



# Algoritma dan Pemrograman Komputer 1

## Bab 10: Array

---

Aslam Pandu Tasminto – 5002241025

M. Ma'ruf Qomaruddin Kafi – 5002241095

November 2024

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

# Daftar Isi

- Pengenalan Array
- Keuntungan Array
- Deklarasi Array
- Mengakses Elemen Array
- Panjang Array
- Inisialisasi Array
- Looping dengan Array
- Studi Kasus
- Tambahan Materi dari Modul
- Best Practices
- Kesimpulan
- Latihan
- Referensi

# Pengenalan Array

---

# Konsep Dasar Array

## Apa itu Array?

Array adalah struktur data yang menyimpan kumpulan elemen dengan **tipe data yang sama** dalam satu variabel.

- Seperti **deretan loker** dengan nomor-nomor berurutan
- Setiap elemen diakses melalui **indeks** (dimulai dari 0)
- Memudahkan pengelolaan data dalam jumlah banyak

## Analogi Array

Bayangkan array seperti deretan **loker sekolah**:

- Setiap loker memiliki **nomor** (indeks)
- Isi setiap loker harus **sejenis** (tipe data sama)
- Sekolah tersebut memiliki jumlah loker tertentu (ukuran array)

# Tanpa Array vs Dengan Array

## Tanpa Array

- Banyak variabel terpisah
- Sulit dikelola
- Tidak efisien

## Kode Tanpa Array

```
1 String pemilik_kotak_1 = "Udin";
2 String pemilik_kotak_2 = "Joko";
3 String pemilik_kotak_3 = "Budi";
4 String pemilik_kotak_4 = "Sari";
5 String pemilik_kotak_5 = "Rina";
6 String pemilik_kotak_6 = "Gibran";
7 // ... dan seterusnya
8 // untuk 100 kotak?
```

## Masalah

- Bagaimana jika ada 1000 kotak?
- Sulit melakukan operasi massal
- Kode menjadi sangat panjang

## Dengan Array

- Satu variabel untuk semua
- Mudah dikelola
- Sangat efisien

## Kode Dengan Array

```
1 String[] pemilik_kotak = {
2     "Udin", "Joko", "Budi",
3     "Sari", "Rina", "Gibran"
4 };
5 // Akses mudah dengan index
6 // pemilik_kotak[0] = "Udin"
7 // pemilik_kotak[1] = "Joko"
```

## Keuntungan

- Dapat menampung ribuan data
- Mudah di-loop dan diproses
- Kode lebih bersih dan terstruktur

## Keuntungan Array

---

# Manfaat Penggunaan Array

Keuntungan	Deskripsi
Efisiensi Memory	Menyimpan banyak data dalam satu variabel
Akses Random	Dapat mengakses elemen mana saja langsung via index
Kode Lebih Rapi	Tidak perlu banyak variabel untuk data sejenis
Mudah di-Loop	Dapat diproses secara massal dengan looping
Kesesuaian Algoritma	Cocok untuk sorting, searching, dll.
Kemudahan Maintenance	Lebih mudah dikelola daripada banyak variabel terpisah

**Tabel 1:** Keuntungan Menggunakan Array dalam Pemrograman

# Deklarasi Array

---



# Cara Mendeklarasikan Array

## 3 Cara Deklarasi Array dalam Java

1. Deklarasi kemudian inisialisasi
2. Deklarasi dan alokasi memory
3. Deklarasi dengan nilai langsung

### Contoh Deklarasi Array

```
1 // Cara 1: Deklarasi kemudian inisialisasi
2 int[] angka;
3 angka = new int[5];
4
5 // Cara 2: Deklarasi dan alokasi memory
6 int[] angka = new int[5];
7
8 // Cara 3: Deklarasi dengan nilai langsung
9 int[] angka = {1, 2, 3, 4, 5};
10 String[] nama = {"Udin", "Joko", "Budi"};
```

# Mengakses Elemen Array

---

# Cara Mengakses dan Mengubah Elemen Array

## Menggunakan Index

- Index dimulai dari **0** sampai **panjang array - 1**
- Syntax: `namaArray[index]`
- Dapat membaca dan mengubah nilai elemen

## Contoh Akses Elemen

```
1 public class AksesArray {  
2     public static void main(String[] args) {  
3         int[] nilai = {85, 90, 78, 92, 88};  
4  
5         // Mengakses elemen  
6         System.out.println("Nilai pertama: " + nilai[0]); // 85  
7         System.out.println("Nilai ketiga: " + nilai[2]); // 78  
8  
9         // Mengubah elemen  
10        nilai[1] = 95; // Mengubah nilai kedua  
11        System.out.println("Nilai kedua sekarang: " + nilai[1]); // 95  
12    }  
13 }
```

## Panjang Array

---

# Menggunakan Property .length

## Property .length

- Setiap array memiliki property .length
- Menghasilkan **ukuran array** (banyaknya elemen)
- Berguna untuk looping dan batas akses array

## Contoh Penggunaan .length

```
1 public class PanjangArray {
2     public static void main(String[] args) {
3         int[] bilangan = {10, 20, 30, 40, 50};
4         String[] buah = {"Apel", "Jeruk", "Mangga"};
5
6         System.out.println("Panjang array bilangan: " + bilangan.length); // 5
7         System.out.println("Panjang array buah: " + buah.length); // 3
8
9         // Looping dengan .length
10        for(int i = 0; i < bilangan.length; i++) {
11            System.out.println("Elemen " + i + ": " + bilangan[i]);
12        }
13    }
14 }
```

# Inisialisasi Array

---

# Inisialisasi Langsung

## Inisialisasi Langsung

```
1 class InisialisasiLangsung {  
2     public static void main(String[] args) {  
3         // Inisialisasi dengan nilai langsung  
4         int[] angka = {15, 25, 30, 45};  
5         String[] nama = {"Jhonny", "Nando", "Queren"};  
6  
7         System.out.println("Array angka: ");  
8         for(int i : angka) {  
9             System.out.print(i + " ");  
10        }  
11  
12        System.out.println("\nArray nama: ");  
13        for(String n : nama) {  
14            System.out.print(n + " ");  
15        }  
16    }  
17 }
```

## Output Program

Array angka: 15 25 30 45

Array nama: Jhonny Nando Queren

# Inisialisasi Per Elemen

## Inisialisasi Elemen Secara Individual

```
1 class InisialisasiPerElemen {
2     public static void main(String[] args) {
3         // Deklarasi array
4         String[] mahasiswa = new String[3];
5
6         // Inisialisasi per elemen
7         mahasiswa[0] = "Jhonny";
8         mahasiswa[1] = "Nando";
9         mahasiswa[2] = "Queren";
10
11        // Menampilkan semua elemen
12        System.out.println("Daftar Mahasiswa:");
13        for(int i = 0; i < mahasiswa.length; i++) {
14            System.out.println((i+1) + ". " + mahasiswa[i]);
15        }
16    }
17 }
```

## Output Program

Daftar Mahasiswa:

1. Jhonny
2. Nando
3. Queren



## Looping dengan Array

---

# For Loop dengan Array

## Traditional For Loop

```
1 public class ForTraditional {
2     public static void main(String[] args) {
3         int[] nilai = {85, 90, 78, 92, 88};
4
5         System.out.println("Daftar Nilai:");
6         for(int i = 0; i < nilai.length; i++) {
7             System.out.println("Nilai " + (i+1) + ": " + nilai[i]);
8         }
9
10        // Menghitung rata-rata
11        int total = 0;
12        for(int i = 0; i < nilai.length; i++) {
13            total += nilai[i];
14        }
15        double rataRata = (double) total / nilai.length;
16        System.out.println("Rata-rata: " + rataRata);
17    }
18 }
```

# Enhanced For Loop (For-Each)

## For-Each Loop untuk Array

```
1 public class ForEachArray {
2     public static void main(String[] args) {
3         int[] bilangan = {1, 2, 3, 4, 5};
4         String[] hari = {"Senin", "Selasa", "Rabu", "Kamis", "Jumat"};
5
6         // Enhanced for loop (for-each)
7         System.out.print("Bilangan: ");
8         for(int angka : bilangan) {
9             System.out.print(angka + " ");
10        }
11
12        System.out.print("\nHari: ");
13        for(String h : hari) {
14            System.out.print(h + " ");
15        }
16
17        // Menghitung jumlah elemen > 3
18        int count = 0;
19        for(int num : bilangan) {
20            if(num > 3) count++;
21        }
22        System.out.println("\nBilangan > 3: " + count + " buah");
23    }
24 }
```

## Studi Kasus

---

# Studi Kasus 1: Statistik Nilai

## Analisis Data Nilai dengan Array

```
1 public class StatistikNilai {
2     public static void main(String[] args) {
3         int[] nilai = {85, 90, 78, 92, 88, 76, 95, 82, 79, 91};
4
5         // Menghitung total
6         int total = 0;
7         for(int n : nilai) {
8             total += n;
9         }
10        double rataRata = (double) total / nilai.length;
11
12        // Mencari nilai tertinggi dan terendah
13        int max = nilai[0];
14        int min = nilai[0];
15        for(int i = 1; i < nilai.length; i++) {
16            if(nilai[i] > max) max = nilai[i];
17            if(nilai[i] < min) min = nilai[i];
18        }
19
20        System.out.println("Statistik Nilai:");
21        System.out.println("Rata-rata: " + rataRata);
22        System.out.println("Nilai tertinggi: " + max);
23        System.out.println("Nilai terendah: " + min);
24        System.out.println("Jumlah siswa: " + nilai.length);
25    }
26 }
```

# Studi Kasus 2: Pencarian dalam Array

## Mencari Elemen dalam Array

```
1 public class PencarianArray {
2     public static void main(String[] args) {
3         int[] array = {10, 20, 25, 30, 45, 50};
4         int target = 30;
5         boolean ditemukan = false;
6         int index = -1;
7
8         // Linear search
9         for(int i = 0; i < array.length; i++) {
10             if(array[i] == target) {
11                 ditemukan = true;
12                 index = i;
13                 break;
14             }
15         }
16
17         if(ditemukan) {
18             System.out.println("Nilai " + target + " ditemukan pada index: " + index);
19         } else {
20             System.out.println("Nilai " + target + " tidak ditemukan dalam array");
21         }
22
23         // Menampilkan semua elemen array
24         System.out.print("Array: ");
25         for(int elemen : array) {
26             System.out.print(elemen + " ");
27         }
28     }
```

# **Tambahan Materi dari Modul**

---

# Copy Array dengan System.arraycopy()

## Konsep System.arraycopy()

- Method built-in Java untuk menyalin array
- Syntax: `System.arraycopy(src, srcPos, dest, destPos, length)`
- Lebih cepat daripada loop manual

## Contoh Penggunaan

```
1 public class CopyArray {  
2     public static void main(String[] args) {  
3         int[] asal = {7, 4, 1, 3, 6, 4, 2};  
4         int[] tujuan = new int[3];  
5  
6         // Salin 3 elemen dari index 0  
7         System.arraycopy(asal, 0, tujuan, 0, 3);  
8  
9         System.out.print("Asal: ");  
10        for(int i : asal) System.out.println(i + " ");  
11  
12        System.out.print("\nTujuan: ");  
13        for(int i : tujuan) System.out.println(i + " ");  
14    }  
15 }
```

**Output:** Asal: 7 4 1 3 6 4 2 Tujuan: 7 4 1



# Command Line Arguments (args[])

## Konsep args[]

- Parameter `String[] args` pada main method
- Menerima input dari command line saat menjalankan program
- Setiap kata dipisah spasi menjadi elemen array

## Contoh Penggunaan

```
1 public class CommandLineArgs {  
2     public static void main(String[] args) {  
3         // Run: java CommandLineArgs Senin November 2024  
4         if(args.length >= 3) {  
5             System.out.println("Hari: " + args[0]);  
6             System.out.println("Bulan: " + args[1]);  
7             System.out.println("Tahun: " + args[2]);  
8         } else {  
9             System.out.println("Input: <hari> <bulan> <tahun>");  
10        }  
11    }  
12 }
```

## Contoh Eksekusi di Terminal

```
1 # Compile program  
2 javac CommandLineArgs.java  
3  
4 # Run dengan arguments  
5 java CommandLineArgs Senin November 2024
```

## Best Practices

---

# Best Practices Penggunaan Array

## Tips dan Rekomendasi

- **Inisialisasi dengan nilai default** - Gunakan loop untuk mengisi array dengan nilai awal
- **Validasi index** - Selalu pastikan index dalam range [0, length-1]
- **Gunakan enhanced for** - Untuk iterasi tanpa perlu mengubah elemen
- **Array.length** - Selalu gunakan property length untuk batas looping
- **Hindari angka langsung** - Jangan tulis angka ukuran array langsung, gunakan variabel
- **Documentasi** - Berikan komentar untuk array yang kompleks

## Contoh Baik

```
1 // BAIK: menggunakan variabel untuk ukuran array
2 int jumlahSiswa = 10;
3 int[] nilai = new int[jumlahSiswa];
4
5 // KURANG BAIK: angka langsung
6 int[] nilai = new int[10];
```

## Kesimpulan

---

## Inti Bab 10: Array

- **Array** adalah struktur data untuk menyimpan kumpulan elemen bertipe sama
- **Deklarasi array** dapat dilakukan dengan 3 cara berbeda
- **Elemen array** diakses melalui index (dimulai dari 0)
- **Property .length** memberikan ukuran array
- **Looping dengan array** menggunakan for traditional atau enhanced for
- **System.arraycopy()** untuk menyalin elemen antar array
- **String[] args** untuk command line arguments
- Array membuat pengelolaan data dalam jumlah banyak menjadi lebih efisien

# Latihan

---

# Latihan 1: Manipulasi Array Dasar

## Soal 1: Operasi Array Sederhana

Buat program yang:

1. Mendeklarasi array integer dengan 5 elemen
2. Mengisi array dengan nilai 2, 4, 6, 8, 10
3. Menampilkan semua elemen array
4. Menghitung dan menampilkan jumlah semua elemen
5. Menampilkan elemen awal dan akhir

### Soal 2: Array Processing

Buat program yang:

1. Memiliki array string: "apel", "jeruk", "mangga", "anggur", "pisang"
2. Mencari index dari buah "mangga"
3. Mengganti "anggur" dengan "durian"
4. Menampilkan array sebelum dan sesudah modifikasi pada point 3.



## Latihan 3: Statistik Array

### Soal 3: Analisis Data Array

Buat program yang:

1. Menerima sebanyak  $n$  input array integer dari user
2. Menghitung:
  - Rata-rata nilai
  - Banyaknya elemen di atas rata-rata
  - Banyaknya elemen genap dan ganjil
3. Menampilkan hasil analisis dalam format yang rapi

## Referensi

---

## Referensi:

- **Modul Praktikum Algoritma dan Pemrograman - Modul 10**  
Departemen Matematika FSAD ITS
- **Oracle Java Tutorials - Arrays**  
`https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html`
- **GeeksforGeeks - Arrays in Java**  
`https://www.geeksforgeeks.org/arrays-in-java/`

## Referensi:

- **W3Schools - Java Arrays**  
[https://www.w3schools.com/java/java\\_arrays.asp](https://www.w3schools.com/java/java_arrays.asp)
- **Programiz - Java Arrays**  
<https://www.programiz.com/java-programming/arrays>

**Terima Kasih**