



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230970
Nama Lengkap	Gregorius Daniel Jodan Perminas
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 KULIAHINI.**

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.

### Pengertian Tipe Data Set dan Mendefinisikan Set

Set merupakan sebuah tipe data Python yang berisi sekumpulan data dan dipakai untuk mengolah himpunan (set). Beberapa ketentuan tipe data set:

- Isi dari set disebut sebagai anggota (member)
- Nilai anggota yang disimpan harus bersifat unik (tidak duplikat).
- Set bersifat unordered (tidak memiliki index) alias tidak berurut
- Anggota dari set harus bersifat immutable. Contohnya integer, float, string, tuple dan lain-lain
- Set bersifat mutable artinya isi dari set dapat ditambah atau dikurangi. Karena sifat ini maka set tidak bisa dimasukkan ke dalam set

Beberapa cara yang dapat digunakan untuk mendefinisikan set:

#### PROGRAM 1

```
# Menggunakan Kurung Kurawal
huruf_vokal = {"a", "i", "u", "e", "o"}
huruf_konsonan = {"b", "c", "d", "f", "g"}
print(huruf_vokal)
print(huruf_konsonan)
```

#### OUTPUT 1

```
['a', 'e', 'o', 'i', 'u']
['b', 'g', 'd', 'f', 'c']
```

#### PROGRAM 2

```
# Menggunakan Fungsi Set (Dimasukkan Dalam List Terlebih Dahulu)
astronot = set(["Neil Armstrong", "Buzz Aldrin"])
print(astronot)
```

#### OUTPUT 2

```
{'Neil Armstrong', 'Buzz Aldrin'}
```

### PROGRAM 3

```
# Mendefinisikan Set Kosong
astronotFunction = set() # Cara Yang Benar
astronotKurawal = {} # Cara Yang Salah
print("Tipe data astronotFunction:", type(astronotFunction))
print("Tipe data astronotKurawal:", type(astronotKurawal))
```

### OUTPUT 3

```
Tipe data astronotFunction: <class 'set'>
Tipe data astronotKurawal: <class 'dict'>
```

## Pengaksesan Set

Sifat tipe data set adalah *unordered* maka set tidak memiliki indeks. Berikut salah satu cara untuk mengakses isi set:

### PROGRAM 1

```
nim = {'71200120', '71200195', '71200214'}
for n in nim:
    print(n)
```

### OUTPUT 1

```
71200214
71200195
71200120
```

Dapat dilihat pada output program di atas urutan nim berbeda dengan deklarasi set sebelumnya. Hal ini membuktikan bahwa set tidak memiliki indeks. Pada tipe data ini posisi anggota menjadi tidak penting. Sama halnya dengan dictionaries, set adalah tipe data mutable dan isinya bisa ditambah atau dikurangi. Berikut beberapa caranya dalam Python:

### PROGRAM 2

```
plat_nomor = set()

plat_nomor.add('AB 1890 XA') # Menambah Plat Nomor 'AB 1890 XA'
plat_nomor.add('AD 6810 MT') # Menambah Plat Nomor 'AD 6810 MT'
plat_nomor.add('AB 1890 XA') # Menambah Plat Nomor Yang Sama Dengan Sebelumnya

print(len(plat_nomor))
for plat in plat_nomor:
    print(plat)
```

## OUTPUT 2

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

Karena set tidak bisa menerima nilai duplikat maka tipe data ini akan selalu mengecek apakah anggota baru yang akan dimasukkan sudah ada di dalam set atau belum. Jika kita memasukkan nilai yang sudah ada pada suatu set, maka nilai tersebut hanya akan muncul atau dimasukkan 1 kali saja.

Untuk menghapus anggota dari sebuah set ada beberapa cara yang dapat digunakan yaitu;

- `remove(value)` = Untuk menghapus elemen yang dicari. Jika elemen yang dicari tidak ada, maka akan error
- `discard(value)` = Untuk menghapus elemen yang dicari. Tidak ada error
- `pop()` = Mengambil salah satu elemen dan menghapusnya dari set. Error jika set kosong
- `clear()` = Menghapus seluruh elemen dalam set. Tidak ada error

## PROGRAM 3

```
bilangan_prima = {13, 23, 7, 29, 11, 5}
print("setBilanganPrima:", bilangan_prima)

# Menghapus angka 5 dari set
bilangan_prima.remove(5)
print("remove(5):", bilangan_prima)

# Menghapus angka 97 (tidak ada)
bilangan_prima.discard(97)
print("discard(97):", bilangan_prima)

# Mengambil dan menghapus salah satu elemen
bilangan = bilangan_prima.pop()
print("pop():", bilangan)
print("setAfterPop():", bilangan_prima)

# Mengosongkan set
bilangan_prima.clear()
print("clear():", bilangan_prima)
```

## OUTPUT 3

```
setBilanganPrima: {29, 23, 5, 7, 11, 13}
remove(5): {29, 23, 7, 11, 13}
discard(97): {29, 23, 7, 11, 13}
pop(): 29
setAfterPop(): {23, 7, 11, 13}
clear(): set()
```

Dapat dilihat pada output program bahwa fungsi discard() memang tidak akan mengembalikan error meskipun anggota yang ingin dihapus tidak terdapat dalam set. Sedangkan fungsi pop() mengambil salah satu anggota set secara acak sekaligus mengembalikan nilai yang diambil.

Jika ingin mengganti nilai dari salah satu anggota set, maka metode yang paling mungkin adalah dengan melakukan operasi replace dengan cara menghapus anggota yang mau diubah dan menambahkan anggota baru sesuai dengan yang diinginkan. Berikut contohnya:

#### PROGRAM 4

```
# Buat Set dari List
ikan = set(['lele', 'nila', 'kembung', 'salmon'])
print(ikan)
# ganti Lele menjadi bandeng
ikan.remove('lele')
ikan.add('bandeng')
print(ikan)
```

#### OUTPUT 4

```
{'lele', 'salmon', 'nila', 'kembung'}
{'salmon', 'bandeng', 'nila', 'kembung'}
```

### Operasi Pada Set

Keunggulan dari tipe data set adalah keunikan anggotanya. Karena keunikan tersebut, python menyediakan berbagai fungsi keanggotaan yang berguna untuk pengolahan data yaitu:

- fungsi union “|” (gabungan) = Menggabungkan dua set menjadi satu
- fungsi intersection “&” (irisan) = Menghasilkan irisan dari dua set
- fungsi difference “-“ (selisih) = Menghasilkan set baru yang merupakan selisih dari dua Set yang dibandingkan
- fungsi symmetric difference “^” (komplement) = Menghasilkan set baru yang merupakan jumlah dari dua set kecuali irisananya

Berikut ini adalah demonstrasinya dengan Python:

#### PROGRAM 1

```
# FUNGSI UNION

merek_hp = {'Samsung', 'Apple', 'Xiaomi', 'Sony'}
merek_ac = {'LG', 'Samsung', 'Panasonic', 'Daikin', 'Sony'}
gabungan = merek_hp.union(merek_ac) # merek_hp | merek_ac
print(gabungan)
```

### OUTPUT 1

```
{'Sony', 'Xiaomi', 'Panasonic', 'Apple', 'Samsung', 'LG', 'Daikin'}
```

### PROGRAM 2

```
# FUNGSI INTERSECTION

renang = {'siti', 'mail', 'ikhsan', 'upin', 'ipin'}
tenis = {'joko', 'mail', 'ipin', 'upin', 'tejo'}
# Menyukai renang dan tenis
renang_tenis = renang.intersection(tenis) # renang & tenis
print(renang_tenis)
```

### OUTPUT 2

```
{'ipin', 'mail', 'upin'}
```

### PROGRAM 3

```
# FUNGSI DIFFERENCE

# bisa berbahasa inggris
english = {'desi', 'tono', 'evan', 'miko', 'takashi', 'ron'}
# bisa berbahasa jepang
japanese = {'ron', 'yuki', 'erika', 'miko'}
# yang hanya bisa berbahasa jepang
only_japanese = japanese.difference(english) # japanese- english
print(only_japanese)
# yang hanya bisa berbahasa inggris
only_english = english.difference(japanese) # english - japanese
print(only_english)
```

### OUTPUT 3

```
{'yuki', 'erika'}
{'desi', 'evan', 'takashi', 'tono'}
```

## PROGRAM 4

```
# FUNGSI SYMMETRIC DIFFERENCE

# bisa berbahasa inggris
english = {'desi', 'tono', 'evan', 'miko', 'takashi', 'ron'}
# bisa berbahasa jepang
japanese = {'ron', 'yuki', 'erika', 'miko'}
# yang hanya bisa satu bahasa
one_languange = english.symmetric_difference(japanese) # english ^ japanese
print(one_languange)
```

## OUTPUT 4

```
{'desi', 'yuki', 'evan', 'tono', 'erika', 'takashi'}
```

## REFERENSI:

[Python Dasar: Tipe Data Set](#) | Jago Ngoding

[Tutorial Belajar Python Part 14: Tipe Data Set dalam Bahasa Python](#) | DuniaIlkom

## 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

#### PROGRAM LATIHAN MANDIRI 12.1

```
# Aplikasi Play Store
def aplikasiPlayStore(n):
    # Membuat dictionary berisi Kategori dan Aplikasi
    dictKategori = dict()
    for i in range(n):
        kategori = str(input(f'Masukkan kategori ke {i+1}: '))
        lstAplikasi = []
        for j in range(5):
            aplikasi = str(input(f'Masukkan aplikasi ke {j+1} dari kategori {kategori}: '))
            lstAplikasi.append(aplikasi)
        else:
            dictKategori[kategori] = lstAplikasi
            print()
    print(dictKategori)
    print()
    # Membuat List Untuk Menyimpan Aplikasi Dari Dictionary
    listAplikasi = []
    for k in dictKategori.keys():
        listAplikasi.append(dictKategori[k])
    # Aplikasi Yang Terdapat Pada Semua Kategori
    hasil = set(listAplikasi[0])
    for i in range(1, len(listAplikasi)):
        hasil = hasil.intersection(set(listAplikasi[i]))
    if len(hasil) > 0:
        print("Aplikasi Yang Terdapat Pada Semua Kategori:", hasil)
    else:
        print('Tidak Ada Aplikasi Yang Muncul Pada Semua Kategori!')
    # Aplikasi Yang Muncul Hanya Pada Satu Kategori (fitur tambahan 1)
    hasil1 = set(listAplikasi[0])
    for i in range(1, len(listAplikasi)):
        hasil1 = hasil1.symmetric_difference((set(listAplikasi[i])))
    if len(hasil1) > 0:
        print("Aplikasi Yang Muncul Hanya Pada Satu Kategori:", hasil1)
    else:
        print('Semua Aplikasi Muncul Pada Semua Kategori!')
    # Aplikasi Yang Muncul Tepat 2 Kategori (fitur tambahan 2)
    if n > 2:
        dictAplikasi = {}
        for lst in listAplikasi:
            for aplikasi in lst:
                if aplikasi in dictAplikasi:
                    dictAplikasi[aplikasi] += 1
                else:
                    dictAplikasi[aplikasi] = 1
        hasil2 = set([k for k, v in dictAplikasi.items() if v == 2])
        if hasil2:
            print('Aplikasi Yang Muncul Tepat 2 Kategori:', hasil2)
        else:
            print('Tidak ada aplikasi yang muncul tepat 2 kategori!')

n = int(input('Masukkan berapa kategori: '))
aplikasiPlayStore(n)
```

## OUTPUT PROGRAM

```
{'Tools': ['Kamera', 'Recorder', 'Galeri', 'SMS', 'Kompas'], 'Games': ['PUBG', 'CSGO', 'Chess', 'Arcade', 'hess', 'Arcade', 'SMS'], 'Other': ['SMS', 'Arcade', 'Tetris', 'Google', 'Youtube']}
```

```
Applikasi Yang Terdapat Pada Semua Kategori: {'SMS'}
```

```
Applikasi Yang Muncul Hanya Pada Satu Kategori: {'Kompas', 'Recorder', 'Galeri', 'Kamera', 'PUBG', 'Google', 'Youtube', 'SMS', 'Chess', 'Tetris', 'CSGO'}
```

```
Applikasi Yang Muncul Tepat 2 Kategori: {'Arcade'}
```

Program ini merupakan program yang digunakan untuk mengelola aplikasi berdasarkan kategorinya, kemudian menemukan aplikasi yang muncul pada semua kategori, hanya pada satu kategori, atau tepat pada dua kategori

- Fungsi aplikasiPlayStore(n): Fungsi ini menerima satu parameter yaitu n yang merupakan jumlah kategori aplikasi.
- Membuat dictionary berisi Kategori dan Aplikasi: Program ini membuat sebuah dictionary kosong, dictKategori. Untuk setiap kategori (sebanyak n kali), program meminta pengguna untuk memasukkan nama kategori dan lima aplikasi yang termasuk dalam kategori tersebut. Semua informasi ini disimpan dalam dictKategori.
- Membuat List Untuk Menyimpan Aplikasi Dari Dictionary: Program ini membuat sebuah list kosong dengan nama listAplikasi, kemudian menambahkan semua aplikasi dari setiap kategori ke dalam list ini.
- Applikasi Yang Terdapat Pada Semua Kategori: Program ini mencari aplikasi yang terdapat pada semua kategori dengan menggunakan metode intersection() dari tipe data set. Jika ada, program akan mencetak nama aplikasi tersebut. Jika tidak, program akan mencetak pesan bahwa tidak ada aplikasi yang muncul pada semua kategori.
- Applikasi Yang Muncul Hanya Pada Satu Kategori: Merupakan fitur tambahan yang pertama, yaitu program ini mencari aplikasi yang muncul hanya pada satu kategori dengan menggunakan metode symmetric\_difference() dari tipe data set. Jika ada, program akan mencetak nama aplikasi tersebut. Jika tidak, program akan mencetak pesan bahwa semua aplikasi muncul pada semua kategori.
- Applikasi Yang Muncul Tepat 2 Kategori: Sebagai fitur tambahan kedua, jika jumlah kategori lebih dari dua, program akan mencari aplikasi yang muncul tepat pada dua kategori. Program ini melakukan ini dengan membuat sebuah dictionary baru, dictAplikasi, di mana kunci adalah nama aplikasi dan nilai adalah jumlah kategori di mana aplikasi tersebut muncul. Jika ada aplikasi yang muncul tepat pada dua kategori, program akan mencetak nama aplikasi tersebut. Jika tidak, program akan mencetak pesan bahwa tidak ada aplikasi yang muncul tepat pada dua kategori.
- Input dan pemanggilan fungsi: Program ini meminta pengguna untuk memasukkan jumlah kategori dan kemudian memanggil fungsi aplikasiPlayStore(n) dengan jumlah kategori sebagai argumen.

## SOAL 2

### PROGRAM LATIHAN MANDIRI 12.2

```
# Program yang mendemonstrasikan konversi list-set-tuple
def konversiListSet(inputList):
    print("PROGRAM KONVERSI LIST KE SET DAN SET KE LIST")
    listSet = set(inputList)
    print("List:", inputList, "ke Set:", listSet)
    print("Tipe Data Sebelum:", type(inputList), "Tipe Data Sesudah:", type(listSet))
    listAwal = list(listSet)
    print("Set:", listSet, "ke List:", listAwal)
    print("Tipe Data Sebelum:", type(listSet), "Tipe Data Sesudah:", type(listAwal))
    print()

def konversiTupleSet(inputTuple):
    print("PROGRAM KONVERSI TUPLE KE SET DAN SET KE TUPLE")
    tupleSet = set(inputTuple)
    print("Tuple:", inputTuple, "ke Set:", tupleSet)
    print("Tipe Data Sebelum:", type(inputTuple), "Tipe Data Sesudah:", type(tupleSet))
    tupleAwal = tuple(tupleSet)
    print("Set:", tupleSet, "ke Tuple:", tupleAwal)
    print("Tipe Data Sebelum:", type(tupleSet), "Tipe Data Sesudah:", type(tupleAwal))
    print()

data1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
konversiListSet(data1)
data2 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
konversiTupleSet(data2)
```

### OUTPUT PROGRAM

```
PROGRAM KONVERSI LIST KE SET DAN SET KE LIST
List: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] ke Set: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Tipe Data Sebelum: <class 'list'> Tipe Data Sesudah: <class 'set'>
Set: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} ke List: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Tipe Data Sebelum: <class 'set'> Tipe Data Sesudah: <class 'list'>

PROGRAM KONVERSI TUPLE KE SET DAN SET KE TUPLE
Tuple: (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) ke Set: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Tipe Data Sebelum: <class 'tuple'> Tipe Data Sesudah: <class 'set'>
Set: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} ke Tuple: (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
Tipe Data Sebelum: <class 'set'> Tipe Data Sesudah: <class 'tuple'>
```

Program ini mendemonstrasikan konversi dari list, set, dan tuple

- Fungsi konversiListSet(inputList): Fungsi ini menerima sebuah list sebagai input dan mengubahnya menjadi set menggunakan fungsi set(). Setelah itu, fungsi ini mencetak list awal, set hasil konversi, dan tipe data sebelum dan sesudah konversi. Kemudian, fungsi ini mengubah set kembali menjadi list menggunakan fungsi list() dan mencetak set awal, list hasil konversi, dan tipe data sebelum dan sesudah konversi.
- Fungsi konversiTupleSet(inputTuple): Fungsi ini menerima sebuah tuple sebagai input dan mengubahnya menjadi set menggunakan fungsi set(). Setelah itu, fungsi ini mencetak tuple awal,

set hasil konversi, dan tipe data sebelum dan sesudah konversi. Kemudian, fungsi ini mengubah set kembali menjadi tuple menggunakan fungsi tuple() dan mencetak set awal, tuple hasil konversi, dan tipe data sebelum dan sesudah konversi.

- Pemanggilan fungsi: Di dalam program ini telah didefinisikan sebuah list dan tuple yang berisi angka dari 1 sampai 10. Kemudian, program ini memanggil fungsi konversiListSet(data1) dan konversiTupleSet(data2) dengan list dan tuple tersebut sebagai argumen.

### SOAL 3

#### PROGRAM LATIHAN MANDIRI 12.3

```
# Program yang Menampilkan Kata yang Muncul Dalam Kedua File Text
def munculKata(file1, file2):
    with open(file1, 'r') as f1, open(file2, 'r') as f2:
        linesFile1 = f1.readlines()
        linesFile2 = f2.readlines()
    # Memproses setiap baris dan memisahkan kata-kata
    wordsFile1 = []
    for line in linesFile1:
        for word in line.split():
            if word.isalpha():
                wordsFile1.append(word.lower())
    wordsFile2 = []
    for line in linesFile2:
        for word in line.split():
            if word.isalpha():
                wordsFile2.append(word.lower())
    # Memasukkan Kata Dalam Set
    tempatKata = set(wordsFile1).intersection(set(wordsFile2))
    print(tempatKata)

munculKata('romeo.txt', 'romeo-full.txt')
```

#### OUTPUT PROGRAM

```
{'envious', 'light', 'it', 'window', 'the', 'juliet', 'and', 'through', 'moon', 'kill', 'with', 'what',
'pale', 'but', 'is', 'who', 'sick', 'fair', 'already', 'yonder'}
```

Program ini akan membaca dua file teks dan menampilkan semua kata-kata yang muncul pada kedua file tersebut.

- Fungsi munculKata(file1, file2): Fungsi ini menerima dua parameter, file1 dan file2, yang merupakan nama dari dua file teks.
- Membaca file: Program ini membuka dan membaca kedua file teks menggunakan fungsi open() dan readlines(). Hasilnya adalah list yang berisi setiap baris dalam file teks.
- Memproses setiap baris dan memisahkan kata-kata: Program ini kemudian akan memproses setiap baris dalam list dan memisahkannya menjadi kata per kata menggunakan fungsi split(). Jika kata tersebut hanya berisi karakter alfabet (diperiksa menggunakan fungsi isalpha()), kata

tersebut ditambahkan ke list kata-kata untuk file tersebut. Semua kata diubah menjadi huruf kecil menggunakan fungsi lower() untuk memastikan bahwa perbandingannya tidak case-sensitive.

- Memasukkan Kata Dalam Set: Program ini mengubah list kata-kata untuk setiap file menjadi set menggunakan fungsi set(). Ini dilakukan untuk menghapus duplikat karena set tidak mengandung elemen duplikat.
- Menemukan kata-kata yang muncul dalam kedua file: Program ini mencari kata-kata yang muncul dalam kedua file dengan menggunakan metode intersection() dari tipe data set. Hasilnya adalah set dari kata-kata yang muncul dalam kedua file.
- Mencetak hasil: Program ini mencetak set dari kata-kata yang muncul dalam kedua file.
- Pemanggilan fungsi: Program ini memanggil fungsi munculKata('romeo.txt', 'romeo-full.txt') untuk menampilkan kata-kata yang muncul dalam kedua file teks tersebut.