

**LAPORAN PRAKTIKUM**  
**MATA KULIAH TEORI ALGORITMA STRUKTUR DATA**

Dosen Pengampu : Triana Fatmawati, S.T, M.T

**PERTEMUAN 15 : GRAPH**



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2024**

1. Jelaskan perbedaan dan persamaan antara List dan Set. Kemudian dalam kasus apa ketika menggunakan kedua struktur data tersebut?

**Jawab :**

List dan Set adalah dua jenis struktur data yang umum digunakan dalam pemrograman dan keduanya merupakan bagian dari Java Collections Framework. Berikut adalah beberapa perbedaan dan persamaan antara keduanya:

A. Persamaan:

- 1) Keduanya digunakan untuk menyimpan elemen.
- 2) Keduanya dapat menyimpan elemen null.
- 3) Keduanya memiliki metode untuk menambah, menghapus, dan mengiterasi elemen.

B. Perbedaan:

- 1) Duplikasi Elemen: List dapat menyimpan elemen duplikat, sedangkan Set tidak dapat. Set hanya menyimpan elemen unik.
- 2) Urutan: List adalah struktur data berurutan, yang berarti elemen ditambahkan berdasarkan urutan mereka. Set, di sisi lain, adalah kumpulan elemen yang tidak berurutan.
- 3) Indeks: Dalam List, elemen dapat diakses dengan indeks mereka. Set tidak memiliki indeks.

C. Gunakan List ketika:

- 1) Menyimpan elemen dalam urutan tertentu.
- 2) Mengakses elemen berdasarkan posisi mereka.
- 3) Menyimpan elemen duplikat.

D. Gunakan Set ketika:

- 1) Kita perlu memastikan bahwa elemen yang disimpan adalah unik.
- 2) Urutan elemen tidak penting.
- 3) Kita tidak perlu mengakses elemen dengan indeks.

2. Apakah perbedaan dari fungsi clear() dan removeAll() pada interface List dan jelaskan alasannya?

**Jawab :** Fungsi clear() dan removeAll() pada interface List di Java memiliki tujuan yang sama yaitu untuk menghapus elemen, tetapi cara mereka bekerja dan penggunaannya berbeda. Method clear() digunakan untuk menghapus semua elemen dari List. Setelah method ini dipanggil, List akan kosong, kita tidak perlu memberikan argumen apa pun ke method ini.

Contoh :

```
list<String> list = new ArrayList<>();  
list.add("A");  
list.add("B");  
list.clear(); // Menghapus semua elemen
```

metode `removeAll(Collection c)` digunakan untuk menghapus semua elemen dalam koleksi yang diberikan dari List. Kita perlu memberikan koleksi elemen yang ingin dihapus sebagai argumen.

Contoh :

```
List<String> list = new ArrayList<>();
list.add("A");
list.add("B");
list.add("C");

List<String> removeElements = new ArrayList<>();
removeElements.add("A");
removeElements.add("B");

list.removeAll(removeElements); // Menghapus "A" dan "B" dari list
```

3. Apakah perbedaan fungsi `add()` dan `offer()` pada Queue, jelaskan?

**Jawab :** Perbedaan utama antara `add()` dan `offer()` adalah bagaimana mereka menangani kondisi ketika queue sudah penuh. `add()` melempar pengecualian, sedangkan `offer()` mengembalikan nilai boolean. Oleh karena itu, `offer()` biasanya lebih disukai ketika bekerja dengan kapasitas queue yang tetap, karena kita dapat dengan mudah mengecek apakah elemen berhasil ditambahkan atau tidak.

4. Menurut Anda, mengapa Java tidak spesifik memiliki class yang menangani operasi-operasi Graph?

**Jawab :**

- 1) Kompleksitas: Graph adalah struktur data yang kompleks dengan banyak variasi dan operasi. Ada berbagai jenis graph (misalnya, directed, undirected, weighted, unweighted) dan setiap jenis memiliki operasi khusus.
- 2) Fleksibilitas: Dengan tidak menyediakan class bawaan untuk Graph, Java memberikan fleksibilitas kepada pengguna untuk mendefinisikan dan mengimplementasikan Graph sesuai dengan kebutuhan mereka. Pengguna dapat memilih struktur data yang paling sesuai untuk merepresentasikan Graph (misalnya, adjacency matrix, adjacency list, edge list, dll.) dan mengimplementasikan operasi yang mereka butuhkan.
- 3) Library Pihak Ketiga: Ada banyak library pihak ketiga yang tersedia yang menyediakan implementasi Graph yang efisien dan fitur-fitur lanjutan. Beberapa contoh termasuk JGraphT, Guava, dan Apache Commons Graph. Dengan menggunakan library ini, pengguna dapat mendapatkan keuntungan dari kode yang sudah diuji dan dioptimalkan, sambil tetap memiliki fleksibilitas untuk memilih library yang paling sesuai dengan kebutuhan mereka.

5. Jelaskan pengertian tipe data generic!

**Jawab :** Tipe data Generic adalah fitur dalam pemrograman yang memungkinkan kita untuk menulis dan menggunakan algoritma yang dapat bekerja pada berbagai jenis data. Dengan kata lain, mereka memungkinkan kita untuk membuat kode yang independen dari tipe data.

Dalam konteks Java, Generic adalah kelas atau interface yang dapat bekerja dengan berbagai jenis objek. Mereka didefinisikan dengan tipe placeholder (biasanya ditulis sebagai satu huruf kapital, seperti E, T, K, V, dll.), yang kemudian dapat digantikan dengan tipe aktual saat kita menggunakan kelas atau interface tersebut.

Contoh sederhana dari Generic adalah kelas ArrayList di Java. Kita dapat membuat ArrayList yang dapat menyimpan objek dari jenis apa pun, seperti Integer, String, atau customisasi sendiri:

6. Mengapa instansiasi pada collections dapat menggunakan konstruktor yang berbeda (bukan class ybs)!

**Jawab :** Dalam Java, ketika kita mendeklarasikan sebuah collection, biasanya menggunakan interface collection (seperti List, Set, atau Map) sebagai tipe, tetapi ketika kita menginstansiasi collection tersebut, kita menggunakan class yang mengimplementasikan interface tersebut (seperti ArrayList, HashSet, atau HashMap). Manfaat dari pendekatan tersebut adalah fleksibilitas. Dengan menggunakan interface sebagai tipe, kita dapat dengan mudah mengubah implementasi yang digunakan tanpa perlu mengubah kode lainnya. Misalnya, kita dapat mengubah ArrayList menjadi LinkedList atau HashSet menjadi TreeSet hanya dengan mengubah baris kode di mana kita menginstansiasi collection.