



Search...

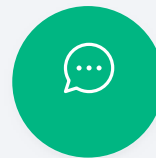
# Struktur Data dalam Java

[Home](#) > [Belajar Koding](#) > [Java](#) > [Struktur Data dalam Java](#)



## Table of Contents

1. Array
2. ArrayList
3. LinkedList
4. HashMap
5. HashSet
6. Contoh Program Sederhana dengan Struktur Data



Struktur data adalah cara untuk menyimpan dan mengorganisir data sehingga dapat digunakan secara efisien. Java menyediakan berbagai struktur data bawaan yang berguna untuk berbagai keperluan. Pada modul ini, Anda akan mempelajari tentang array, array list, linked list, hash map, dan set.

## Array

Array adalah struktur data yang menyimpan elemen-elemen dengan tipe data yang sama dalam urutan yang tetap. Array memiliki ukuran tetap setelah dibuat.

### Membuat Array:

```
// Deklarasi dan inisialisasi array
int[] numbers = {1, 2, 3, 4, 5};

// Cara lain untuk mendeklarasikan dan menginisialisasi array
int[] moreNumbers = new int[5]; // array dengan ukuran 5
moreNumbers[0] = 10;
moreNumbers[1] = 20;
moreNumbers[2] = 30;
moreNumbers[3] = 40;
moreNumbers[4] = 50;
```

### Akses Elemen Array:

```
System.out.println(numbers[0]); // Output: 1
System.out.println(moreNumbers[2]); // Output: 30
```

### Iterasi Array:

```
for (int i = 0; i < numbers.length; i++) {
    System.out.println(numbers[i]);
}

// Menggunakan for-each loop
for (int number : numbers) {
    System.out.println(number);
}
```

## ArrayList

ArrayList adalah struktur data yang lebih fleksibel daripada array, yang dapat berubah ukurannya secara dinamis. ArrayList adalah bagian dari pustaka koleksi Java.

### Membuat ArrayList:

```
import java.util.ArrayList;

// Deklarasi dan inisialisasi ArrayList
ArrayList<String> names = new ArrayList<>();
names.add("Alice");
names.add("Bob");
names.add("Charlie");
```

#### Akses Elemen ArrayList:

```
System.out.println(names.get(0)); // Output: Alice
System.out.println(names.get(1)); // Output: Bob
```

#### Menambahkan dan Menghapus Elemen:

```
names.add("David");
names.remove("Alice");
names.remove(0); // Menghapus elemen di indeks 0
```

#### Iterasi ArrayList:

```
for (String name : names) {
    System.out.println(name);
}
```

## LinkedList

LinkedList adalah struktur data yang menyimpan elemen-elemen dalam bentuk node, di mana setiap node berisi elemen data dan referensi ke node berikutnya. LinkedList juga merupakan bagian dari pustaka koleksi Java.

#### Membuat LinkedList:

```
import java.util.LinkedList;
```

```
// Deklarasi dan inisialisasi LinkedList
LinkedList<String> cities = new LinkedList<>();
cities.add("New York");
cities.add("Los Angeles");
cities.add("Chicago");
```

#### Akses Elemen LinkedList:

```
System.out.println(cities.get(0)); // Output: New York
System.out.println(cities.get(1)); // Output: Los Angeles
```

#### Menambahkan dan Menghapus Elemen:

```
cities.add("Houston");
cities.remove("New York");
cities.remove(0); // Menghapus elemen di indeks 0
```

#### Iterasi LinkedList:

```
for (String city : cities) {
    System.out.println(city);
}
```

## HashMap

HashMap adalah struktur data yang menyimpan pasangan kunci-nilai (key-value pairs) yang tidak terurut. HashMap berguna untuk mencari nilai berdasarkan kunci dengan cepat.

#### Membuat HashMap:

```
import java.util.HashMap;

// Deklarasi dan inisialisasi HashMap
```

```
HashMap<String, Integer> ages = new HashMap<>();
ages.put("Alice", 30);
ages.put("Bob", 25);
ages.put("Charlie", 35);
```

#### Akses dan Mengubah Nilai:

```
System.out.println(ages.get("Alice")); // Output: 30
ages.put("Alice", 31); // Mengubah nilai
System.out.println(ages.get("Alice")); // Output: 31
```

#### Menghapus Elemen:

```
ages.remove("Bob");
```

#### Iterasi HashMap:

```
for (String name : ages.keySet()) {
    System.out.println("Name: " + name + ", Age: " + ages.get(name));
}
```

## HashSet

HashSet adalah struktur data yang menyimpan elemen-elemen unik yang tidak terurut. HashSet berguna untuk menyimpan elemen-elemen tanpa duplikasi.

#### Membuat HashSet:

```
import java.util.HashSet;

// Deklarasi dan inisialisasi HashSet
HashSet<String> fruits = new HashSet<>();
fruits.add("Apple");
fruits.add("Banana");
```

```
fruits.add("Cherry");
```

#### Menambahkan dan Menghapus Elemen:

```
fruits.add("Durian");  
fruits.remove("Banana");
```

#### Iterasi HashSet:

```
for (String fruit : fruits) {  
    System.out.println(fruit);  
}
```

## Contoh Program Sederhana dengan Struktur Data

Berikut adalah contoh program sederhana yang menggabungkan penggunaan array, ArrayList, LinkedList, HashMap, dan HashSet:

```
import java.util.ArrayList;  
import java.util.LinkedList;  
import java.util.HashMap;  
import java.util.HashSet;  
  
public class Main {  
    public static void main(String[] args) {  
        // Array  
        int[] numbers = {1, 2, 3, 4, 5};  
        for (int number : numbers) {  
            System.out.println("Number: " + number);  
        }  
  
        // ArrayList  
        ArrayList<String> names = new ArrayList<>();  
        names.add("Alice");  
        names.add("Bob");
```

```

names.add("Charlie");
for (String name : names) {
    System.out.println("Name: " + name);
}

// LinkedList
LinkedList<String> cities = new LinkedList<>();
cities.add("New York");
cities.add("Los Angeles");
cities.add("Chicago");
for (String city : cities) {
    System.out.println("City: " + city);
}

// HashMap
HashMap<String, Integer> ages = new HashMap<>();
ages.put("Alice", 30);
ages.put("Bob", 25);
ages.put("Charlie", 35);
for (String name : ages.keySet()) {
    System.out.println("Name: " + name + ", Age: " + ages.get(name));
}

// HashSet
HashSet<String> fruits = new HashSet<>();
fruits.add("Apple");
fruits.add("Banana");
fruits.add("Cherry");
for (String fruit : fruits) {
    System.out.println("Fruit: " + fruit);
}
}
}

```

Dengan memahami struktur data dalam Java, Anda dapat menyimpan dan mengelola data secara efisien, memungkinkan Anda untuk menulis program yang lebih kompleks dan efisien. Jika ada pertanyaan lebih lanjut atau bagian yang perlu diperjelas, jangan ragu untuk bertanya.

What are your Feelings



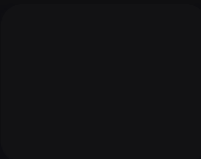
Share This Article :



Still stuck? How can we help?

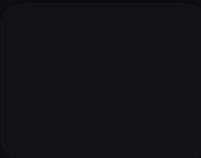
Updated on June 27, 2024

### MOST POPULAR



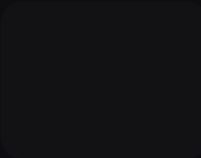
#### Cara Menggunakan Foreach di Node.js?

February 13, 2024    69 Views



#### Cara Install XAMPP (Apache, MariaDB, PHP dan Perl) pada Windows

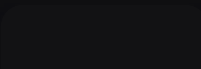
November 26, 2023    50 Views



#### Bagaimana Prospek Karir Jurusan Teknik Informatika?

February 14, 2024    47 Views

### OUR PICKS



#### Kenalan dengan Danfo.js: Apa Itu, Kapan Dipakai, dan Perbandingan Cepat vs



Pandas

November 8, 2025

Agen Kode Berbasis AI Mulai Menguasai Dunia Korporat: Ini Daftar Tools-nya

August 11, 2025

Vibe Coding: Cara Baru Ngoding Bareng AI yang Lagi Ngetren di 2025

August 5, 2025

## Subscribe to Updates

Get the latest creative news from FooBar about art, design and business.

Your email address..

SUBSCRIBE

☐

By signing up, you agree to the our terms and our Privacy Policy agreement.



[JOBS](#)

[PROJECT FREELANCE](#)

[ABOUT ME](#)

[CONTACT](#)

Copyright © 2025 Lampung Dev. Developed by Abdan Zam Zam Ramadhan.