

# MODUL 07: MODULARISASI DENGAN FUNGSI

## MASTERCLASS PYTHON DASAR

---

Meningkatkan Efisiensi Struktur Kode dengan Pendekatan Modular. Definisi Fungsi `def`, Nilai Balik (`return`), Konsep Cakupan Variabel (\*Scope\*), Argumen Default, serta Pengenalan Fungsi Singkat Lambda.

**Kusuma Web Edu Series**  
Kelas Dasar Pemrograman — Modul 7 dari 10  
Tanggal Rilis: 29 Juni 2026

## Konsep DRY & Deklarasi Fungsi Dasar

Dalam membangun aplikasi besar, menuliskan kode logika yang sama berulang kali di banyak tempat merupakan praktik buruk yang merusak efisiensi kerja developer. Konsep ini melanggar aturan utama arsitektur software: **\*\*DRY (\*Don't Repeat Yourself\*\*)**.

### 1. Mengetahui Fungsi Sebagai Blok Mandiri

Fungsi (*\*function\**) adalah blok kode terorganisir, modular, dan dapat digunakan kembali (*\*reusable\**) untuk menjalankan suatu perintah tindakan yang spesifik.

### 2. Sintaks Pendeklarasian Fungsi dengan `def`

Fungsi dideklarasikan menggunakan kata kunci `def`, diikuti nama fungsi, parameter di dalam kurung, dan diakhiri tanda titik dua. Gunakan kata kunci `return` untuk mengembalikan nilai hasil pemrosesan keluar dari fungsi.

### 3. Contoh Implementasi Fungsi

```
1 # Mendefinisikan fungsi perhitungan luas segitiga
2 def hitung_luas_segitiga(alas, tinggi):
3     luas = 0.5 * alas * tinggi
4     return luas # Mengembalikan hasil perhitungan ke pemanggil
5
6 # Memanggil fungsi secara berulang dengan parameter berbeda
7 luas_ke_1 = hitung_luas_segitiga(10, 5)
8 luas_ke_2 = hitung_luas_segitiga(14, 8)
9
10 print(f"Hasil Luas 1: {luas_ke_1} cm2 | Luas 2: {luas_ke_2} cm2")
```

## Cakupan Variabel (\*Scope\*) & Parameter Default

Memahami area aksesibilitas suatu variabel sangat krusial untuk menghindari konflik penamaan variabel di dalam memori saat program dijalankan.

### 1. Perbedaan Variabel Lokal vs Global

- **Lokal (Local Scope):** Variabel yang diciptakan di dalam suatu fungsi. Variabel ini bersifat privat, artinya hanya bisa diakses di dalam area internal fungsi itu sendiri dan akan dihancurkan dari memori setelah fungsi selesai bekerja.
- **Global (Global Scope):** Variabel yang dideklarasikan di luar fungsi manapun. Variabel ini bersifat publik, artinya bebas dibaca dari sudut kode manapun secara bebas.

### 2. Fungsi dengan Argumen Default

Kita dapat mengisi nilai awal cadangan pada parameter fungsi. Jika saat fungsi dipanggil pengguna tidak menyertakan argumen pengisi parameter, maka nilai default ini yang akan otomatis bekerja mengisi kekosongan data.

### 3. Contoh Penerapan Kode

```
1 # Variabel global
2 kategori = "PROG"
3
4 def sapa_peserta(nama, platform="Kusuma Web"):
5     # nama & platform adalah variabel lokal
6     print(f"[{kategori}] Halo {nama}, selamat belajar di {platform}!")
7
8 sapa_peserta("Budi") # Memanfaatkan nilai default platform
```

## Memahami Fungsi Anonim: Lambda

Python mendukung pembuatan fungsi ringkas satu baris tanpa perlu menggunakan struktur formal `def`. Fungsi jenis ini dinamakan fungsi anonim atau dikenal dengan sebutan **\*\*Lambda Function\*\***.

### 1. Mengapa Menggunakan Lambda?

Lambda sangat berguna ketika Anda membutuhkan fungsi sederhana sekali pakai untuk mendukung fungsionalitas fungsi tingkat tinggi lainnya (seperti saat menggunakan metode `filter()` atau `map()`).

### 2. Aturan Menuliskan Sintaks Lambda

Sintaksis lambda dibatasi hanya boleh berisi satu buah baris ekspresi kalkulasi saja, dengan format standar:

```
lambda parameter : ekspresi_kalkulasi
```

### 3. Contoh Perbandingan Implementasi

```
1 # 1. Penulisan cara konvensional
2 def hitung_kuadrat(x):
3     return x ** 2
4
5 # 2. Penulisan ringkas menggunakan Lambda
6 kuadrat_lambda = lambda x : x ** 2
7
8 print("Hasil Kuadrat Lambda dari 9 adalah:", kuadrat_lambda(9))
9 # Output: 81
```

## Studi Kasus: Sistem Penghitung Diskon Transaksi

Mari kita bangun aplikasi mini kasir pembayaran lisensi Kusuma Web yang memanfaatkan fungsi terstruktur untuk memisahkan logika perhitungan matematika utama dari alur proses interaksi tampilan antarmuka.

### 1. Aturan Logika Bisnis

- Buat fungsi `hitung_diskon(harga_total)` untuk menghitung nominal potongan harga 10% jika harga belanja di atas Rp 200.000.
- Gunakan fungsi modular tersebut untuk menghasilkan nota bersih pembayaran akhir.

### 2. Kode Lengkap Program

```
1 # Fungsi Modular Penghitung Diskon Transaksi
2 def hitung_diskon(total_belanja):
3     diskon = 0
4     if total_belanja >= 200000:
5         diskon = 0.10 * total_belanja # Potongan diskon 10%
6     return diskon
7
8 print("=== PROGRAM SISTEM TRANSAKSI KUSUMA WEB ===")
9 nama_produk = "Paket Edu Series Masterclass Python"
10 harga_satuan = 250000
11
12 # Kalkulasi transaksi menggunakan fungsi modular
13 potongan_diskon = hitung_diskon(harga_satuan)
14 tagihan_akhir = harga_satuan - potongan_diskon
15
16 # Menampilkan laporan nota pembayaran
17 print("-----")
18 print(f"Pembelian: {nama_produk}")
19 print(f"Harga Normal: Rp {harga_satuan}")
20 print(f"Potongan Diskon: Rp {potongan_diskon}")
21 print("=====")
22 print(f"TOTAL YANG WAJIB DIBAYAR: Rp {tagihan_akhir}")
23 print("=====")
```