

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
Minggu ke / Materi	12 / 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
2023

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

Pengenalan dan Mendefinisikan Set

Set adalah salah satu tipe data pada Python yang dapat digunakan untuk menyimpan sekumpulan data yang semuanya unik. Dikenal juga dengan istilah himpunan. Beberapa sifat *Set* pada Python sebagai berikut:

- Isi dari *Set* disebut sebagai anggota (*member*).
- Anggota dari *Set* harus bersifat *immutable*.
- *Set* sendiri bersifat *mutable*, artinya dapat menambah atau mengurangi isi dari sebuah *Set*. Karena itu *Set* tidak dapat dimasukkan ke dalam *Set*.

Beberapa tipe data *immutable* pada Python:

1. Integer,
2. Float,
3. String,
4. Tuple, dan lain-lain.

Dengan demikian *list* dan *dictionary* (*mutable*) tidak dapat dimasukkan ke dalam *Set*.

Perbedaan List, Tuple, Set, dan Dictionary.

- List dapat berubah dan dideklarasikan dengan tanda kurung siku [].
- Tuple tidak dapat diubah dan diapit dalam tanda kurung ().
- Set bisa berubah dan tidak memiliki elemen duplikat dalam tanda kurung kurawal {}.
- Dictionary bisa berubah dan kunci tidak mengizinkan duplikat. Dilampirkan oleh tanda kurung kurawal dalam bentuk pasangan *key-value*.

source: <https://revou.co/panduan-teknis/python-set>

Membuat Set

```
set_1 = {}  
set_2 = set()
```

Mengakses Set

```
kata = {"Alfa", "Omega", "Datang"}  
print(type(kata))  
print(kata)  
print(kata[1]) # error: tidak bisa diakses lewat index
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
<class 'set'>
{'Alfa', 'Omega', 'Datang'}
Traceback (most recent call last):
  File "d:\71200645_PrakAlpro_12\Modul.py", line 74, in <module>
    print(kata[1]) # tidak bisa diakses lewat index
          ~~~~~^^
TypeError: 'set' object is not subscriptable
PS D:\71200645_PrakAlpro_12>
```

Menggunakan for loop

```
kata = {"Alfa", "Omega", "Datang"}
for i in kata:
    print(i)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
Alfa
Datang
Omega
PS D:\71200645_PrakAlpro_12>
```

Mengubah tipe data lain menjadi Set.

List ke Set

```
kata = set(["Kasih", "Yesus", "Selamanya"])
print(type(kata))
```

Tuple ke Set

```
kata2 = set(("Hidup", "Kekal", "Surga"))
print(type(kata2))
```

Pengaksesan Set

Set tidak memiliki indeks, karena itu tidak dapat mengakses anggota-anggota dari sebuah *Set* secara langsung.

- *discard()*: Menghapus satu elemen yang disebutkan, tidak menghasilkan error jika elemen yang ingin dihapus tidak ditemukan.
- *remove()*: Menghapus satu elemen yang disebutkan, menghasilkan error jika elemen yang ingin dihapus tidak ditemukan.
- *pop()*: Mengambil salah satu dan menghapusnya dari *Set* (tidak tentu).
- *clear()*: Menghapus seluruh elemen di dalam *Set*.

Menambah elemen dalam Set

```
simpan = {"a", "b", 1, 5, "c"}  
simpan.add("d") # hanya bisa menambah satu data
```

Menghapus elemen dalam Set

Dengan `discard()`

```
simpan = {"a", "b", 1, 5, "c"}  
simpan.discard(2) # tidak ada error  
simpan.discard(5)
```

Dengan `remove()`

```
simpan = {"a", "b", 1, 5, "c"}  
# simpan.remove(2) # error  
simpan.remove("b")  
print(simpan)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs  
/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py  
{1, 'd', 'c', 'a'}  
PS D:\71200645_PrakAlpro_12>
```

Menghitung jumlah daerah.

Menghapus duplikasi: mengubah list menjadi set

```
def hitung_daerah(daftar):  
    daerah_unik = set(daftar)  
    jumlah_daerah = len(daerah_unik)  
    return daerah_unik, jumlah_daerah  
  
daftar = ["Jakarta", "Bandung", "Surabaya", "Jakarta", "Medan", "Bandung",  
"Bali", "Surabaya", "Bali", "Makassar"]  
daerah_unik, jumlah_daerah = hitung_daerah(daftar)  
print("Daerah yang diwakili:", daerah_unik)  
print("Jumlah daerah:", jumlah_daerah)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs  
/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py  
Daerah yang diwakili: {'Bandung', 'Surabaya', 'Bali', 'Jakarta',  
'Makassar', 'Medan'}  
Jumlah daerah: 6  
PS D:\71200645_PrakAlpro_12> □
```

MATERI 2

OPERASI-OPERASI PADA SET

Menggabungkan Set

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

1. Operator *Union*: Menggabungkan dua *Set* menjadi satu. Dapat menggunakan operator `|` maupun fungsi `union()`.

Menggunakan fungsi `union()`.

```
set_1 = {"Hello", "Jesus", 3}
set_2 = {"Bapa", "Putra", "Roh Kudus"}
set_gabung = set_1.union(set_2)
print(set_gabung)
```

Menggunakan operator `|`

```
set_gabung_2 = set_1 | set_2
print(set_gabung_2)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
{3, 'Putra', 'Bapa', 'Jesus', 'Roh Kudus', 'Hello'}
{3, 'Putra', 'Bapa', 'Jesus', 'Roh Kudus', 'Hello'}
PS D:\71200645_PrakAlpro_12>
```

2. Operator *Intersection*: Menghasilkan irisan dari dua *Set*. Dapat menggunakan operator `&` maupun fungsi `intersection()`.

Menggunakan fungsi `intersection()`.

```
set_1 = {"Hello", "Jesus", 3}
set_4 = {"Jesus", "Manusia", "Bapa"}
set_iris = set_1.intersection(set_4)
print(set_iris)
```

Menggunakan operator `&`

```
set_iris = set_1 & set_4
print(set_iris)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
{'Jesus'}
PS D:\71200645_PrakAlpro_12>
```

3. Operator *Difference*: Menghasilkan *Set* baru yang merupakan selisih dari dua *Set* yang dibandingkan. Dapat menggunakan operator `-` maupun fungsi `difference()`.

Menggunakan fungsi `difference()`.

```
set_1 = {"Hello", "Jesus", 3}
set_4 = {"Jesus", "Manusia", "Bapa"}
set_selisih = set_1.difference(set_4)
print(set_selisih)
```

Menggunakan operator `-`.

```
set_selisih = set_1 - set_4
print(set_selisih)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
{3, 'Hello'}
PS D:\71200645_PrakAlpro_12>
```

4. Operator *Symmetric Difference*: Menghasilkan *Set* baru yang merupakan jumlah dari dua *Set* kecuali irisannya. Dapat menggunakan operator `^` maupun fungsi `symmetric_difference()`.

Menggunakan fungsi `symmetric_difference()`.

```
set_1 = {"Hello", "Jesus", 3}
set_4 = {"Jesus", "Manusia", "Bapa"}
set_baru = set_1.symmetric_difference(set_4)
print(set_baru)
```

Menggunakan operator `^`.

```
set_baru = set_1 ^ set_4
print(set_baru)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
{3, 'Bapa', 'Hello', 'Manusia'}
PS D:\71200645_PrakAlpro_12>
```

Metode `update()`

```
set_3 = {"Bumi", "Surga", "Neraka"}
set_1.update(set_3)
print(set_1)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_12/Modul.py
{3, 'Hello', 'Bumi', 'Surga', 'Neraka', 'Jesus'}
PS D:\71200645_PrakAlpro_12>
```

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

Latihan 12.1 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

Pengerjaan soal dilakukan dengan meminta *input* dari *user* berapa banyak kategori yang diinginkan dalam variabel n . *Dictionary* `data_aplikasi` menyimpan `nama_kategori` sebagai *key* dan aplikasi yang menyimpan `nama_aplikasi` sebagai *value*. Dengan *dictionary* `frekuensi_aplikasi` digunakan untuk menghitung berapa kali setiap aplikasi muncul di semua kategori, jika *count* aplikasi hanya muncul satu kali dianggap hanya muncul dalam satu kategori, sehingga menampilkan nama aplikasi tersebut. Untuk input kategori lebih dari dua, penghitungan sama dengan `frekuensi_aplikasi`, hanya saja `count == 2`.

SOURCE CODE:

```
n = int(input("Masukkan jumlah kategori: "))

data_aplikasi = {}
for i in range(n):
    nama_kategori = input("Masukkan nama kategori: ")
    print("Masukkan 5 nama aplikasi di kategori", nama_kategori)

    aplikasi = []
    for j in range(5):
        nama_aplikasi = input("Nama aplikasi: ")
        aplikasi.append(nama_aplikasi)

    data_aplikasi[nama_kategori] = aplikasi

print(data_aplikasi)

daftar_aplikasi_list = []
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("Aplikasi yang muncul di semua kategori:", hasil)

# tambahan: aplikasi yang hanya muncul di satu kategori
frekuensi_aplikasi = {}
for kategori in data_aplikasi.values():
    for aplikasi in kategori:
        if aplikasi in frekuensi_aplikasi:
            frekuensi_aplikasi[aplikasi] += 1
        else:
            frekuensi_aplikasi[aplikasi] = 1

unik_satu_kategori = {aplikasi for aplikasi, count in
frekuensi_aplikasi.items() if count == 1}
print("Aplikasi yang hanya muncul di satu kategori:", unik_satu_kategori)

if n > 2:
    dua_kategori = {aplikasi for aplikasi, count in
frekuensi_aplikasi.items() if count == 2}
    print("Aplikasi yang muncul tepat di dua kategori:", dua_kategori)

```

OUTPUT:

```

{'Tuhan': ['Roh', 'Yesus', 'Firman', 'Alfa', 'Omega'], 'Iblis': ['Bumi', 'Roh', 'Tirus', 'Dosa', 'Binasa'],
'Manusia': ['Tubuh', 'Jiwa', 'Roh', 'Binasa', 'Bumi']}
[{'Omega', 'Roh', 'Alfa', 'Firman', 'Yesus'}, {'Binasa', 'Bumi', 'Tirus', 'Roh', 'Dosa'}, {'Binasa', 'Bumi',
'Roh', 'Tubuh', 'Jiwa'}]
Aplikasi yang muncul di semua kategori: {'Roh'}
Aplikasi yang hanya muncul di satu kategori: {'Jiwa', 'Tirus', 'Omega', 'Dosa', 'Alfa', 'Firman', 'Tubuh',
'Yesus'}
Aplikasi yang muncul tepat di dua kategori: {'Bumi', 'Binasa'}
PS D:\71200645_PrakAlpro_12> 

```

SOAL 2

Latihan 12.2 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. ■

Pengerjaan dilakukan dengan daftar data dari *list* dan *tuple*. Menggunakan fungsi `print()` untuk menampilkan data awal. Perubahan data ke *list*, ke *tuple*, dan ke *set*, berdasarkan parameter kedua “target” dari fungsi konversi dan menggunakan percabangan *if-else*. Jika berupa “*set*”, maka mengubah data menjadi *set* dengan fungsi `set()`, apabila “*list*”, jika data dari bentuk *set*, maka diubah menjadi *list*. Jika “*tuple*”, maka data dari bentuk *set* diubah menjadi *tuple*.

SOURCE CODE:

```
def konversi(data, target):
    print("Data:", data)
    hasil = None

    if target == "set":
        hasil = set(data)
        print("Konversi ke Set:", hasil)
    elif target == "list":
        if isinstance(data, set):
            hasil = list(data)
        else:
            hasil = list(data)
        print("Konversi ke List:", hasil)
    elif target == "tuple":
        if isinstance(data, set):
            hasil = tuple(data)
        else:
            hasil = tuple(data)
        print("Konversi ke Tuple:", hasil)

    return hasil

list_data = ["Hello", 2, 23.4, 2, 1]
tuple_data = (4, 5, "Langkah", 5, 4)

list_to_set = konversi(list_data, "set")
set_to_list = konversi(list_to_set, "list")
tuple_to_set = konversi(tuple_data, "set")
set_to_tuple = konversi(tuple_to_set, "tuple")
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_12/Latihan 12.2.py"
Data: ['Hello', 2, 23.4, 2, 1]
Konversi ke Set: {1, 2, 'Hello', 23.4}
Data: {1, 2, 'Hello', 23.4}
Konversi ke List: [1, 2, 'Hello', 23.4]
Data: (4, 5, 'Langkah', 5, 4)
Konversi ke Set: {'Langkah', 4, 5}
Data: {'Langkah', 4, 5}
Konversi ke Tuple: ('Langkah', 4, 5)
PS D:\71200645_PrakAlpro_12>
```

SOAL 3

Latihan 12.3 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.

Pengerjaan dilakukan dengan menggunakan fungsi *try-except*, di bagian *except* untuk menampilkan pesan *error* apabila *file* tidak ditemukan (*FileNotFoundError*) atau tidak bisa dibaca. Pada bagian *try* dimulai dengan meminta *input* dua buah *file* dari pengguna kemudian membuka *file* dengan fungsi *with open()* dalam mode *read*. Pada saat membuka dan membaca *file*, baris-baris kata dari *file* diubah menjadi *lower()* dan *split()* untuk mendapatkan setiap kata yang dipisahkan dengan *whitespace*. Untuk mencari kata yang sama yang muncul di kedua file digunakan fungsi *intersection()* untuk *kata_set1* dan *kata_set2*. Lalu selama *for loop*, dilakukan pula *sorted()* untuk merapikan urutan kata yang muncul.

SOURCE CODE:

```
def kata():
    try:
        nama_file1 = input("Masukkan nama file pertama: ")
        nama_file2 = input("Masukkan nama file kedua: ")

        with open(nama_file1, 'r') as file1:
            kata_set1 = set(file1.read().lower().split())

        with open(nama_file2, 'r') as file2:
            kata_set2 = set(file2.read().lower().split())

        # Mencari kata yang muncul di kedua file
        kata_sama = kata_set1.intersection(kata_set2)
        print("Kata-kata yang muncul di kedua file: ")
        for i in sorted(kata_sama):
            print(i)

    except FileNotFoundError as e:
        print(f"Error: file tidak ditemukan - {e}")
    except Exception as e:
        print(f"Error: terjadi kesalahan - {e}")

kata()
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_12> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_12/Latihan 12 .3.py"
Masukkan nama file pertama: file1.txt
Masukkan nama file kedua: file2.txt
Kata-kata yang muncul di kedua file:
dia
sampai
yang
yesus
PS D:\71200645_PrakAlpro_12> █
```

Link Github:

https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_12