

Pengantar Collection Type

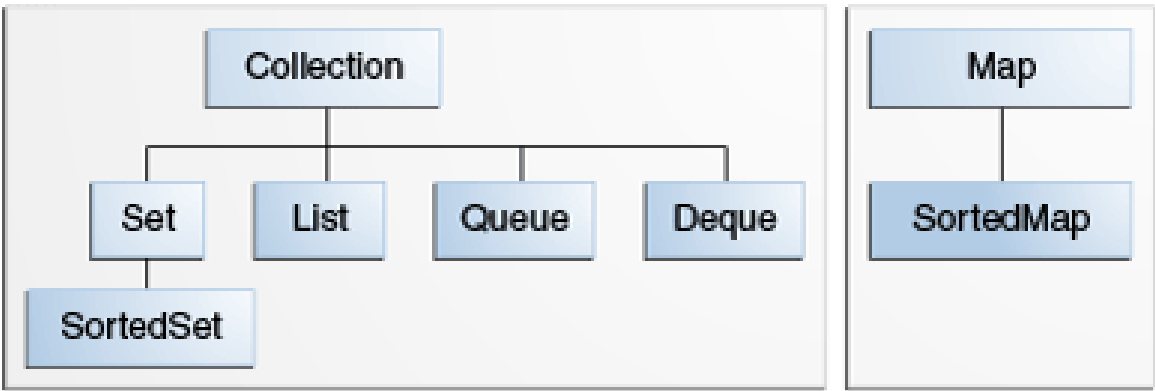
Collection adalah kumpulan interface yang digunakan sebagai wadah untuk mengumpulkan beberapa elemen menjadi satu kesatuan. Dari definisi tersebut bisa ditafsirkan Collection mirip dengan Array karena keduanya merepresentasikan struktur data. Java Collections Framework terdiri dari tiga komponen, yaitu:

- 1. Interface: Tipe abstrak dari Collection yang membentuk suatu hierarki.
- 2. Implementation: Tipe konkrit yang mengimplementasikan Interface dari Collection serta dan merupakan suatu struktur data yang bisa langsung digunakan.
- 3. Algorithm: Ini adalah method yang melakukan komputasi berguna, misal *sorting* atau *searching*.

Dari komponen pembentuk Java Collections Framework bisa kita lihat salah satu perbedaan Collection dengan Array. Misalnya Array tidak mempunyai method untuk *sorting* atau jumlah elemen di Array tidak bisa dinamis. Sementara Collection bisa dinamis.

Dapat disimpulkan bahwa Collection adalah struktur data yang lebih kompleks dan lebih canggih dibanding Array. Untuk itu setiap *programmer* wajib menguasai Java Collections Framework. Sebabnya, kita akan selalu berurusan dengan data saat membuat aplikasi.

Core Collection Interface dapat dirangkum pada gambar di bawah ini.



Gambar diambil dari <https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html>

Dalam materi ini kita hanya akan membahas Java Collections Framework yang paling sering digunakan yaitu List, Set, dan Map. Sebagian Implementasi dari ketiga *interface* tersebut bisa kita lihat di tabel di bawah ini.

List	Set	Map
ArrayList	TreeSet	TreeMap
LinkedList	HashSet	HashMap
	LinkedHashSet	LinkedHashMap

Codelab Collection

Untuk dapat melihat hierarki dari Java Collections Framework secara lebih lengkap kita bisa menggunakan fitur Type Hierarchy di IntelliJ.

1. Buatlah proyek baru dengan nama Collection dengan nama package

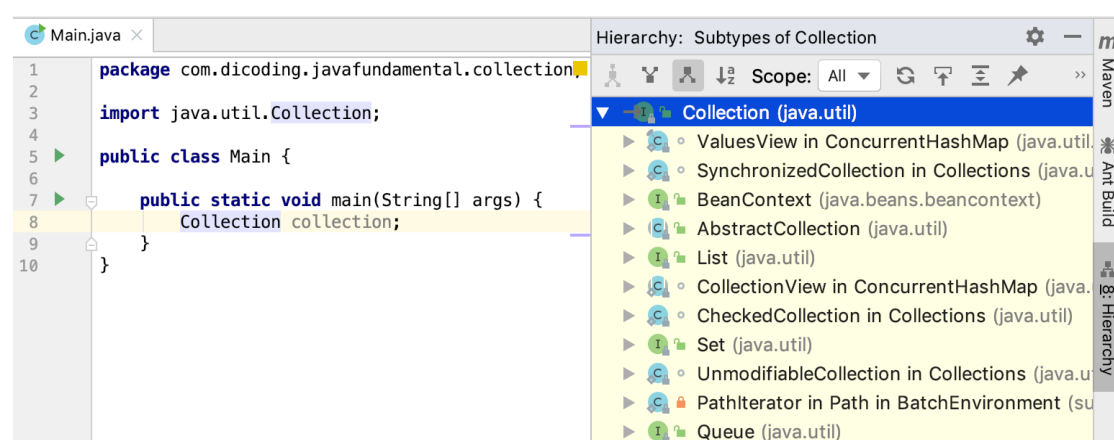
`com.dicoding.javafundamental.collection` di dalamnya:



2. Buatlah sebuah kelas baru di dalamnya dengan nama `Main`, kemudian tambahkan kode berikut:

```
1. package com.dicoding.javafundamental.collection;
2.
3. import java.util.Collection;
4.
5. public class Main {
6.
7.     public static void main(String[] args) {
8.         Collection collection;
9.     }
10. }
```

3. Blok deklarasi interface Collection lalu tekan ctrl + h (atau melalui menu Navigate - Type Hierarchy). Akan muncul panel Hierarchy dari interface Collection.



4. Perhatikan panel Hierarchy. Ada banyak *subtype* atau turunan dari interface Collection. Masih ingat dengan materi inheritance kan? Fitur Type Hierarchy di IntelliJ sangat membantu programmer untuk melihat struktur atau hierarki inheritance dari interface atau class. Selain itu ada fitur Method Hierarchy untuk melihat struktur inheritance dari method dan Call Hierarchy yang fungsinya untuk melihat di mana saja suatu method dipanggil (*invoked*).

5. Di panel Hierarchy terlihat interface Collection mempunyai turunan, salah satunya List dan Set sesuai dengan gambar Core Collection Interface di atas.

OK mari kita bahas satu per satu dari ketiga interface yang sering digunakan yaitu List, Set, dan Map.

[< Sebelumnya](#)

[Selanjutnya >](#)