



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230987</b>
<b>Nama Lengkap</b>	<b>Prastha Pradipta Purusa</b>
<b>Minggu ke / Materi</b>	<b>13/ Tipe Data Set</b>

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 tipe data Python yang digunakan untuk menyimpan sekumpulan data yang semuanya unik dan biasa dikenal sebagai himpunan. Ciri-ciri set:

1. Isi dari Set disebut sebagai anggota.
2. Anggota dari Set harus bersifat immutable.
3. Set sendiri bersifat mutable, dapat menambah atau mengurangi isi dari sebuah Set. Set tidak dapat dimasukkan ke dalam Set

Untuk mendefinisikan Set menggunakan notasi {} dan fungsi set(). Contoh:

```
ganjil = {1,3,5,7,9}
genap = 0,2,4,6,8
genap = set(genap)
print(ganjil)
print(genap)
```

```
{1, 3, 5, 7, 9}
{0, 2, 4, 6, 8}
```

### Pengaksesan Set

Set tidak memiliki indeks dan tidak dapat langsung mengakses index. Contoh program:

```
44 isi = 'Hlo', 4632,45403.5
45 jumlah_isi = len(isi)
46 print(jumlah_isi)
47 for i in isi:
48     print(i)
```

```
3
Hlo
4632
45403.5
```

Pada tipe data Set, posisi tidak penting. Set adalah tipe data mutable yang berarti isinya bisa bertambah atau berkurang dengan add(). Contoh program:

```

50 angka = set()
51 angka.add(1)
52 angka.add(2)
53 angka.add(3)
54 print(len(angka))
55 print(angka)

```

```

3
{1, 2, 3}

```

Set dapat mengecek apakah elemen baru yang akan dimasukkan sudah ada di dalam Set (cek duplikasi). Jika belum ada, maka anggota tersebut bisa masuk ke dalam Set tetapi jika ada dalam Set sudah ada dengan nilai yang sama maka fungsi `add()` tidak akan menambah elemen dari Set. Untuk menghapus anggota dari sebuah Set dapat menggunakan `discard()`, `remove()`, `pop()` dan `clear()`. Perbedaan dari 4 fungsi tersebut:

<code>discard()</code>	<code>remove()</code>	<code>pop()</code>	<code>clear()</code>
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

```

56
57 isi = {'fksjfodsf', 8590438095, 47, 968}
58 isi.remove(47)
59 print(isi)

```

```

[Running] python -u "c:\Users\HP\p
{968, 8590438095, 'fksjfodsf'}

```

```

57 isi = {'fksjfodsf', 8590438095, 47, 968}
58 isi.discard(47)
59 print(isi)

```

```
[Running] python -u "c:\Users\HP\prasss\prasss.py"
{968, 8590438095, 'fksjfodsf'}
```

```
57 isi = {'fksjfodsf', 8590438095, 47, 968}
58 isipop = isi.pop()
59 print(isi)
60 print(isipop)
```

```
[Running] python -u "c:\Users\HP\prasss\prasss.py"
{8590438095, 'fksjfodsf', 47}
968
```

```
57 isi = {'fksjfodsf', 8590438095, 47, 968}
58 isi.clear()
59 print(isi)
```

```
[Running]
set()
```

Cara mengubah anggota di set:

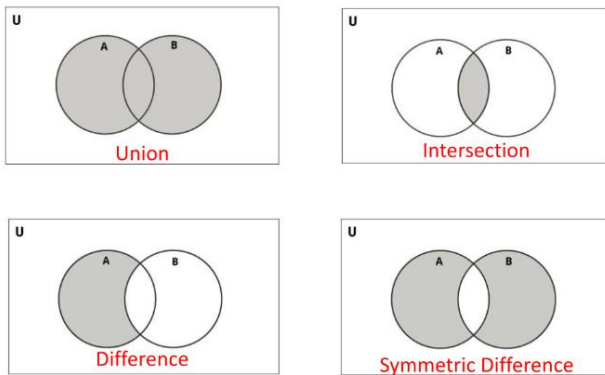
```
57 isi = {'fksjfodsf', 8590438095, 47, 968}
58 isi.remove('fksjfodsf')
59 isi.add('JDHSFJDSHFOH')
60 print(isi)
```

```
[Running] python -u "c:\Users\HP\prasss\prasss.py"
{968, 8590438095, 'JDHSFJDSHFOH', 47}
```

Harus menghapus nilai di elemen terlebih dahulu supaya nilai di elemen hilang kemudian diisi dengan add untuk menambah elemen baru. Setiap kali ada operasi penambahan dan penghapusan maka urutan anggota di dalam Set biasanya akan berubah.

## Operasi-Operasi pada Set

Operasi pada set:



Operator Union. Menggabungkan dua Set menjadi satu menggunakan operator `|` maupun fungsi `union()`. Union menggabungkan 2 set dan yang sama hanya akan tertulis 1 kali. Contoh program:

```
73 isi1 = {"halo", 'hai', "apa", "tiga"}
74 isi2 = {"aku", "dia", "halo", "apa"}
75 gabungan = isi1 | isi2
76 print(gabungan)
```

```
[Running] python -u "c:\Users\HP\prasss020904\l
{'hai', 'dia', 'halo', 'apa', 'tiga', 'aku'}
```

Operator Intersection. Menghasilkan irisan dari dua Set menggunakan operator `&` maupun fungsi `intersection()`. Intersection berarti anggota-anggota yang berada dalam set. Contoh program:

```
78 isi1 = {"halo", 'hai', "apa", "tiga"}
79 isi2 = {"aku", "dia", "halo", "apa"}
80 irisan = isi1 & isi2
81 print(irisan)
```

```
[Running] python -u
{'halo', 'apa'}
```

Operator Difference. Menghasilkan Set baru yang merupakan selisih dari dua Set yang dibandingkan menggunakan operator `-` maupun fungsi `difference()`. Operator symmetric difference akan menghasilkan Set baru yang merupakan gabungan dari dua Set tetapi tidak termasuk irisannya. Contoh program:

```
83 isi1 = {"halo", 'hai', "apa", "tiga"}
84 isi2 = {"aku", "dia", "halo", "apa"}
85 beda = isi1 - isi2
86 print(beda)
```

```
[Running] python -u  
{'hai', 'tiga'}
```

Operator Symmetric Difference. Menghasilkan Set baru yang merupakan jumlah dari dua Set kecuali irisannya menggunakan operator  $\wedge$  atau fungsi `symmetric_difference()`. Operator symmetric difference akan menghasilkan Set baru yang merupakan gabungan dari dua Set tetapi tidak termasuk irisannya. Contoh program:

```
88 isi1 = {"halo", 'hai', "apa", "tiga"}  
89 isi2 = {"aku", "dia", "halo", "apa"}  
90 beda = isi1 ^ isi2  
91 print(beda)
```

```
[Running] python -u "c:\Users\HPV  
{'hai', 'dia', 'aku', 'tiga'}
```

Dapat digunakan dengan cara lain seperti program ini:

```
98 isi1 = {"halo", 'hai', "apa", "tiga"}  
99 isi2 = {"aku", "dia", "halo", "apa"}  
100 isi = isi1 | isi2 - isi1 & isi2  
101 print(isi)
```

```
[Running] python -u "c:\Users\HP\prasss020904\ba  
{'tiga', 'aku', 'halo', 'hai', 'dia', 'apa'}
```

## 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

Jupyter Prastha Pradipta Purusa(71230987) Laprak 13 Last Checkpoint: 2 menit yang li

```
File Edit View Insert Cell Kernel Widgets Help
+ %< > - Run C Code

In [1]: n = int(input("Masukkan jumlah kategori: "))
data_aplikasi = {}
for i in range(n):
    nama_kategori = input("Masukkan nama kategori: ")
    aplikasi = []
    jumlah_aplikasi = int(input("Berapa banyak aplikasi di kategori: "))
    for j in range(jumlah_aplikasi):
        nama_aplikasi = input("Nama Aplikasi: ")
        aplikasi.append(nama_aplikasi)
    data_aplikasi[nama_kategori] = aplikasi
print("Data Aplikasi per Kategori:")
print(data_aplikasi)

hitung_aplikasi = {}

for kategori, aplikasi_list in data_aplikasi.items():
    for aplikasi in aplikasi_list:
        if aplikasi in hitung_aplikasi:
            hitung_aplikasi[aplikasi].add(kategori)
        else:
            hitung_aplikasi[aplikasi] = {kategori}
print("Aplikasi yang hanya muncul di satu kategori:")
for aplikasi, kategori_set in hitung_aplikasi.items():
    if len(kategori_set) == 1:
        print({aplikasi})

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

Masukkan jumlah kategori: 3
Masukkan nama kategori: Game
Berapa banyak aplikasi di kategori: 2
Nama Aplikasi: PUBG
Nama Aplikasi: ML
Masukkan nama kategori: Online shop
Berapa banyak aplikasi di kategori: 2
Nama Aplikasi: Tokopedia
Nama Aplikasi: Facebook
Masukkan nama kategori: Sosmed
Berapa banyak aplikasi di kategori: 3
Nama Aplikasi: Instagram
Nama Aplikasi: X
Nama Aplikasi: Facebook
Data Aplikasi per Kategori:
{'Game': ['PUBG', 'ML'], 'Online shop': ['Tokopedia', 'Facebook'], 'Sosmed': ['Instagram', 'X', 'Facebook']}
Aplikasi yang hanya muncul di satu kategori:
{'PUBG'}
{'ML'}
{'Tokopedia'}
{'Instagram'}
{'X'}
Aplikasi yang muncul di dua kategori:
{'Facebook'}
```

Line1: membuat input untuk jumlah kategori

Line2: membuat dictionary kosong untuk data\_aplikasi

Line3: Memulai loop yang akan berjalan sesuai inputan n

Line4: membuat input untuk nama kategori

Line5: Membuat list kosong aplikasi yang digunakan untuk menyimpan nama-nama aplikasi

Line6: membuat input untuk jumlah aplikasi yang ada di dalam kategori

Line7: Memulai loop yang akan berjalan sebanyak inputan jumlah\_aplikasi

Line8: memasukan input untuk nama aplikasi

Line9: menambahkan nama aplikasi ke dalam list aplikasi

Line10: Menyimpan list aplikasi ke dalam dictionary data\_aplikasi dengan kunci nama\_kategori

Line11: Memunculkan output teks "Data Aplikasi per Kategori:".

Line12: memunculkan output isi dictionary data\_aplikasi

Line13: Membuat dictionary kosong hitung\_aplikasi yang digunakan untuk menghitung berapa kategori yang dimiliki oleh setiap aplikasi.

Line14: loop yang berjalan melalui setiap item dalam data\_aplikasi. Kategori dijadikan keys dan aplikasi\_list sebagai value.

Line15: : Loop setiap aplikasi dalam aplikasi\_list.

Line16: Memeriksa apakah aplikasi sudah ada dalam hitung\_aplikasi.

Line17: Jika aplikasi sudah ada dalam hitung\_aplikasi maka ditambahkan kategori ke dalam set di aplikasi.y

Line18: kondisi lain

Line19: : Jika aplikasi belum ada di hitung\_aplikasi tambahkan aplikasi dengan keys berupa nama aplikasi dan value berupa set yang berisi kategori tersebut.

Line20: Memunculkan output teks "Aplikasi yang hanya muncul di satu kategori:"

Line21: Loop yang berjalan melalui setiap item dalam hitung\_aplikasi. aplikasi sebagai key dan kategori\_set adalah value

Line22: Memeriksa panjang kategori\_set adalah 1 yang berarti aplikasi dalam satu kategori

Line23: Memunculkan output aplikasi dalam bentuk set.

Line24: Mengecek apakah jumlah kategori (n) lebih dari 2.

Line25: Memunculkan output teks "Aplikasi yang muncul di dua kategori:" jika n lebih dari 2

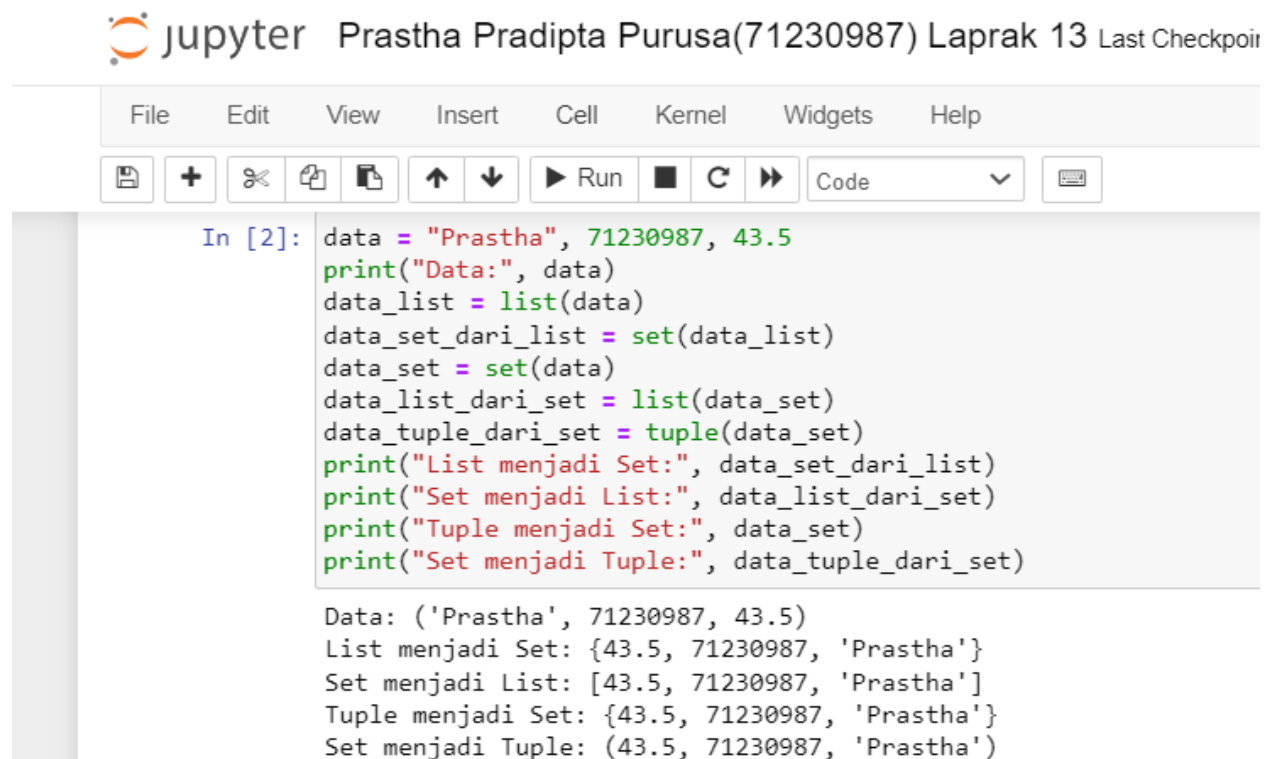
Line26: Loop setiap item dalam hitung\_aplikasi.

Line27: Memeriksa panjang kategori adalah 2 yang berarti aplikasi ada dalam dua kategori

Line28: Memunculkan output aplikasi dalam bentuk set.



## SOAL 2



The image shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "Prastha Pradipta Purusa(71230987) Laprak 13" and "Last Checkpoint". Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, undo, redo, running, and other functions. The main area contains a code cell with the following code:

```
In [2]: data = "Prastha", 71230987, 43.5
print("Data:", data)
data_list = list(data)
data_set_dari_list = set(data_list)
data_set = set(data)
data_list_dari_set = list(data_set)
data_tuple_dari_set = tuple(data_set)
print("List menjadi Set:", data_set_dari_list)
print("Set menjadi List:", data_list_dari_set)
print("Tuple menjadi Set:", data_set)
print("Set menjadi Tuple:", data_tuple_dari_set)
```

The output of the code is displayed below the code cell:

```
Data: ('Prastha', 71230987, 43.5)
List menjadi Set: {43.5, 71230987, 'Prastha'}
Set menjadi List: [43.5, 71230987, 'Prastha']
Tuple menjadi Set: {43.5, 71230987, 'Prastha'}
Set menjadi Tuple: (43.5, 71230987, 'Prastha')
```

Line1: isi data

Line2: memunculkan output teks "Data:" dan isi data

Line3: mengubah data menjadi list

Line4: menjadikan data adalah set yang sebelumnya adalah list

Line5: membuat data sebagai set

Line6: membuat data\_set menjadi list

Line7: membuat data\_set menjadi tuple

Line8: membuat output text "List menjadi Set" dengan data\_set\_dari\_list

Line9: membuat output text "Set menjadi List" dengan data\_list\_dari\_set

Line10: membuat output text "Tuple menjadi Set" dengan data\_set saja karena data awal sudah berbentuk tuple

Line11: : membuat output text "Set menjadi Tuple" dengan data\_tuple\_dari\_set

### SOAL 3

```
In [ ]: file1 = input("Masukkan nama file1: ")
file2 = input("Masukkan nama file2: ")
try:
    f1 = open(file1)
    bacafile1 = f1.read().lower()
    f1.close()
    print(bacafile1)
except:
    print("File tidak ditemukan/tidak bisa dibaca")
    bacafile1 = None
try:
    f2 = open(file2)
    bacafile2 = f2.read().lower()
    f2.close()
    print(bacafile2)
except:
    print("File tidak ditemukan/tidak bisa dibaca")
    bacafile2 = None
```

```
Masukkan nama file1: txt1.txt
Masukkan nama file2: txt2.txt
asdjoaofoadhsudhfwhdwo rkjekpjr
gjiperhgnfdijsbiufewiofxfjeipj dkad
```

```
Masukkan nama file1: txt1.txt
Masukkan nama file2: txt3.txt
asdjoaofoadhsudhfwhdwo rkjekpjr
File tidak ditemukan/tidak bisa dibaca
```

Line1 dan Line2 berfungsi untuk membuat input file1 dan file2. Menggunakan try except, nama f1,f2 untuk membaca file 1 dan file2 kemudian untuk membaca file menggunakan read() dan lower() untuk membuat semua huruf menjadi huruf kecil lalu di close untuk menutup file setelah dibaca dan memunculkan isi txt menggunakan print(bacafile1)/print(bacafile2). Menggunakan except jika file tidak ditemukan/tidak bisa dibaca dan None untuk menandakan tidak ada nilai.

Link github:

<https://github.com/prasss020904/Laprak13-71230987.git>