RTI Saham', 'Poems', 'EClass', 'VSCode'}

Aplikasi yang muncul TEPAT di DUA kategori:
{'Notes'}

```

Penjelasan:

Program dimulai dengan meminta input jumlah kategori. Nilai input ini akan menjadi batas perulangan. Perulangan menggunakan for dengan range dari jumlah kategori. Perulangan akan terus meminta inputan nama kategori dan nama-nama aplikasi sebanyak 5 kali untuk setiap kategori. Inputan nama aplikasi disimpan dalam list, lalu dimasukkan ke dalam dictionary dengan nama kategori sebagai kuncinya. Setelah semua kategori dan aplikasinya dimasukkan, program akan menampilkan dictionary yang berisi data aplikasi per kategori. Kemudian, program mengonversi setiap daftar aplikasi menjadi himpunan (set), dan semua himpunan ini dikumpulkan ke dalam list. Tujuannya agar operasi himpunan bisa dilakukan. Setelah itu, program mencari aplikasi yang muncul di semua kategori menggunakan operasi intersection. Hasilnya ditampilkan sebagai aplikasi yang muncul di semua kategori. Setelah bagian utama selesai, program dilanjutkan dengan tambahan kemampuan. Pertama, semua aplikasi yang sudah dimasukkan akan dihitung frekuensinya menggunakan Counter dari modul collections. Semua aplikasi dari setiap kategori digabung menjadi satu list besar, lalu dihitung berapa kali tiap aplikasi muncul. Jika aplikasi hanya muncul satu kali, maka itu berarti aplikasi tersebut hanya muncul di satu kategori. Hasilnya dimasukkan ke dalam himpunan dan ditampilkan sebagai aplikasi yang hanya muncul di satu kategori saja. Kemudian jika jumlah kategori lebih dari dua, maka program juga akan mencari aplikasi yang muncul tepat di dua kategori saja, yaitu aplikasi dengan frekuensi muncul dua kali. Hasilnya juga dimasukkan ke dalam himpunan dan ditampilkan sebagai output terakhir.

SOAL 2

Buatlah sebuah program yang mendemonstrasikan konversi dari:

- List menjadi Set
- Set menjadi List
- Tuple menjadi Set

- Set menjadi Tuple Tampilkan isi data sebelum dan sesudah konversi.

Jawab:

```

1  def list_ke_set(data_list):
2      print(set(data_list))
3
4  def set_ke_list(data_set):
5      print(list(data_set))
6
7  def tuple_ke_set(data_tuple):
8      print(set(data_tuple))
9
10 def set_ke_tuple(data_set):
11     print(tuple(data_set))
12
13 list_awal = [1, 2, 3, 4, 5, 3, 2]
14 set_awal = {10, 20, 30, 40, 50}
15 tuple_awal = (100, 200, 300, 200, 100)
16
17 set_dari_list = list_ke_set(list_awal)
18 list_dari_set = set_ke_list(set_awal)
19 set_dari_tuple = tuple_ke_set(tuple_awal)
20 tuple_dari_set = set_ke_tuple(set_awal)

```

Output:

```

{1, 2, 3, 4, 5}
[50, 20, 40, 10, 30]
{200, 100, 300}
(50, 20, 40, 10, 30)

```

Penjelasan:

Program membuat beberapa fungsi yang setiap fungsi bertugas mengubah tipe data tertentu. Fungsi `list_ke_set` menerima list dan mengembalikan set dari list tersebut, otomatis menghilangkan duplikat. Fungsi `set_ke_list` mengubah set jadi list agar bisa diakses berurutan. Fungsi `tuple_ke_set` mengubah tuple menjadi set untuk hilangkan duplikat. Fungsi `set_ke_tuple` mengubah set menjadi tuple yang isinya tidak bisa diubah. Setelah fungsi dibuat, program memanggil fungsi-fungsi tersebut dengan data contoh dan menampilkan hasil sebelum dan sesudah konversi.

SOAL 3

Buatlah sebuah program yang dapat membaca dua file teks dan menampilkan semua kata-kata yang muncul pada kedua file tersebut. Beberapa hal yang perlu anda perhatikan:

- Nama file adalah input user. Tampilkan pesan error jika file tidak ditemukan/tidak bisa dibaca.
- Semua kata dikonversi dulu menjadi lowercase.
- Sertakan contoh file teks yang anda pakai saat mengumpulkan laporan.

Jawab:

```
def baca_file(filename):
    try:
        with open(filename, 'r') as file:
            isi = file.read()
            kata_list = isi.lower().split()
            return set(kata_list)
    except FileNotFoundError:
        print(f"Error: File '{filename}' tidak ditemukan.")
        return None
    except IOError:
        print(f"Error: File '{filename}' tidak bisa dibaca.")
        return None

file1 = input("Masukkan nama file pertama: ")
file2 = input("Masukkan nama file kedua: ")

kata_file1 = baca_file(file1)
kata_file2 = baca_file(file2)

if kata_file1 is not None and kata_file2 is not None:
    kata_sama = kata_file1.intersection(kata_file2)
    if kata_sama:
        print("\nKata-kata yang muncul di kedua file:")
        for kata in sorted(kata_sama):
            print(kata)
    else:
        print("\nTidak ada kata yang sama di kedua file.")
```

Output:

```
Masukkan nama file pertama: p
Masukkan nama file kedua: p
Error: File 'p' tidak ditemukan.
Error: File 'p' tidak ditemukan.
PS D:\Kuliah\Semester_6\Alpro\Laporan_14
Masukkan nama file pertama: puisi1.txt
Masukkan nama file kedua: puisi2.txt

Kata-kata yang muncul di kedua file:
dan
jalan
manis
oleh
```

Penjelasan:

Program meminta nama file pertama dan kedua dari user. Setiap file akan dibaca oleh fungsi `baca_file` yang mencoba membuka file tersebut. Jika file tidak ditemukan, fungsi akan menampilkan pesan error dan mengembalikan `None`. Jika file berhasil dibaca, isi file akan diubah menjadi lowercase dan dipisah menjadi kata-kata unik menggunakan `set`. Setelah kedua file berhasil dibaca, program mencari kata yang muncul di kedua file dengan operasi `intersection` pada dua `set` kata tadi. Hasil kata yang sama ditampilkan secara terurut, dan jika tidak ada kata yang sama, program akan menampilkan pesan bahwa tidak ada kata yang sama.

Link GitHub:

https://github.com/TimotiusEkanaT/Alpro/tree/main/Laporan_14