

KUSUMA WEB ACADEMY

DASAR-DASAR JALUR JAVA

Modul 15: Struktur Data Dinamis - Java Collections Framework

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

Keterbatasan Array Tradisional

- ▶ Ukuran kapasitas data bersifat statis (tetap). Kita tidak bisa menambahkan data baru jika array sudah penuh tanpa melakukan instansiasi ulang.

Solusi: Collections Framework

- ▶ Sekumpulan kelas dan antarmuka terintegrasi yang menyediakan struktur data berukuran dinamis yang kaya fungsionalitas optimasi.

Tiga Pilar Utama JCF

- ▶ **List:** Koleksi elemen dinamis yang terurut dan memperbolehkan adanya duplikasi data.
- ▶ **Set:** Koleksi unik yang tidak memperbolehkan duplikasi data sama sekali.
- ▶ **Map:** Struktur penyimpanan berpasangan berupa Kunci & Nilai (*Key-Value Pair*).

Struktur Dinamis Terurut: Interface List

ArrayList vs LinkedList

- ▶ **ArrayList:** Menggunakan array dinamis di belakang layar. Sangat cepat dalam proses pembacaan acak (*get/search*).
- ▶ **LinkedList:** Memanfaatkan skema rangkaian node berantai. Sangat efisien untuk proses penambahan atau pembuangan data di barisan tengah.

Implementasi ArrayList

```
import java.util.ArrayList;
import java.util.List;

public class DemoList {
    public static void main(String[]
        args) {
        // Deklarasi list tipe String
        List<String> list = new
            ArrayList<>();
        list.add("Java");
        list.add("Python");
        list.add("Java"); // Duplikasi
            diizinkan

        System.out.println(list.get(1))
            ; // "Python"
        System.out.println("Jumlah: " +
            list.size());
    }
}
```

Struktur Tanpa Duplikasi: Interface Set

Konsep HashSet

- ▶ Set tidak mepedulikan urutan penyimpanan elemen, melainkan berfokus penuh pada ****keunikan absolut data****.
- ▶ Jika Anda mencoba memasukkan data duplikat yang sudah ada di dalamnya, data tersebut otomatis akan diabaikan tanpa memicu timbulnya error.

Implementasi HashSet

```
import java.util.HashSet;
import java.util.Set;

public class DemoSet {
    public static void main(String[]
        args) {
        Set<String> set = new HashSet
            <>();
        set.add("ID101");
        set.add("ID102");
        set.add("ID101"); // Otomatis
            ditolak

        System.out.println(set); //
            Output: [ID101, ID102]
        System.out.println(set.contains
            ("ID102")); // true
    }
}
```



Struktur Kamus Berpasangan: Interface Map

Konsep HashMap (Key & Value)

- ▶ Memetakan nilai (*Value*) berdasarkan kunci pengenal (*Key*) yang unik.
- ▶ **Aturan Utama:**
 - ▶ Kunci pencari (*Key*) wajib unik.
 - ▶ Isi data (*Value*) diperbolehkan duplikat.

Implementasi HashMap

```
import java.util.HashMap;
import java.util.Map;

public class DemoMap {
    public static void main(String[]
        args) {
        Map<String, Integer> map = new
            HashMap<>();
        map.put("Budi", 85);
        map.put("Andi", 90);
        map.put("Cici", 75);

        // Akses data berdasarkan Key
        int nilaiAndi = map.get("Andi")
            ;
        System.out.println("Nilai Andi:
            " + nilaiAndi); // 90
    }
}
```