

KUSUMA WEB ACADEMY

# DASAR-DASAR JALUR JAVA

Modul 06: Kontrol Aliran - Perulangan (Looping)

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

# Perulangan Terhitung: Struktur for & foreach

## Perulangan for Standar

- ▶ Ideal digunakan saat kita sudah tahu secara pasti jumlah iterasi perulangan yang ingin dilakukan.
- ▶ **Sintaksis:** `for (inisialisasi; kondisi; update) { }`

```
// Mencetak angka 1 s/d 5
for (int i = 1; i <= 5; i++) {
    System.out.println("Angka: " + i);
}
```

## Enhanced For-Loop (foreach)

- ▶ Diperkenalkan untuk menyederhanakan pembacaan elemen dari struktur koleksi atau array secara sequential tanpa bermain index.

```
int[] nilaiUjian = {80, 90, 75, 85};

// Mengakses data array secara cepat
for (int nilai : nilaiUjian) {
    System.out.println("Nilai: " + nilai);
    ;
}
```

# Perulangan Kondisional: while & do-while

## Perulangan while

- ▶ Melakukan evaluasi kondisi di **awal** sebelum mengeksekusi tubuh program. Jika bernilai **false**, blok tidak tereksekusi sama sekali (0 kali).

```
int counter = 1;
while (counter <= 5) {
    System.out.println("Iterasi: " +
        counter);
    counter++; // Wajib ada update state!
}
```

## Perulangan do-while

- ▶ Mengeksekusi blok kode terlebih dahulu **minimal satu kali**, lalu mengecek kondisi di bagian akhir.

```
int angka = 10;
do {
    System.out.println("Berjalan sekali!");
    angka++;
} while (angka < 5); // false, tapi sudah print
```

## Bahaya Infinite Loop!

Selalu pastikan ada kondisi pengubah (\*update expression\*) agar loop bisa berhenti dan tidak membekukan sistem komputer Anda.

# Pernyataan Lompatan: Keyword break & continue

## Pernyataan break

- ▶ Menghentikan jalannya perulangan saat itu juga secara paksa dan langsung memindahkan kendali ke baris kode di luar perulangan.

```
for (int i = 1; i <= 10; i++) {  
    if (i == 5) {  
        break; // Langsung keluar loop  
    }  
    System.out.print(i + " "); // Output:  
    1 2 3 4  
}
```

## Pernyataan continue

- ▶ Melewati sisa baris kode di iterasi saat ini, lalu segera lompat ke evaluasi kondisi/update iterasi berikutnya.

```
for (int i = 1; i <= 5; i++) {  
    if (i == 3) {  
        continue; // Lewati angka 3  
    }  
    System.out.print(i + " "); // Output:  
    1 2 4 5  
}
```

# Perulangan Bersarang (Nested Loops)

## Konsep Dasar Nested Loop

- ▶ Kondisi di mana suatu struktur perulangan berada di dalam perulangan lainnya.
- ▶ Setiap satu putaran pada perulangan luar (\*outer loop\*), perulangan dalam (\*inner loop\*) akan diselesaikan penuh hingga selesai.

## Penggunaan Klasik

Sangat berguna untuk pengolahan data array multi-dimensi, pembuatan grafik matriks, serta algoritma pencarian.

## Contoh Pembuatan Pola Matriks

```
// Baris luar (Outer) dan Kolom dalam (Inner)
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 3; j++) {
        System.out.print("[ " + i + " , "
            + j + " ] ");
    }
    System.out.println(); // Baris baru
}
```

## Hasil Output:

```
[1,1] [1,2] [1,3]
[2,1] [2,2] [2,3]
[3,1] [3,2] [3,3]
```