

# **PRAKTIKUM DASAR PEMROGRAMAN**

## **PERTEMUAN KE-8**

### **SET**

Set (himpunan) merupakan sebuah koleksi objek. Set merupakan list dengan syarat bahwa setiap elemennya harus unik (tidak boleh ada elemen yang sama). Dalam set urutan kemunculan elemen tidak diperhatikan.

#### **TUJUAN PRAKTIKUM**

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu merealisasikan set ke dalam bahasa pemrograman Python serta mengaplikasikan masalah/kasus yang perlu diselesaikan dengan menggunakan set.

#### **TOOLS**

Tools yang diperlukan untuk melakukan praktikum ini adalah interpreter Python yang telah terinstall di komputer.

#### **MATERI PRAKTIKUM:**

Set merupakan list dengan syarat bahwa setiap elemennya harus unik. Dalam set, urutan kemunculan elemen tidak diperhatikan. Himpunan A sama dengan himpunan B jika himpunan A memiliki elemen yang sama dengan B, meskipun urutan/posisi elemennya berbeda.

Pada praktikum ini mahasiswa akan belajar merealisasikan type kolektif set ke dalam bahasa Python dalam paradigma fungsional. Dalam paradigma fungsional, set didefinisikan sebagaimana list dengan syarat tambahan bahwa setiap elemennya harus unik. Oleh karena itu, set juga dapat didefinisikan secara rekursif seperti pada list.

Meskipun bahasa Python memiliki type bawaan berupa set, namun set dalam praktikum ini akan tetap direalisasikan (dalam paradigma fungsional/ secara rekursif) sebagai list dengan syarat khusus sebagaimana yang dipelajari dalam kuliah atau Diklat.

Semua konstruktor, selektor, dan fungsi lainnya yang telah dibuat pada list dapat digunakan di dalam set.

Berikut adalah langkah-langkah pada praktikum ini:

1. Buatlah sebuah file python bernama `set_<nim>.py`, contohnya: `set_24060119110023.py`.
2. Tuliskan nama file, deskripsi, pembuat, dan tanggal pada bagian awal file Anda sebagai komentar.
3. Agar dapat menggunakan konstruktor, selektor, dan fungsi yang telah dibuat pada list, lakukan import dengan menambahkan baris kode seperti berikut:

```
from list import *
```

File program list dan set berada dalam satu folder yang sama. Ubah nama 'list' sesuai nama file program list masing-masing.

4. Berikut adalah contoh realisasi sebuah fungsi digunakan dalam set dalam bahasa Python.

```

set.py
1  #Nama File: set.py
2  #Deskripsi: berisi type dan operasi set yang menggunakan list
3  #Pembuat:
4  #Tanggal:
5
6  #DEFINISI DAN SPESIFIKASI TYPE
7  # Set adalah sebuah list dengan syarat setiap elemen harus unik
8  # Semua konstruktor, selektor, dan operasi yang telah didefinisikan
9  # untuk list juga berlaku pada set
10
11  from list import *
12
13  #DEFINISI DAN SPESIFIKASI OPERASI LIST YANG DIPERLUKAN UNTUK HIMPUNAN
14
15  # Rember: elemen, list -> list
16  # Rember(x,L) menghapus sebuah elemen x dari list L
17  # Jika x ada di list L, maka elemen L berkurang 1.
18  # Jika x tidak ada di list L maka L tetap.
19  # List kosong tetap menjadi list kosong.
20  def rember(x,L):
21      if(isEmpty(L)):
22          return []
23      else:
24          if(firstElmt(L) == x):
25              return tail(L)
26          else:
27              return konsor(firstElmt(L),rember(x,tail(L)))

```

5. Setiap kali merealisasikan sebuah fungsi, aplikasikan fungsi tersebut, sehingga dapat diketahui apakah realisasi sudah sesuai dengan spesifikasi yang diharapkan.
- Contoh aplikasi fungsi berikut dipanggil di dalam fungsi `print()` agar hasil aplikasi atau output dapat langsung ditampilkan pada output program.

```

60  #APLIKASI
61  print(rember(2,[]))
62  print(rember(2,[3]))
63  print(rember(2,[3,4,5,2,3,5,7,8,2]))

```

Berikut adalah output dari aplikasi fungsi tersebut:

```

[]
[3]
[3, 4, 5, 3, 5, 7, 8, 2]

```

6. Kerjakan fungsi-fungsi berikutnya yang ada pada bagian latihan.
7. Pada akhir sesi praktikum, laporkan hasil praktikum pada link berikut <https://bit.ly/daspro24lap>.

## TUGAS/LATIHAN

Buatlah realisasi python untuk fungsi-fungsi lainnya sebagai berikut:

1. Realisasi fungsi `Rember(x,L)` yang dicontohkan akan menghapus elemen `x` yang ditemui pertama kali dari list `L`. Buatlah realisasi versi 2 untuk menghapus elemen `x` yang ditemui terakhir kali pada list `L`, beri nama dengan `Rember2(x,L)`.
2. Buatlah realisasi dari fungsi `MultiRember(x,L)`.
3. Buatlah realisasi fungsi `MakeSet(L)` dalam dua versi:
  - a. Versi 1: dengan memanfaatkan fungsi `IsMember(x,L)` untuk mengecek duplikasi elemen.
  - b. Versi 2: dengan memanfaatkan fungsi `MultiRember(x,L)` untuk menghapus duplikasi elemen.Perhatikan output yang dihasilkan dari kedua versi tersebut. Bagaimana perbedaannya?
4. Buatlah realisasi fungsi `KonsoSet(e,H)` yang menambahkan sebuah elemen `e` sebagai elemen pertama set `H` dengan syarat `e` belum ada di dalam himpunan `H`.
5. Buatlah realisasi fungsi `IsSet(L)`.
6. Buatlah realisasi fungsi `IsSubset(H1,H2)`
7. Buatlah realisasi fungsi `IsEqualSet` dalam dua versi:
  - a. Versi 1: memanfaatkan fungsi `IsSubset(H1,H2)`.
  - b. Versi 2: dengan mengecek satu persatu elemen pada `H1` dan `H2`.
8. Buatlah realisasi fungsi `IsIntersect(H1,H2)`.
9. Buatlah realisasi fungsi `MakeIntersect(H1,H2)` dalam 2 versi:
  - a. Versi 1: rekursif terhadap `H1`
  - b. Versi 2: rekursif terhadap `H2`Perhatikan output yang dihasilkan dari kedua versi tersebut. Bagaimana perbedaannya?
10. Buatlah realisasi fungsi `MakeUnion(H1,H2)` dalam 2 versi:
  - a. Versi 1: rekursif terhadap `H1`
  - b. Versi 2: rekursif terhadap `H2`Perhatikan output yang dihasilkan dari kedua versi tersebut. Bagaimana perbedaannya?
11. Buatlah realisasi fungsi `NBIntersect(H1,H2)`.
12. Buatlah realisasi fungsi `NBUnion(H1,H2)`.

Untuk setiap fungsi yang telah direalisasikan buatlah aplikasinya dengan berbagai variasi input untuk menguji apakah output fungsi sudah sesuai dengan yang seharusnya.

### #DEFINISI DAN SPESIFIKASI OPERASI LIST YANG DIPERLUKAN UNTUK HIMPUNAN

```
# Rember: elemen, list -> list
# Rember(x,L) menghapus sebuah elemen x dari list L
# Jika x ada di list L, maka elemen L berkurang 1.
# Jika x tidak ada di list L maka L tetap.
# List kosong tetap menjadi list kosong.
```

```
# MultiRember: elemen, list -> list
# MultiRember(x,L) menghapus semua kemunculan elemen x dari list L.
# List baru yang dihasilkan tidak lagi memiliki elemen x.
# List kosong tetap menjadi list kosong.
```

### #DEFINISI DAN SPESIKASI KONSTRUKTOR SET DARI LIST

```
# MakeSet: list -> set
# MakeSet(L) membuat set dari list L dengan menghapus semua kemunculan lebih dari satu kali
# list kosong tetap menjadi himpunan kosong
```

#### **#DEFINISI DAN SPESIKASI KONSTRUKTOR SET**

```
# KonsoSet: elemen,set -> set
# konsoSet(e,H) menambahkan sebuah elemen e sebagai elemen pertama set H
# dengan syarat e belum ada di dalam himpunan H
```

#### **#DEFINISI DAN SPESIFIKASI PREDIKAT**

```
# IsSet: list -> boolean
# IsSet(L) mengembalikan true jika L adalah sebuah set

# IsSubset: 2 set -> boolean
# IsSubset(H1,H2) mengembalikan true jika H1 merupakan subset dari H2

# IsEqualSet: 2 set -> boolean
# IsEqualSet(H1,H2} benar jika H1 adalah set yang sama dengan H2

# IsIntersect: 2 set -> boolean
# IsIntersect(H1,H2) benar jika H1 beririsan dengan H2
```

#### **#DEFINISI DAN SPESIFIKASI OPERASI TERHADAP HIMPUNAN**

```
# MakeIntersect: 2 set -> set
# MakeIntersect(H1,H2) menghasilkan set baru yang merupakan hasil irisan antara H1 dan H2

# MakeUnion: 2 set -> set
# MakeUnion(H1,H2) menghasilkan set baru yang merupakan hasil gabungan antara H1 dan H2

# NBIntersect: 2 set -> integer
# NBIntersect(H1,H2) menghasilkan jumlah elemen yang beririsan pada H1 dan H2
# tanpa memanfaatkan fungsi MakeIntersect(H1,H2).

# NBUnion: 2 set -> integer
# NBUnion(H1,H2) menghasilkan jumlah elemen hasil gabungan antara H1 dan H2
# tanpa memanfaatkan fungsi MakeUnion(H1,H2).
```

\*\*\*\*\*Selamat Mengerjakan dan Berlatih \*\*\*\*\*