

KUSUMA WEB ACADEMY

DASAR-DASAR JALUR JAVA

Modul 04: Operator Aritmatika, Logika, Relasional, Bitwise & Ternary

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

Operator Aritmatika & Assignment (Penugasan)

Operator Aritmatika Dasar

- ▶ Digunakan untuk melakukan perhitungan matematika standar pada tipe data numerik.
- ▶ + (Penjumlahan), - (Pengurangan).
- ▶ * (Perkalian), / (Pembagian).
- ▶ % (Modulo / Sisa Hasil Bagi).

Operator Assignment (Shorthand)

- ▶ Menyederhanakan operasi matematika dan langsung memasukkan hasil ke variabel awal.
- ▶ Contoh: `a += 5` ekuivalen dengan `a = a + 5`.

Increment & Decrement

- ▶ **Pre-increment (++x):** Tambah nilai dulu, baru evaluasi expression.
- ▶ **Post-increment (x++):** Evaluasi expression dulu, baru tambah nilai.

```
int a = 5;
int b = ++a; // a=6, b=6 (Pre)

int x = 5;
int y = x++; // x=6, y=5 (Post)

int sisa = 10 % 3; // sisa = 1
```

Operator Relasional & Logika Boolean

Operator Relasional (Perbandingan)

- ▶ Membandingkan dua nilai dan menghasilkan output berupa tipe data boolean (true atau false).
- ▶ == (Sama dengan), != (Tidak sama dengan).
- ▶ > (Lebih dari), < (Kurang dari).
- ▶ >= (Lebih dari sama dengan), <=.

Peringatan Penting!

Jangan gunakan == untuk membandingkan isi objek seperti String. Gunakan method `.equals()` untuk tipe referensi.

Operator Logika (Logical Operators)

- ▶ **AND (&&):** Menghasilkan true jika kedua operand bernilai true.
- ▶ **OR (| |):** Menghasilkan true jika salah satu atau kedua operand bernilai true.
- ▶ **NOT (!):** Membalikkan nilai logika (!true menjadi false).

```
int nilai = 85;
int absensi = 90;

// Evaluasi kelulusan
boolean lulus = (nilai >= 80) && (absensi
    >= 85);
System.out.println("Lulus: " + lulus); //
    true
```



Operator Bitwise & Operator Ternary

Operator Bitwise (Manipulasi Bit)

- ▶ Bekerja langsung pada representasi level biner (bit demi bit) dari tipe data integer.
- ▶ & (Bitwise AND), | (Bitwise OR).
- ▶ ^ (Bitwise XOR), ~ (Bitwise NOT/Complement).
- ▶ << (Left Shift): Menggeser bit ke kiri (mengalikan dengan 2).
- ▶ >> (Right Shift): Menggeser bit ke kanan (membagi dengan 2).

Operator Ternary (? :)

- ▶ Shorthand cerdas dari pernyataan if-else. Mengambil keputusan berbasis kondisi dalam satu baris instruksi singkat.
- ▶ **Sintaks:** kondisi ? nilaiJikaTrue : nilaiJikaFalse;

```
// Contoh Ternary
int skor = 75;
String status = (skor >= 60) ? "Lolos" :
                "Gagal";
System.out.println(status); // "Lolos"
```

```
// Contoh Bitwise
int x = 5; // biner: 0101
int y = 3; // biner: 0011
int z = x & y; // biner: 0001 (Desimal:
1)
```

Urutan Prioritas Operator (Operator Precedence)

Saat ekspresi matematika mengandung banyak operator berbeda, Java mengevaluasinya berdasarkan hierarki kekuatan berikut:

Prioritas	Tipe Operator	Simbol Operator
1 (Tertinggi)	Postfix / Prefix	expr++, expr--, ++expr, --expr
2	Perkalian & Pembagian	*, /, %
3	Penjumlahan & Pengurangan	+, -
4	Pergeseran Bit (Shift)	<<, >>, >>>
5	Relasional / Perbandingan	<, >, <=, >=, instanceof
6	Kesamaan (Equality)	==, !=
7	Logika AND & OR	&&,
8 (Terendah)	Ternary & Assignment	? :, =, +=, -=, *=, /=

Rekomendasi Praktis

Gunakan tanda kurung () secara eksplisit untuk memperjelas urutan operasi dan menghindari kebingungan logika kompilasi.