

Materi IV

Nama : Roy Agus Martin Marbun

Kelas : XII – RPL

Mata Pelajaran : Pemrograman Berorientasi Objek

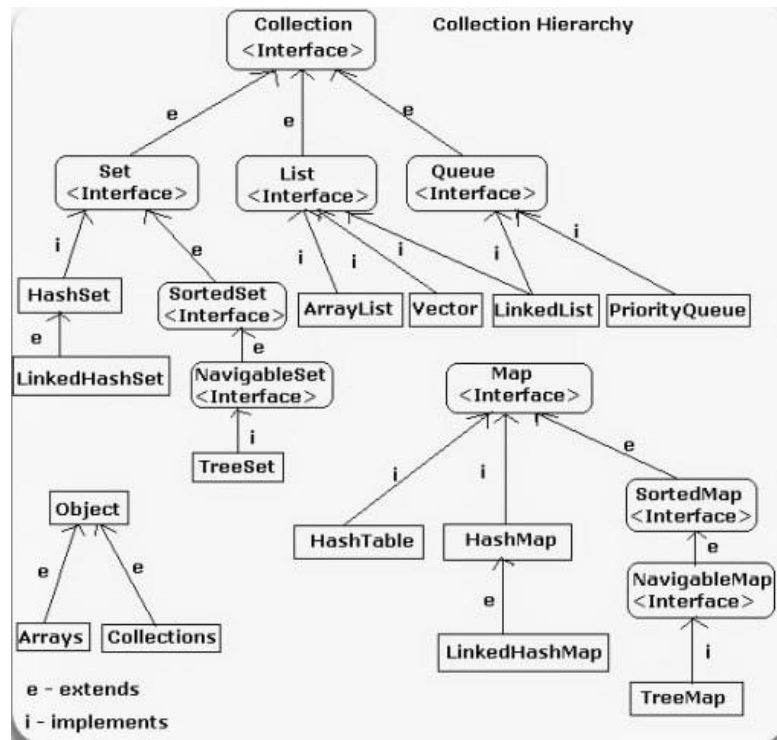
BAB I

Data Bentuk String dan Pelbagai Propertinya

A. *Collection* sebagai Media Penyimpanan Data

Collection adalah suatu objek yang bisa digunakan untuk menyimpan sekumpulan objek. Objek yang ada dalam *Collection* disebut elemen. *Collection* menyimpan elemen yang bertipe *Object*, sehingga berbagai tipe objek bisa disimpan dalam *collection*. *Class-class* mengenai *collection* tergabung dalam *Java Collection Framework*. *Class-class* *collection* diletakkan dalam *package java.util* dan mempunyai dua *interface* utama yaitu *collection* dan *map*. Selain untuk menyimpan, *collection* juga bisa digunakan untuk mencari, mengurutkan, memasukkan dan menghapus data.

Salah satu *collection* yang paling sederhana yaitu *Array*. *Collection* terbagi menjadi 3 kelompok yaitu *Set*, *List* dan *Map*. Berikut struktur hierarki *interface* dan *class* yang termasuk dalam kelompok *collection* ini.



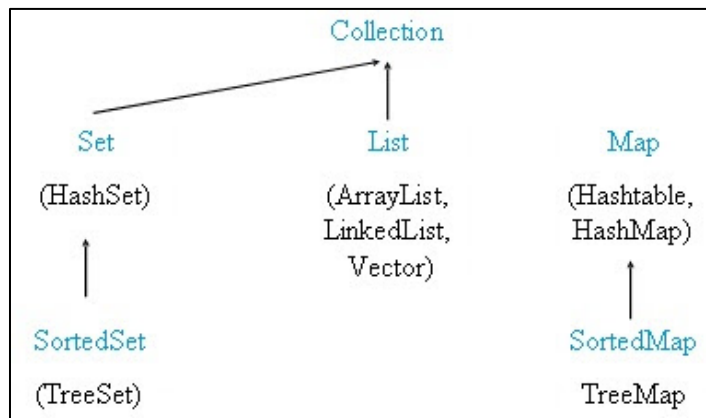
Beberapa kelebihan utama dalam menggunakan *collection framework* yaitu sebagai berikut.

- Mengurangi *effort* dalam membuat program, karena sudah tersedia struktur data dan algoritma tanpa harus menulis sendiri.
- Meningkatkan performa, karena setiap implementasi dapat berfungsi maksimal sesuai kasus yang ada.
- Mudah dipelajari, sehingga mengurangi *effort* untuk mempelajari cara menggunakan API.
- Dapat dikembangkan dan fleksibel terhadap tipe *object* yang ada dalam *Collection*.

1. Pengelompokan *Collection*.

Jenis pengelompokan *collection* merupakan pengelompokan satu dimensi. Berdasarkan Teknik *Collection Interface*, pengelompokannya terbagi menjadi tiga kelompok, yaitu *set*, *list* dan *queue*.

Terdapat lima implementasi dasar yang digunakan antara lain *Hash Table*, *Resizable Array*, *Balanced Tree*, *Linked List*, dan *Hash Table + Linked List*.



Pengelompokan *Collection*

Java collection framework terbagi menjadi tiga kelompok yaitu sebagai berikut.

a) *Set*

Set mengikuti model himpunan, dimana objek yang tersimpan dalam *set* haruslah unik. Urutan maupun letak dari objek tidaklah penting, hanya saja keberadaan objek saja yang penting. *Class-class* yang mengimplementasikan *interface set* adalah *Hash Set*. *Interface Sorted Set* merupakan *sub Interface* dari *interface Set*. Untuk mengurutkan *set*, dapat menggunakan *class* yang mengimplementasikan *interface Sorted Set* yaitu *Tree Set*.

b) *List*

List merupakan variabel yang mengandung beberapa item dan dideklarasikan menggunakan kurung siku. Elemen *list* tersimpan berdasarkan urutan masukan. Dalam metodenya *list* menerima duplikat. Cara penyimpanannya sama seperti *array*, oleh sebab itu memiliki posisi awal dan posisi akhir, menyisipkan objek pada posisi tertentu, mengakses dan menghapus isi *list*, dimana semua proses ini selalu didasarkan pada

urutannya. *Class-class* yang mengimplementasikan *interface List* adalah *Vector*, *Stack*, *Linked List* dan *Array List*.

Terdapat *interface Queue* yang cara penyimpanannya sama seperti *List*. *Interface* ini menyimpan objek menggunakan metode *FIFO (First In First Out)* yaitu objek yang masuk pertama keluar pertama. *Class-class* yang mengimplementasikan *interface Queue* adalah *Priority Queue* dan *Linked List*. Data yang tersimpan pada objek *Priority Queue* akan diurutkan, data tersebut harus mengimplementasikan objek *Comparable* atau *Comparator*. *List* juga menyediakan *method* untuk merandom isi dari *collection*.

c) Map

Map adalah objek yang memetakan *key* dengan *value*. Disebut juga asosiatif *array* atau *dictionary*. Perbedaan mendasar *map* dengan *collection* yang lain adalah untuk menyimpan objek pada *map*, diperlukan sepasang objek yaitu *key* yang bersifat unik dan nilai yang disimpan. Untuk mengakses nilai tersebut maka kita perlu mengetahui *key* dari nilai tersebut. *Map* juga dikenal sebagai *dictionary/kamus*. Pada saat menggunakan kamus, diperlukan suatu kata yang digunakan untuk pencarian. *Class-class* yang mengimplementasikan *Map* adalah *Hash table*, *Hash Map*, dan *Linked Hash Map*. Untuk mengurutkan *Map* digunakan *interface SortedMap*. *Class* yang mengimplementasikan *interface Map* adalah *Tree Map*.