
MODUL 03: MENGUASAI OPERATOR PYTHON

MASTERCLASS PYTHON DASAR

Manipulasi, Komparasi, dan Evaluasi Logika Program secara Presisi.
Panduan Lengkap Operator Aritmatika, Perbandingan Relasional, Logika Boolean,
serta Urutan Operasi Eksekusi.

Kusuma Web Edu Series
Kelas Dasar Pemrograman — Modul 3 dari 10
Tanggal Rilis: 29 Juni 2026

Operator Aritmatika & Eksponen

Operator aritmatika digunakan untuk melakukan operasi matematika standar di dalam aplikasi. Selain penambahan dan pengurangan biasa, Python memiliki operator khusus untuk menangani pembagian tingkat lanjut secara presisi.

1. Daftar Lengkap Operator Aritmatika

- **Penjumlahan (+):** Menambahkan nilai operan kiri dan kanan.
- **Pengurangan (-):** Mengurangi operan kiri dengan operan kanan.
- **Perkalian (*):** Mengalikan nilai operan kiri dengan operan kanan.
- **Pembagian Riil (/):** Membagi nilai dan menghasilkan bentuk desimal float.
- **Pembagian Bulat (//) / *Floor Division*:** Membagi nilai dan membulatkan hasilnya ke bawah menuju bilangan bulat terdekat.
- **Sisa Pembagian (%) / *Modulo*:** Mengembalikan sisa dari hasil operasi pembagian.
- **Perpangkatan (**) / *Eksponen*:** Memangkatkan angka operan kiri dengan angka operan kanan.

2. Implementasi Kode Operasi Aritmatika

```
1 # Eksperimen Operasi Numerik
2 a = 17
3 b = 5
4
5 print("Pembagian Riil (a / b):", a / b) # Output: 3.4
6 print("Pembagian Bulat (a // b):", a // b) # Output: 3
7 print("Sisa Bagi / Modulo (a % b):", a % b) # Output: 2
8 print("Eksponen (b pangkat 3):", b ** 3) # Output: 125
```

Operator Perbandingan Relasional

Operator perbandingan digunakan untuk mengevaluasi relasi matematis antara dua objek data. Hasil evaluasi perbandingan ini ****pasti**** menghasilkan tipe data Boolean (`True` atau `False`).

1. Daftar Operator Perbandingan

- **Sama Dengan (==):** Menghasilkan `True` jika kedua nilai operan bernilai sama.
- **Tidak Sama Dengan (!=):** Menghasilkan `True` jika kedua nilai operan berbeda.
- **Lebih Besar (>):** Menghasilkan `True` jika operan kiri lebih besar dari operan kanan.
- **Lebih Kecil (<):** Menghasilkan `True` jika operan kiri lebih kecil dari operan kanan.
- **Lebih Besar Sama Dengan (>=):** Menghasilkan `True` jika operan kiri lebih besar atau sama dengan operan kanan.
- **Lebih Kecil Sama Dengan (<=):** Menghasilkan `True` jika operan kiri lebih kecil atau sama dengan operan kanan.

2. Contoh Evaluasi Relasional

```
1 x = 10
2 y = 12
3
4 print("Apakah x == y?", x == y) # Output: False
5 print("Apakah x != y?", x != y) # Output: True
6 print("Apakah x <= 10?", x <= 10) # Output: True
```

Asosiasi Penting

Operator `==` digunakan untuk komparasi nilai logika, sedangkan operator `=` digunakan untuk proses ***assignment*** (penugasan pengisian variabel). Jangan sampai tertukar!

Menguasai Operator Logika Boolean

Operator logika digunakan untuk menghubungkan dan mengevaluasi beberapa kondisi perbandingan sekaligus. Hal ini menjadi dasar pembentukan algoritma pengambilan keputusan di dalam program.

1. Tiga Operator Logika Utama

- **and (Konjungsi):** Menghasilkan nilai `True` ****hanya jika**** semua kondisi pengujian bernilai `True`.
- **or (Disjungsi):** Menghasilkan nilai `True` jika ****salah satu saja**** dari kondisi pengujian bernilai `True`.
- **not (Negasi):** Membalikkan status boolean (mengubah `True` menjadi `False`, dan sebaliknya).

2. Tabel Kebenaran Logika

```
1 kondisi_a = True
2 kondisi_b = False
3
4 # Evaluasi Logika AND
5 print(kondisi_a and kondisi_b) # Output: False
6
7 # Evaluasi Logika OR
8 print(kondisi_a or kondisi_b) # Output: True
9
10 # Evaluasi Negasi NOT
11 print(not kondisi_a) # Output: False
```

Aturan Urutan Operator (Operator Precedence)

Saat mengeksekusi operasi gabungan kompleks, Python memiliki aturan prioritas pengerjaan: 1. Tanda Kurung () → 2. Eksponen ** → 3. Perkalian & Pembagian → 4. Penambahan & Pengurangan → 5. Komparasi Relasional → 6. Operator Logika (`not`, kemudian `and`, lalu `or`).

Studi Kasus: Sistem Validasi Kelayakan Beasiswa

Untuk menyatukan konsep-konsep operator yang telah dipelajari, mari kita buat rancangan simulasi otomatis program penilaian beasiswa berbasis komputer.

1. Syarat Kelayakan Beasiswa

Peserta akan dinyatakan lolos seleksi beasiswa jika memenuhi dua kriteria utama berikut secara mutlak:

- Nilai Indeks Prestasi Kumulatif (IPK) harus lebih besar dari **3.5**.
- Nilai ujian keahlian teknis (Technical Score) tidak boleh kurang dari **80**.

2. Kode Lengkap Program

```
1 # Skenario Validasi Beasiswa Kusuma Web
2 print("=== PORTAL SELEKSI BEASISWA KUSUMA WEB ===")
3
4 # 1. Mengambil data prestasi kandidat
5 nama_peserta = input("Nama Kandidat: ")
6 ipk = float(input("Masukkan Nilai IPK Anda (skala 4.0): "))
7 tech_score = int(input("Masukkan Nilai Ujian Teknis (skala 100): "))
8
9 # 2. Proses Evaluasi dengan Operator Perbandingan dan Logika
10 apakah_ipk_lolos = (ipk > 3.5)
11 apakah_tech_lolos = (tech_score >= 80)
12
13 # Menilai kelayakan mutlak menggunakan operator logika 'and'
14 apakah_diterima = apakah_ipk_lolos and apakah_tech_lolos
15
16 # 3. Menampilkan status keputusan kelayakan
17 print("-----")
18 print(f>Nama Siswa: {nama_peserta}")
19 print(f>Apakah IPK memenuhi syarat?: {apakah_ipk_lolos}")
20 print(f>Apakah Skor Teknis memenuhi syarat?: {apakah_tech_lolos}")
21 print("=====")
22 print(f>STATUS KELAYAKAN BEASISWA: {apakah_diterima}")
23 print("=====")
```