

# Lab 7 Dasar-Dasar Pemrograman 1

## Set

**DEADLINE : 27 Oktober 2017 pukul 15:40 waktu Scele.**

Selamat datang di Tutorial Lab Dasar - Dasar Pemrograman yang ketujuh.

Dalam Tutorial kali ini, kalian akan mempelajari materi terkait Set.

*"A set is an unordered collection of objects, called elements or members of the set. A set is said to contain its elements. We write  $a \in A$  to denote that  $a$  is an element of the set  $A$ . The notation  $a \notin A$  denotes that  $a$  is not an element of the set  $A$ ."*

-Kenneth H. Rosen, Discrete Mathematics And Its Applications

---

### 1. Definisi Set

Set (himpunan) merupakan salah satu struktur data yang berisi kumpulan objek. Set seringkali dipakai untuk mengumpulkan data - data dalam jumlah tertentu / banyak menjadi sebuah kumpulan data yang unik dan memiliki elemen yang berbeda beda.

Ciri - ciri set antara lain :

- ☐ Elemen - elemen yang berada di dalam set bersifat *hashable*.
- ☐ Setiap elemen di dalam set adalah unik / tidak sama.
- ☐ Set merupakan sebuah kumpulan yang tidak memiliki urutan tertentu.
- ☐ Set bersifat *mutable*.

### 2. Inisiasi set

Terdapat dua jenis cara untuk menginisiasi / membuat sebuah set.

- a. Menginisiasi sebuah set kosong terlebih dahulu

```
a_set = set() #deklarasi set kosong
a_set.add(1) #menambahkan 1 ke dalam set
a_set.add('a') #menambahkan 'a' ke dalam set
a_set
>>> {1, 'a'}
```

b. Menginisiasi sebuah set langsung dengan isi dari set tersebut.

1. Daftar elemen didefinisikan secara eksplisit di dalam set.

```
a_set = {'a', 'n', 4, 2.0, 4, 2.0}
a_set
>>> {'n', 2.0, 4, 'a'}
```

2. `set(iterable)`

```
a_set = {'ano'}
a_set
>>> {'o', 'a', 'n'}
```

3. *Set comprehension*

```
a_set = {element for element in 'this is tutorial lab 7'}
a_set
>>> {'r', 'l', ' ', 'h', 'b', '7', 't', 'i', 'a', 's', 'u', 'o'}
```

### 3. Operasi pada Set

Beberapa operasi pada set pada umumnya antara lain

<code>len(a_set)</code>	: mengembalikan banyaknya elemen dari <code>a_set</code>
<code>x in a_set</code>	: mengembalikan nilai True atau False yang menandakan apakah elemen <code>x</code> terdapat di dalam <code>a_set</code>
<code>x not in a_set</code>	: mengembalikan nilai True atau False yang menandakan apakah elemen <code>x</code> tidak terdapat di dalam <code>a_set</code>
<code>min(a_set)</code>	: mengembalikan nilai terkecil dalam <code>a_set</code>
<code>max(a_set)</code>	: mengembalikan nilai terbesar dalam <code>a_set</code>
<code>sum(a_set)</code>	: mengembalikan hasil penjumlahan dari seluruh elemen di <code>a_set</code>
<code>a_set.isdisjoint(t)</code>	: mengembalikan nilai True atau False yang menandakan apakah <code>a_set</code> dan <code>t</code> memiliki elemen sekutu atau tidak.
<code>a_set.issubset(t)</code>	: mengembalikan nilai True atau False yang menandakan apakah setiap elemen dari <code>a_set</code> juga merupakan elemen dari <code>t</code> .

`a_set.issuperset(t)` : mengembalikan nilai True atau False yang menandakan apakah setiap elemen dari `t` juga merupakan elemen dari `s`.

Selain itu, terdapat beberapa operasi penting yang harus dipahami dalam Set, yaitu *union*, *intersection*, *difference*, *copy*, dan *symetric\_difference*.

#### 1. Copy

```
a_set = set('Matematika') #membuat sebuah set berisikan karakter - karakter pada kata Matematika
a_set
>>> {'k', 'e', 'M', 'm', 'i', 't', 'a'} #set yang telah dibuat sebelumnya
b_set = a_set.copy() #mengisi b_set dengan copy dari a_set
b_set
>>> {'k', 'e', 'M', 'm', 'i', 't', 'a'}
a_set.remove('M') #menghapus elemen 'M' di dalam a_set
a_set
>>> {'k', 'e', 'm', 'i', 't', 'a'}
b_set
>>> {'k', 'e', 'M', 'm', 'i', 't', 'a'} #elemen pada b_set tetap sama dan menghapus elemen pada a_set tidak mempengaruhi b_set yang berisi copy dari a_set
```

#### 2. Union / gabungan (|)

```
a_set = {1, 2}
b_set = {2, 4, 5}
c_set = a_set.union(b_set) #a_set | b_set
c_set
>>> {1, 2, 4, 5} #gabungan dari elemen pada a_set dan b_set tanpa ada duplikasi
```

#### 3. Intersection / Irisan (&)

```
a_set = {1, 2}
b_set = {2, 4, 5}
c_set = a_set.intersection(b_set) #a_set & b_set
c_set
>>> {2} #irisn dari elemen pada a_set dan b_set tanpa ada duplikasi
```

#### 4. Difference / Beda (-)

```
a_set = {1, 2}
b_set = {2, 4, 5}
c_set = a_set.difference(b_set) #a_set - b_set
c_set
>>> {1} #elemen pada a_set yang tidak terdapat pada b_set
```

#### 5. Symmetric Difference / Beda Simetrik

```
a_set = {1, 2}
b_set = {2, 4, 5}
c_set = a_set.symmetricdifference(b_set) #a_set ^ b_set
c_set
>>> {1, 4, 5} #gabungan dari elemen pada a_set dan b_set yang unik
(elemen pada a_set yang tidak terdapat pada b_set dan elemen pada
b_set yang tidak terdapat pada a_set)
```

# Benny's Plagiarism Filter

Benny membutuhkan bantuan kalian untuk membuat plagiarism check/filter untuk TP 3 dan TP 4! Benny mempunyai N set (S1, S2, ..., SN) yang masing-masing berisi file-file submisi tugas pemrograman, untuk mempermudah tugas kalian, masing-masing submisi tugas sudah direpresentasikan oleh hashcode 64 bit! Setiap submisi tugas yang berbeda akan memiliki hashcode yang berbeda dan dua submisi tugas yang memiliki hashcode sama berarti memiliki konten yang sama persis (plagiat). Benny menginginkan agar dari semua set submisi tugas, kalian membuat sebuah set baru yang berisi submisi tugas yang tidak plagiat. Dengan kata lain kalian harus membuat sebuah set S dimana setiap elemen pada S hanya muncul satu kali pada semua set yang Benny berikan. Benny menjamin bahwa pada setiap set yang ia berikan (S1, S2, ..., SN) sudah memenuhi properti dari set.

## Format input:

Baris pertama input berisi sebuah bilangan bulat N.

Untuk setiap N baris selanjutnya, terdapat sebuah set of 64 bit hashcode (integer 64 bit)

## Format output:

Output hanya terdiri dari satu baris yaitu set yang berisi submisi tugas yang tidak plagiat

## Contoh input 1:

2

{41251512, 25151512}

{25151512, 54676367}

## Contoh output 1:

{41251512, 54676367}

## Contoh input 2:

3

{127461734, 491273471, 328947101}

{162349322, 328947101, 234784376}

{213434432, 123432429, 127461734}

## Contoh output 2:

{491273471, 162349322, 234784376, 213434432, 123432429}

Hint:

Gunakan method `str.replace(find, replacement)` dan method `str.split(splitter)`