



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71210695
Nama Lengkap	Cahaya Sampebua
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%)

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.

### Pengenalan dan Mendefinisikan Set

Set adalah salah satu tipe data dalam Python untuk menyimpan banyak data. Set sendiri bentuknya merupakan himpunan, sehingga semua anggotanya haruslah berbeda satu sama yang lain. Semua anggota dalam set harus bersifat immutable. Meskipun begitu, set sendiri merupakan tipe data mutable sehingga isinya bisa ditambah dan dikurangi. Oleh karena ini juga, tidak boleh ada set sebagai anggota set yang lain. Ada beberapa sifat dari tipe data set yaitu

- Set bersifat unordered alias tidak berurut, artinya kita tidak bisa menggunakan indeks untuk mengakses nilai

```
>>> a = {'a', 'b'}
>>> print(a[0])
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print(a[0])
TypeError: 'set' object is not subscriptable
```

Selain itu dalam set juga setelah kita mendefinisikan 5 anggota set dengan urutan 'a','b','c','d','e' , setelah itu kita print kita akan mendapatkan urutan yang berbeda

```
>>> a = {'a', 'b', 'c', 'd', 'e'}
>>> print(a)
{'b', 'd', 'c', 'a', 'e'}
```

- Set bersifat unchangable, yang berarti nilai yang sudah kita masukan tidak bisa diubah lagi, tetapi kita masih bisa menambahkan dan menghapus anggota pada set, dan karena sifat ini set hanya bisa menerima anggota dari tipe data yang bersifat immutable

```
>>> b = {1, 2, 3, [4, 5]}
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    b = {1, 2, 3, [4, 5]}
TypeError: unhashable type: 'list'
```

Kode diatas error karena tidak bisa memasukan list sebagai anggota karena list bersifat mutable

- Tidak bisa menerima nilai duplikat, jadi jika kita memasukan nilai yang sama pada suatu set maka nilai yang akan muncul hanya 1 saja

```
>>> a = {'a', 'a', 'c', 'd', 'e'}
>>> print(a)
{'a', 'e', 'd', 'c'}
```

Secara umum kita bisa membuat set dengan dua cara yaitu dengan kurung kurawal “{ }” atau menggunakan fungsi set()

Contoh menggunakan kurung kurawal :

```
>>> huruf = {'a', 'b', 'c', 'd'}
>>> print(huruf)
{'a', 'd', 'b', 'c'}
```

Contoh menggunakan fungsi set :

```
>>> warna = set(['hitam', 'merah', 'putih'])
>>> print(warna)
{'putih', 'hitam', 'merah'}
```

## Pengaksesan Set

Dalam list, kita bisa mengakses anggotanya dengan menggunakan indeks. Tidak demikian dengan set. Kita bisa mengakses anggota – anggota set dengan menggunakan perulangan for. Terkadang, data yang diakses mempunyai urutan yang berbeda dari data yang dimasukkan di dalam set. Hal ini terjadi karena urutan tidak dipermasalahkan pada set.

```
warna = {'hitam', 'merah', 'putih'}
print(len(warna)) # untuk mengetahui jumlah anggota jadi outputnya 3

for i in warna:
    print(i) # menampilkan set satu persatu
```

gambar 1

```
3
merah hitam putih
```

output gambar 1

Meskipun set tidak bisa diubah tetapi tetap bisa ditambah dan dihapus, kita bisa menambah anggota baru dengan menggunakan fungsi add() atau fungsi update()

```
warna = set()
warna.add('kuning') #menambah satu satu
print(warna)

warna.update({'hitam','merah','putih'}) #menambah lebih dari satu
print(warna)
```

*gambar 2*

```
{'kuning'}
{'putih', 'merah', 'kuning', 'hitam'}
```

*output gambar 2*

Untuk menghapus anggota dari set terdapat 4 cara yaitu dengan fungsi :

- remove(), untuk menghapus nilai yang dicari, jika nilai yang dicari tidak ada maka akan error

```
warna = {'hitam','merah','putih'}
warna.remove('hitam')
print(warna)

warna.remove('hijau')
print(warna)
```

```
{'putih', 'merah'}
Traceback (most recent call last):
  File "d:\kuli\prakAlpro\12_coba\cth1.py", line 5, in <module>
    warna.remove('hijau')
KeyError: 'hijau'
```

- discard(), untuk menghapus nilai yang dicari, jika nilai yang dicari tidak ada maka tidak akan error

```
warna = {'hitam','merah','putih'}
warna.discard('hitam')
print(warna)

warna.discard('hijau')
print(warna)
```

```
{'merah', 'putih'}
{'merah', 'putih'}
```

- `pop()`, mengambil salah satu dan menghapusnya dari set, akan terjadi error kalau set kosong

```
warna = {'hitam', 'merah', 'putih'}
warna.pop()
warna = {'hitam', 'merah', 'putih'}
warna.clear()
print(warna)

nama = {}
nama.clear()
print(nama)

Traceback (most recent call last):
set() "d:\kuli\prakAlpro\12_coba\cth1.py", line 6, in <module>
{}    na.pop()
TypeError: pop expected at least 1 argument, got 0
```

- `clear()`, menghapus seluruh element yang ada di dalam set

## Operasi – Operasi pada Set

Set dapat digunakan untuk melakukan operasi himpunan Matematika. Berikut daftar operasi pada

Set:

- Operator Union, menggabungkan dua buah set menjadi satu, dapat menggunakan operator palang (`|`) atau dengan menggunakan fungsi `union()`

```
a = {1, 2, 3, 4, 5}
b = {4, 5, 6, 7, 8}
print(a|b)

print(a.union(b))
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
{1, 2, 3, 4, 5, 6, 7, 8}
```

- Operator Intersection, menghasilkan irisan dari dua set, dapat menggunakan operator jangkar (&) atau dengan menggunakan fungsi intersection()

```
a = {1, 2, 3, 4, 5}
b = {4, 5, 6, 7, 8}
print(a&b)

print(a.intersection(b))
```

```
{4, 5}
{4, 5}
```

- Operator Difference, menghasilkan set baru yang merupakan selisih dari dua set yang dibandingkan, dapat menggunakan operator kurang (-) atau dengan menggunakan fungsi difference()

```
a = {1, 2, 3, 4, 5}
b = {4, 5, 6, 7, 8}
print(a-b)

print(a.difference(b))
```

```
{1, 2, 3}
{1, 2, 3}
```

- Operator Symmetric Difference, menghasilkan set baru yang merupakan jumlah dari dua set kecuali irisannya, dapat menggunakan operator (^) atau dengan menggunakan fungsi symmetric\_difference()

```
a = {1, 2, 3, 4, 5}
b = {4, 5, 6, 7, 8}
print(a^b)

print(a.symmetric_difference(b))
```

```
{1, 2, 3, 6, 7, 8}
{1, 2, 3, 6, 7, 8}
```

## 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.

Link GitHub → <https://github.com/cahaya-sampebua/Latihan-Mandiri-PrakALPro-12.git>

### SOAL 1

```
Tugas 1.py > ...
1  n = int(input('Masukkan Jumlah Kategori : '))
2  data_aplikasi = {}
3  for i in range(n):
4      nama_kategori = input('Masukkan Nama Kategori : ')
5      print('Masukkan 5 nama aplikasi di kategori',nama_kategori)
6      aplikasi = list()
7      for j in range(5):
8          nama_app = input('Masukkan Nama Aplikasi : ')
9          aplikasi.append(nama_app)
10     data_aplikasi[nama_kategori] = aplikasi
11 print(data_aplikasi)
12 daftar_aplikasi_list = list()
13
14 for aplikasi in data_aplikasi.values():
15     daftar_aplikasi_list.append(set(aplikasi))
16 print(daftar_aplikasi_list)
17 hasil = daftar_aplikasi_list
18
19 #print satu kategori saja
20 unik = daftar_aplikasi_list[0] ^ daftar_aplikasi_list[1]
21 print('Aplikasi yang hanya muncul di satu kategori saja adalah ',unik)
22
23 #print nama app yang muncul di dua kategori
24 if n>2:
25     inter = daftar_aplikasi_list[0] & daftar_aplikasi_list[1]
26     print('Aplikasi yang muncul di dua kategori sekaligus adalah ',inter)
```

## Output :

```
Masukkan Jumlah Kategori : 2
Masukkan Nama Kategori : Finance
Masukkan 5 nama aplikasi di kategori Finance
Masukkan Nama Aplikasi : RTI Saham
Masukkan Nama Aplikasi : Mirae
Masukkan Nama Aplikasi : IPOT
Masukkan Nama Aplikasi : Poems
Masukkan Nama Aplikasi : Calculator
Masukkan Nama Kategori : Utilities
Masukkan 5 nama aplikasi di kategori Utilities
Masukkan Nama Aplikasi : Photo
Masukkan Nama Aplikasi : Weather
Masukkan Nama Aplikasi : Calculator
Masukkan Nama Aplikasi : Camera
Masukkan Nama Aplikasi : Notes
{'Finance': ['RTI Saham', 'Mirae', 'IPOT', 'Poems', 'Calculator'], 'Utilities': ['Photo', 'Weather', 'Calculator', 'Camera', 'Notes']}
[{'Poems', 'Calculator', 'RTI Saham', 'IPOT', 'Mirae'}, {'Photo', 'Weather', 'Camera', 'Notes', 'Calculator'}]
Aplikasi yang hanya muncul di satu kategori saja adalah {'Camera', 'RTI Saham', 'Poems', 'Photo', 'Weather', 'Notes', 'IPOT', 'Mirae'}
```

## Penjelasan Source Code :

Baris 1 – 17 merupakan Source code dari contoh kategori kasus di PlayStore

Baris 20-21 merupakan variable untuk menampilkan nama-nama aplikasi yang hanya muncul di satu kategori disini kita memakai operator symmetric difference

Baris 24-26 kita buat suatu kondisi yaitu dimana inputan kategori dari user lebih dari 2 maka kita akan print nama aplikasi yang ada di dua kategori disini kita menggunakan operator Intersection



## SOAL 2

```
Tugas 2.py > ...
1  ulang = 'Y'
2  while ulang == 'Y':
3      print('***** Program Demonstrasi Konversi *****')
4      print('='*49)
5      print('Pilih jenis konversi :\n1. List ke Set\n2. Set ke List\n3. Tuple ke Set\n4. Set ke Tuple')
6      pilih = int(input('Masukkan pilihan anda : '))
7      print('='*49)
8      #list ke set
9      if pilih == 1:
10         finance = ['RTI Saham','IPOT','Poems','Mirae']
11         konversi = set(finance)
12         print(f'List ke Set\nSebelum : {finance}\nSesudah : {konversi}')
13     #set ke list
14     elif pilih == 2:
15         finance = {'RTI Saham','IPOT','Poems','Mirae'}
16         konversi = list(finance)
17         print(f'Set ke List\nSebelum : {finance}\nSesudah : {konversi}')
18     #tuple ke set
19     elif pilih == 3:
20         finance = ('RTI Saham','IPOT','Poems','Mirae')
21         konversi = set(finance)
22         print(f'Tuple ke Set\nSebelum : {finance}\nSesudah : {konversi}')
23     #set ke tuple
24     elif pilih == 4:
25         finance = {'RTI Saham','IPOT','Poems','Mirae'}
26         konversi = tuple(finance)
27         print(f'Set ke Tuple\nSebelum : {finance}\nSesudah : {konversi}')
28     else:
29         print('Pilihan tidak ada dalam menu')
30     print('='*49)
31     ulang = input('    Apakah anda ingin coba Lagi? Y/N : ').upper()
32     if ulang == "N":
33         print(f'\n                "Terimakasih"')
```

**Output :**

```
=====
Pilih jenis konversi :
1. List ke Set
2. Set ke List
3. Tuple ke Set
4. Set ke Tuple
Masukkan pilihan anda : 1
=====
List ke Set
Sebelum : ['RTI Saham', 'IPOT', 'Poems', 'Mirae']
Sesudah : {'Mirae', 'Poems', 'IPOT', 'RTI Saham'}
=====
    Apakah anda ingin coba Lagi? Y/N : n

    "Terimakasih"
```

### Penjelasan Source Code :

Baris 1-2 kita buat variable "Y" sebagai parameter untuk perulangan yang ada di baris 2

Baris 3-5 hanya mempercantik saja dengan print biasa dan baris 7 juga

Baris 6 buat variable inputan untuk user untuk menentukan jenis konversi apa yang user pilih

Baris 9-12 kita buat kondisi yaitu dimana Ketika input pilihan user = 1, maka buat variable list setelah itu buat juga variable dan pada variable itu kita masukan fungsi set untuk mengubah variable yang awalnya list dikonversi ke set, setelah itu print hasil sebelum dan sesudah.

Baris 14-17 kita buat kondisi dengan elif yaitu dimana Ketika input pilihan user = 2, maka buat variable set setelah itu buat juga variable dan pada variable itu kita masukan fungsi list untuk mengubah variable yang awalnya set dikonversi ke list, setelah itu print hasil sebelum dan sesudah.

Baris 19-22 kita buat kondisi dengan elif yaitu dimana Ketika input pilihan user = 3, maka buat variable tuple setelah itu buat juga variable dan pada variable itu kita masukan fungsi set untuk mengubah variable yang awalnya tuple dikonversi ke set, setelah itu print hasil sebelum dan sesudah.

Baris 24-27 kita buat kondisi dengan elif yaitu dimana Ketika input pilihan user = 4, maka buat variable set setelah itu buat juga variable dan pada variable itu kita masukan fungsi tuple untuk mengubah variable yang awalnya set dikonversi ke tuple, setelah itu print hasil sebelum dan sesudah.

Baris 28-29 kita buat kondisi selain dari kondisi diatas maka print pilihan tidak ada dalam menu.

Baris 30 hanya untuk mempercantik

Baris 31 kita buat variable yang sama seperti baris 1 untuk meminta user apakah dia ingin mencoba lagi atau tidak setelah itu kita tambahkan fungsi upper untuk mengantisipasi inputan user dengan huruf kecil

Baris 32-33 kita buat kondisi yaitu dimana inputan pada baris 31 = N maka print terimakasih

### SOAL 3

```
Tugas 3.py > ...
1  try:
2      teks1 = open(input('Masukkan Nama File 1 : '))
3      teks2 = open(input('Masukkan Nama File 2 : '))
4      read_teks1 = teks1.read().lower()
5      read_teks2 = teks2.read().lower()
6      print(f'Teks File 1 \n{read_teks1}\n\nTeks File 2 \n{read_teks2}')
7
8  except:
9      print('File tidak ditemukan/tidak bisa dibaca')
```

#### Output :

```
Masukkan Nama File 1 : file 1.txt
Masukkan Nama File 2 : file 2.txt
Teks File 1
tugas Selasa 14 Mei
harus sebisa mungkin
nyelesaain laprak prakalpro

Teks File 2
mencoba sebisanya
sajaaaa :)
```

#### Penjelasan Source Code :

Baris 1 untuk menguji apakah statement yang ada dalam baris 2-6 menghasilkan error atau tidak. Jika menghasilkan error, maka akan diganti dengan statement yang ada pada baris 8-9.

Baris 2-3 merupakan inputan user untuk file apa yang mau dia buka.

Baris 4-5 kita buat variable dengan method `read` untuk membaca seluruh teks dan kita ubah seluruh teksnya dengan fungsi `lower` untuk menghasilkan huruf kecil.

Baris 6 kita print hasil teks baris 4-5.