

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



<b>NIM</b>	<b>71231006</b>
<b>Nama Lengkap</b>	<b>Fransiskus Tedyanto seran</b>
<b>Minggu ke / Materi</b>	<b>12 / 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  
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

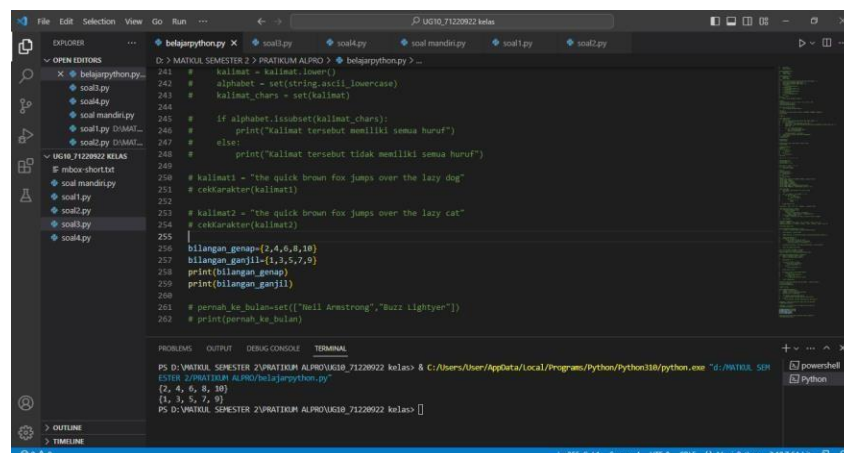
#### Pengenalan dan Mendefinisikan Set

Set pada python merupakan salah satu jenis struktur data yang digunakan untuk menyimpan kumpulan-kumpulan elemen unik. Set diimplementasikan sebagai himpunan (set) dari elemen-elemen yang tidak memiliki urutan tertentu. Dalam set juga setiap elemen harus unik, yang dimana ada duplikat dalam set tersebut.

Bberapa karakteristik set yang ada pada python yaitu:

1. Unik: set tidak dapat memiliki elemen yang duplikat.
2. Tidak berurutan: elemen-elemen yang ada dalam set tidak memiliki urutan tertentu.
3. Mutable: set dapat diubah setelah dibuat.
4. Dpat berisi elemen-elemen dengan tipe data yang berbeda: Set dapat berisi elemen-elemen dengan tipe data yang berbeda, seperti angka, string, tuple, dan lainnya.

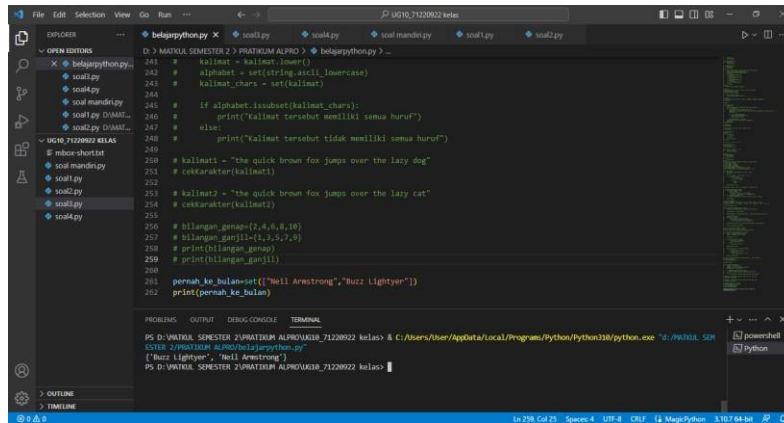
Untuk mendefinisikan set, ada beberapa cara yang dapat dilakukan yaitu dengan menggunakan notasi {} dan fungsi set() seperti contoh berikut:



```
241 # kalimat = kalimat.lower()
242 # alphabet = set(string.ascii_lowercase)
243 # kalimat_chars = set(kalimat)
244
245 # if alphabet.issubset(kalimat_chars):
246 #     print("kalimat tersebut memiliki semua huruf")
247 # else:
248 #     print("kalimat tersebut tidak memiliki semua huruf")
249
250 # kalimat1 = "the quick brown fox jumps over the lazy dog"
251 # cekKarakter(kalimat1)
252
253 # kalimat2 = "the quick brown fox jumps over the lazy cat"
254 # cekKarakter(kalimat2)
255
256 # bilangan_genap = (2, 4, 6, 8, 10)
257 # bilangan_ganjil = (1, 3, 5, 7, 9)
258 # print(bilangan_genap)
259 # print(bilangan_ganjil)
260
261 # pernah_ke_bulan = set(["Neil Armstrong", "Buzz Lightyear"])
262 # print(terakhir_ke_bulan)
```

Terminal Output:

```
PS D:\VIAKUL SEMESTER 2\PRATIUM ALPRO\UGR_71228922 kelas> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\VIAKUL SEMESTER 2\PRATIUM ALPRO\UGR_71228922 kelas\belajarpython.py"
(2, 4, 6, 8, 10)
(1, 3, 5, 7, 9)
PS D:\VIAKUL SEMESTER 2\PRATIUM ALPRO\UGR_71228922 kelas>
```



```
241 # kalimat = kalimat.lower()
242 # alphabet = set(string.ascii_lowercase)
243 # kalimat_chrs = set(kalimat)
244
245 # if alphabet.issubset(kalimat_chrs):
246 #     print("kalimat tersebut memiliki semua huruf")
247 # else:
248 #     print("kalimat tersebut tidak memiliki semua huruf")
249
250 # kalimat1 = "the quick brown fox jumps over the lazy dog"
251 # cekKarakter(kalimat1)
252
253 # kalimat2 = "the quick brown fox jumps over the lazy cat"
254 # cekKarakter(kalimat2)
255
256 # bilangan_genap=(2,4,6,8,10)
257 # bilangan_ganjil=(1,3,5,7,9)
258 # print(bilangan_genap)
259 # print(bilangan_ganjil)
260
261 pernah_ke_bulan=set(["Neil Armstrong","Buzz Lightyear"])
262 print(pernah_ke_bulan)
```

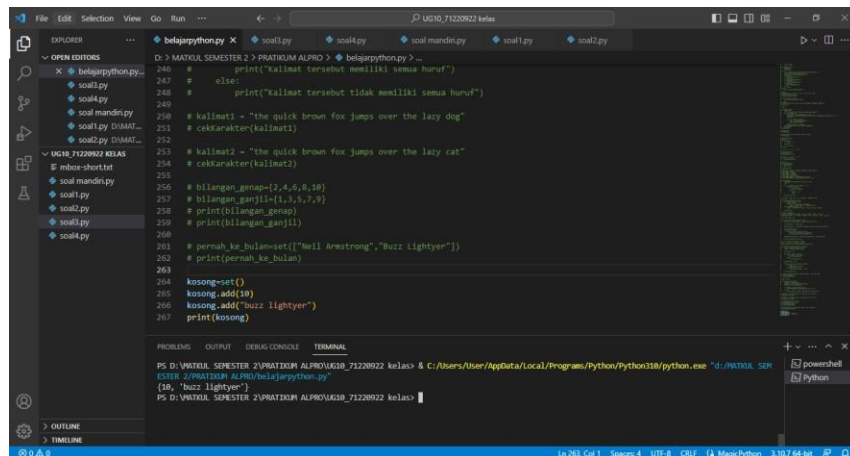
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\MATEKUL SEMESTER 2\PRATIUM ALPRO\UGUR 71220022 kelas> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\PRATIUM ALPRO\belajarython.py"

["Neil Lightyear", "Neil Armstrong"]

PS D:\MATEKUL SEMESTER 2\PRATIUM ALPRO\UGUR 71220022 kelas>

Untuk mendefinisikan set kosong anda tidak dapat menggunakan notasi {}, tetapi harus menggunakan fungsi set() seperti berikut merupakan contoh bagaimana cara membuat serta mengisi set kosong tersebut:



```
246 #     print("kalimat tersebut memiliki semua huruf")
247 #     else:
248 #         print("kalimat tersebut tidak memiliki semua huruf")
249
250 # kalimat1 = "the quick brown fox jumps over the lazy dog"
251 # cekKarakter(kalimat1)
252
253 # kalimat2 = "the quick brown fox jumps over the lazy cat"
254 # cekKarakter(kalimat2)
255
256 # bilangan_genap=(2,4,6,8,10)
257 # bilangan_ganjil=(1,3,5,7,9)
258 # print(bilangan_genap)
259 # print(bilangan_ganjil)
260
261 # pernah_ke_bulan=set(["Neil Armstrong","Buzz Lightyear"])
262 # print(pernah_ke_bulan)
263
264 kosong=set()
265 kosong.add(10)
266 kosong.add("Buzz Lightyear")
267 print(kosong)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\MATEKUL SEMESTER 2\PRATIUM ALPRO\UGUR 71220022 kelas> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\PRATIUM ALPRO\belajarython.py"

["Neil Lightyear", "Neil Armstrong"]

PS D:\MATEKUL SEMESTER 2\PRATIUM ALPRO\UGUR 71220022 kelas>

Contoh program di atas pertama saya menggunakan variabel kosong, dan memakai set() gunanya untuk membuat set kosong. Lalu saya menggunakan fungsi add yang gunanya untuk menambahkan value yang mau ditambahkan.

## MATERI 2

### Pengaksesan Set

Set tidak memiliki indeks, karena itu kita tidak dapat mengakses anggota-anggota yang ada pada sebuah set secara langsung menggunakan indeks. Ini merupakan contoh program:

```
D:\> MATKUL SEMESTER 2 > PRATIUM ALPRO > belajarpython.py > ...
250 # kalimat = "the quick brown fox jumps over the lazy dog"
251 # cekKarakter(kalimat)
252
253 # kalimat2 = "the quick brown fox jumps over the lazy cat"
254 # cekKarakter(kalimat2)
255
256 # bilangan_genap=[2,4,6,8,10]
257 # bilangan_ganjil=[1,3,5,7,9]
258 # print(bilangan_genap)
259 # print(bilangan_ganjil)
260
261 # pernah_ke_bulan=set(("Neil Armstrong","Buzz Lightyear"))
262 # print( pernah_ke_bulan)
263
264 # kosong=set()
265 # kosong.add(10)
266 # kosong.add("buzz lightyear")
267 # print(kosong)
268
269 nim=("71220922","71220923","71220924")
270 jumlah_nim=len(nim)
271 print(jumlah_nim)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO> python.exe "d:\matkul\_sem...  
ENTER 2\PRATIUM ALPRO\belajarpthon.py  
3  
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO>

Berikut contoh merupakan menampilkan set satu-persatu dengan menggunakan perulangan:

```
D:\> MATKUL SEMESTER 2 > PRATIUM ALPRO > belajarpython.py > ...
250 # kalimat = "the quick brown fox jumps over the lazy dog"
251 # cekKarakter(kalimat)
252
253 # kalimat2 = "the quick brown fox jumps over the lazy cat"
254 # cekKarakter(kalimat2)
255
256 # bilangan_genap=[2,4,6,8,10]
257 # bilangan_ganjil=[1,3,5,7,9]
258 # print(bilangan_genap)
259 # print(bilangan_ganjil)
260
261 # pernah_ke_bulan=set(("Neil Armstrong","Buzz Lightyear"))
262 # print( pernah_ke_bulan)
263
264 # kosong=set()
265 # kosong.add(10)
266 # kosong.add("buzz lightyear")
267 # print(kosong)
268
269 nim=("71220922","71220923","71220924")
270 for i in nim:
271     print(i)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO> python.exe "d:\matkul\_sem...  
ENTER 2\PRATIUM ALPRO\belajarpthon.py  
71220922  
71220923  
71220924  
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO>

Di dalam Set ada beberapa fungsi yang bisa digunakan contohnya jika kita memakai fungsi add yang dimana jika ada anggota dengan sebuah nilai yang sama maka fungsi tersebut tidak akan menambahkan nilai tersebut didalam set.

```
D:\> MATKUL SEMESTER 2 > PRATIUM ALPRO > belajarpython.py > ...
265 # kosong.add(10)
266 # kosong.add("buzz lightyear")
267 # print(kosong)
268
269 nim=("71220922","71220923","71220924")
270 # for i in nim:
271 #     print(i)
272
273 bilangan=set()
274 bilangan.add(10)
275 bilangan.add(20)
276 bilangan.add(30)
277 bilangan.add(40)
278 print(bilangan)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO> python.exe "d:\matkul\_sem...  
ENTER 2\PRATIUM ALPRO\belajarpthon.py  
[10, 20, 30]  
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO>

Untuk menghapus anggota dari sebuah set, ada beberapa cara yaitu dengan fungsi `discard()`, `remove()`, `pop()`, dan `clear()`. Berikut merupakan contoh program yang mendemonstrasikan penghapusan anggota dari sebuah set:

```

104 # kosong=set()
105 # kosong.add(10)
106 # kosong.add("buzz lightyear")
107 # print(kosong)
108
109 # nim=("71220922","71220923","71220924")
110 # for i in nim:
111 #     print(i)
112
113 # bilangan=set()
114 # bilangan.add(10)
115 # bilangan.add(20)
116 # bilangan.add(30)
117 # bilangan.add(10)
118 # print(bilangan)
119
120 bilangan_prima=[13,23,7,11,5]
121 bilangan_prima.remove(5)
122 print(bilangan_prima)
  
```

Terminal Output:

```

PS D:\MATRIK SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:\Users\User\AppData\Local\Programs\Python\Python10/python.exe "d:\MATRIK SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas\belajarpypython.py"
(23, 7, 11, 13)
PS D:\MATRIK SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
  
```

Ini merupakan beberapa fungsi-fungsi yang bisa digunakan untuk menghapus anggota dari set.

<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

Lalu ini merupakan beberapa contoh dari fungsi-fungsi gambar di atas:

```

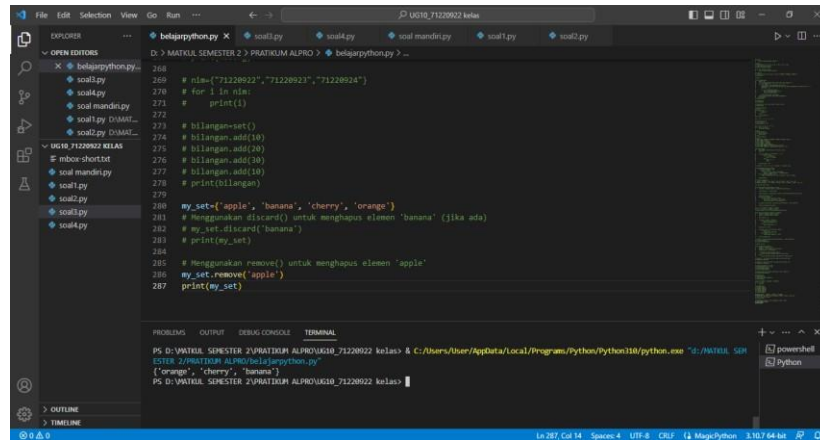
200 my_set={"apple", "banana", "cherry", "orange"}
201 # menggunakan discard() untuk menghapus elemen 'banana' (jika ada)
202 my_set.discard("banana")
203 print(my_set)
  
```

Terminal Output:

```

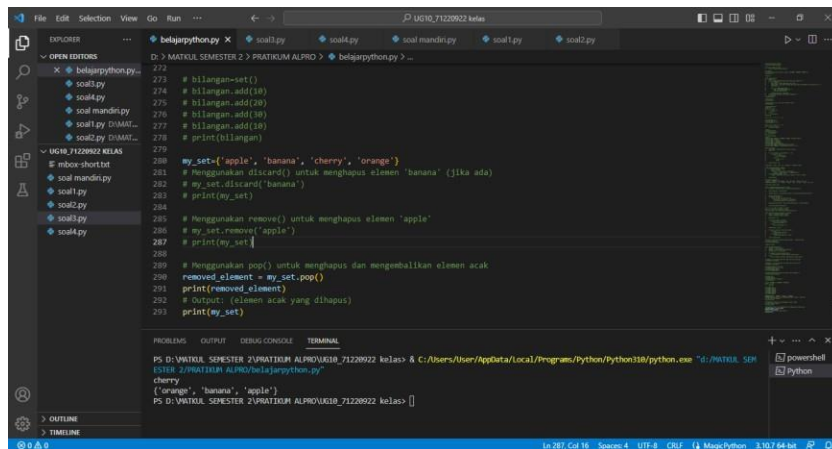
PS D:\MATRIK SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:\Users\User\AppData\Local\Programs\Python\Python10/python.exe "d:\MATRIK SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas\belajarpypython.py"
{'cherry', 'apple', 'orange'}
PS D:\MATRIK SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
  
```

*Fungsi Discard*



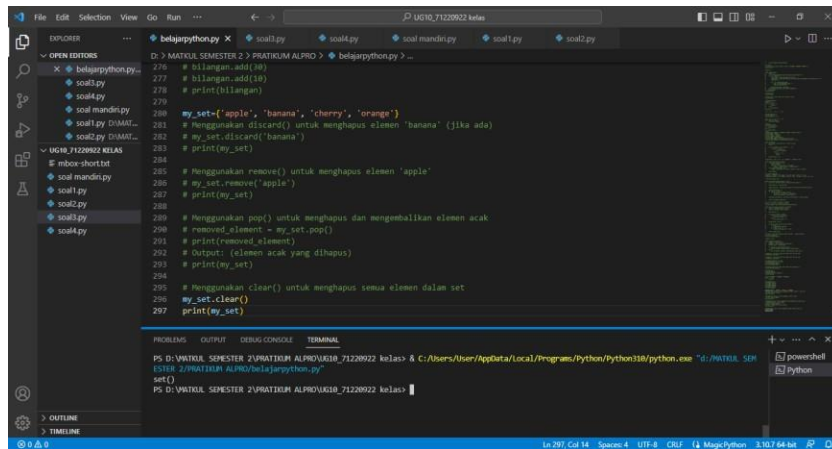
```
265 # MATRIKUL SEMESTER 2 > PRATIUM ALPRO > belajarpython.py > ...
266 # new("71220922","71220923","71220924")
267 # for i in nim:
268 #     print(i)
269
270 # bilangan=set()
271 # bilangan.add(10)
272 # bilangan.add(20)
273 # bilangan.add(30)
274 # bilangan.add(40)
275 # bilangan.add(50)
276 # print(bilangan)
277
278 my_set={'apple', 'banana', 'cherry', 'orange'}
279 # Menggunakan discard() untuk menghapus elemen 'banana' (jika ada)
280 # my_set.discard('banana')
281 # print(my_set)
282
283 # Menggunakan remove() untuk menghapus elemen 'apple'
284 # my_set.remove('apple')
285 # print(my_set)
286
287 PS D:\VITEK\SEMESTER 2\PRATIUM ALPRO\UE38_71220922 kelas>
PS D:\VITEK\SEMESTER 2\PRATIUM ALPRO\UE38_71220922 kelas> |
```

### Fungsi Remove



```
272 # bilangan=set()
273 # bilangan.add(10)
274 # bilangan.add(20)
275 # bilangan.add(30)
276 # bilangan.add(40)
277 # bilangan.add(50)
278 # print(bilangan)
279
280 my_set={'apple', 'banana', 'cherry', 'orange'}
281 # Menggunakan discard() untuk menghapus elemen 'banana' (jika ada)
282 # my_set.discard('banana')
283 # print(my_set)
284
285 # Menggunakan remove() untuk menghapus elemen 'apple'
286 # my_set.remove('apple')
287 # print(my_set)
288
289 # Menggunakan pop() untuk menghapus dan mengembalikan elemen acak
290 removed_element = my_set.pop()
291 print(removed_element)
292 # Output: (elemen acak yang dihapus)
293 print(my_set)
294
295 PS D:\VITEK\SEMESTER 2\PRATIUM ALPRO\UE38_71220922 kelas>
PS D:\VITEK\SEMESTER 2\PRATIUM ALPRO\UE38_71220922 kelas> |
```

### Fungsi Pop



```
276 # bilangan.add(10)
277 # bilangan.add(20)
278 # print(bilangan)
279
280 my_set={'apple', 'banana', 'cherry', 'orange'}
281 # Menggunakan discard() untuk menghapus elemen 'banana' (jika ada)
282 # my_set.discard('banana')
283 # print(my_set)
284
285 # Menggunakan remove() untuk menghapus elemen 'apple'
286 # my_set.remove('apple')
287 # print(my_set)
288
289 # Menggunakan pop() untuk menghapus dan mengembalikan elemen acak
290 removed_element = my_set.pop()
291 print(removed_element)
292 # Output: (elemen acak yang dihapus)
293 print(my_set)
294
295 # Menggunakan clear() untuk menghapus semua elemen dalam set
296 my_set.clear()
297 print(my_set)
298
299 PS D:\VITEK\SEMESTER 2\PRATIUM ALPRO\UE38_71220922 kelas>
PS D:\VITEK\SEMESTER 2\PRATIUM ALPRO\UE38_71220922 kelas> |
```

### Fungsi Clear

Fungsi discard() tidak akan menghasilkan error jika anggota yang ingin dihapus tidak ada di dalam set. Sedangkan fungsi pop() akan mengambil salah satu anggota (secara acak), dan mengeluarkannya dari set. Fungsi pop () akan berguna jika kita ingin memproses isi dari set satu-persatu



tanpa memperdulikan urutan/posisi dari setiap anggota yang ada di dalam set. Sedangkan fungsi clear untuk menghapus anggota-anggota yang ada dalam set, dan menjadi set kosong.

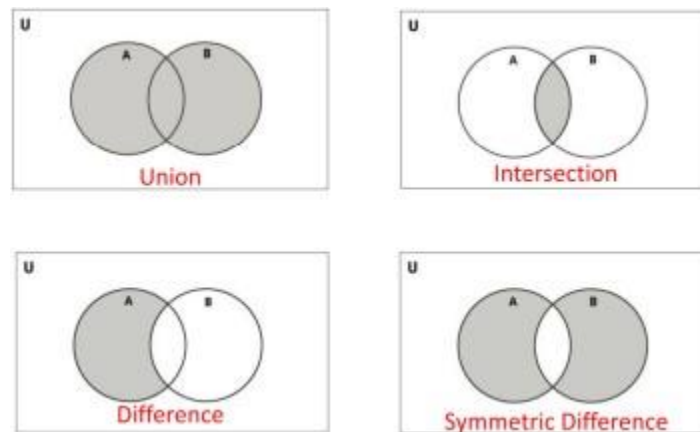
## MATERI 3

### Operasi-operasi pada set

Operasi-operasi pada set merupakan operasi-operasi pada himpunan. Berikut ini adalah daftar operasi set yang biasa digunakan pada python:

1. Operator Union: menggabungkan dua set menjadi satu.
2. Operator Intersection: Menghasilkan irisan dari dua set.
3. Operator Difference: Menghasilkan set baru yang merupakan selisih dari dua set yang dibandingkan.
4. Operator symmetric Difference: Menghasilkan set baru yang merupakan jumlah dari dua set kecuali irisannya.

Contoh gambar operator-operator pada set:



Berikut merupakan contoh program dari operasi-operasi pada set:

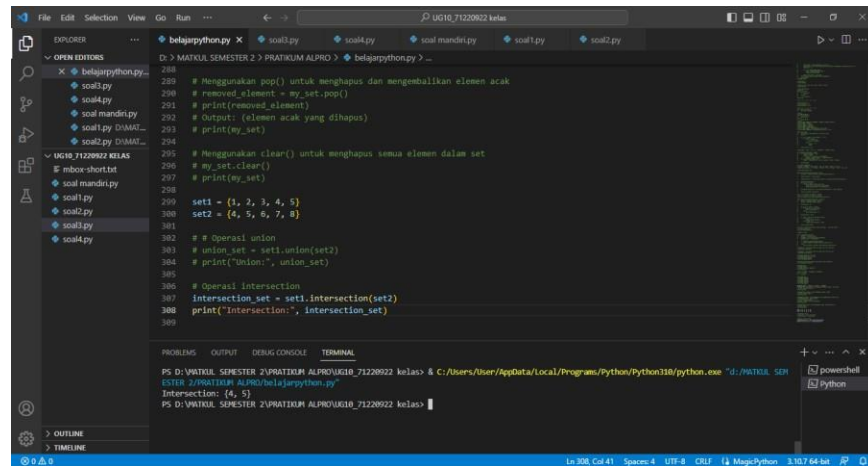
```
File Edit Selection View Go Run ...
D:\MATRIK SEMESTER 2 > PRATIUM ALPRIO > belajarpython.py
287 # print(my_set)
288
289 # Menggunakan pop() untuk menghapus dan mengembalikan elemen acak
290 # removed_element = my_set.pop()
291 # print(removed_element)
292 # Output: (elemen acak yang dihapus)
293 # print(my_set)
294
295 # Menggunakan clear() untuk menghapus semua elemen dalam set
296 # my_set.clear()
297 # print(my_set)
298
299 set1 = {1, 2, 3, 4, 5}
300 set2 = {4, 5, 6, 7, 8}
301
302 # Operasi union
303 union_set = set1.union(set2)
304 print("Union:", union_set)
305
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\MATRIK SEMESTER 2\PRATIUM ALPRIO\K3307220022 kelas> C:\Users\User\AppData\Local\Programs\Python\Python110\python.exe "D:\MATRIK SEMESTER 2\PRATIUM ALPRIO\belajarpthon.py"
Union: {1, 2, 3, 4, 5, 6, 7, 8}
PS D:\MATRIK SEMESTER 2\PRATIUM ALPRIO\K3307220022 kelas>
```

Operasi Union

Operasi union menggabungkan elemen-elemen dari kedua set menjadi satu set yang berisi semua elemen unik dari kedua set



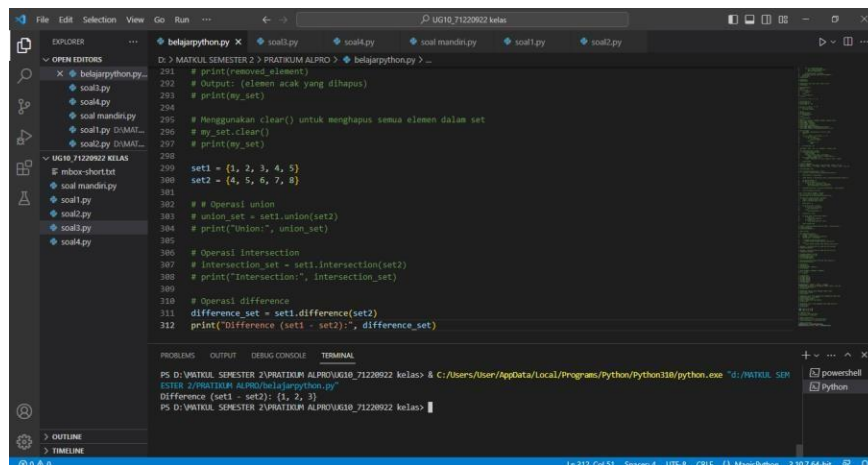
```
248
249 # Menggunakan pop() untuk menghapus dan mengembalikan elemen acak
250 removed_element = my_set.pop()
251 print(removed_element)
252 # Output: (elemen acak yang dihapus)
253 # print(my_set)
254
255 # Menggunakan clear() untuk menghapus semua elemen dalam set
256 my_set.clear()
257 # print(my_set)
258
259 set1 = {1, 2, 3, 4, 5}
260 set2 = {4, 5, 6, 7, 8}
261
262 # # Operasi union
263 union_set = set1.union(set2)
264 # print("Union:", union_set)
265
266 # Operasi Intersection
267 intersection_set = set1.intersection(set2)
268 print("Intersection:", intersection_set)
269
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VITEKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:\VITEKUL SEMESTER 2\PRATIUM ALPRO\belajarython.py"
Intersection: {4, 5}
PS D:\VITEKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
```

### Operasi Intersection

Operasi intersection menghasilkan set baru yang berisi elemen-elemen yang ada di kedua set.



```
291 # print(removed_element)
292 # Output: (elemen acak yang dihapus)
293 # print(my_set)
294
295 # Menggunakan clear() untuk menghapus semua elemen dalam set
296 my_set.clear()
297 # print(my_set)
298
299 set1 = {1, 2, 3, 4, 5}
300 set2 = {4, 5, 6, 7, 8}
301
302 # # Operasi union
303 union_set = set1.union(set2)
304 # print("Union:", union_set)
305
306 # Operasi Intersection
307 intersection_set = set1.intersection(set2)
308 # print("Intersection:", intersection_set)
309
310 # Operasi Difference
311 difference_set = set1.difference(set2)
312 print("Difference (set1 - set2):", difference_set)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VITEKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:\VITEKUL SEMESTER 2\PRATIUM ALPRO\belajarython.py"
Difference (set1 - set2): {1, 2, 3}
PS D:\VITEKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
```

### Operasi Difference

Operasi difference menghasilkan set baru yang berisi elemen-elemen yang ada di set pertama tetapi tidak ada di set kedua (untuk set pertama-set kedua) atau elemen-elemen yang ada di set kedua tetapi tidak di set pertama.



```
296 # my_set.clear()
297 # print(my_set)
298
299 set1 = {1, 2, 3, 4, 5}
300 set2 = {4, 5, 6, 7, 8}
301
302 # # Operasi union
303 # union_set = set1.union(set2)
304 # print("Union:", union_set)
305
306 # Operasi Intersection
307 # intersection_set = set1.intersection(set2)
308 # print("Intersection:", intersection_set)
309
310 # Operasi Difference
311 # difference_set = set1.difference(set2)
312 # print("Difference (set1 - set2):", difference_set)
313
314 # Operasi Symmetric Difference
315 symmetric_difference_set = set1.symmetric_difference(set2)
316 print("Symmetric Difference:", symmetric_difference_set)
```

PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10\_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/belajarpython.py"

Symmetric Difference: {1, 2, 3, 6, 7, 8}

### *Symetric Difference*

Operasi symmetric difference menghasilkan set baru yang berisi elemen-elemen yang ada di salah satu set tetapi tidak ada di kedua set.