

# MODUL 06: STRUKTUR DATA KOLEKSI PYTHON

## MASTERCLASS PYTHON DASAR

---

Mengelola Kumpulan Informasi Kompleks dalam Wadah Terstruktur.  
Eksplorasi Mendalam List (Mutable), Tuple (Immutable), Set (Unik), serta Dictionary (Key-Value), Lengkap dengan Metode Manipulasinya.

**Kusuma Web Edu Series**

Kelas Dasar Pemrograman — Modul 6 dari 10

Tanggal Rilis: 29 Juni 2026

## Eksplorasi Mendalam Struktur Data List

List adalah struktur data paling serbaguna di Python. Digunakan untuk menyimpan kumpulan elemen dalam satu tempat tunggal yang terurut secara berurutan berdasarkan nomor indeks.

### 1. Karakteristik Utama List

- **Mutable:** Elemen penyusun di dalam list dapat ditambah, diganti, maupun dihapus sewaktu-waktu.
- **Ordered:** Urutan data tetap konsisten berdasarkan urutan penulisan indeks (dimulai dari indeks 0).
- **Allow Duplicates:** Dapat menampung nilai elemen yang sama berulang kali.

### 2. Manipulasi Kode Menggunakan Metode Bawaan List

```
1 # Deklarasi list awal
2 kursus_web = ["HTML", "CSS", "Python"]
3
4 # 1. Menambahkan item baru di baris paling akhir
5 kursus_web.append("Django")
6
7 # 2. Mengubah nilai elemen pada indeks spesifik
8 kursus_web[1] = "Tailwind CSS"
9
10 # 3. Menghapus item berdasarkan nama nilai
11 kursus_web.remove("HTML")
12
13 print("Isi List Terbaru:", kursus_web)
14 # Output: ['Tailwind CSS', 'Python', 'Django']
```

## Struktur Data Koleksi: Tuple vs Set

Memilih struktur data yang tepat di Python sangat dipengaruhi oleh kebutuhan keamanan data serta kecepatan eksekusi proses pengolahan data.

### 1. Struktur Data Tuple (Koleksi Konstan)

Tuple didefinisikan menggunakan kurung biasa (). Karakter utamanya adalah **Immutable**, yaitu nilainya tidak akan pernah bisa diubah sama sekali setelah selesai dideklarasikan di memori. Sangat berguna untuk menjamin keamanan parameter data konstan sistem.

### 2. Struktur Data Set (Koleksi Unik)

Set didefinisikan menggunakan kurung kurawal {}. Karakter utamanya adalah **Unordered** (tidak berindeks) dan **Unik** (otomatis menolak dan menghapus elemen duplikat yang dimasukkan ke dalamnya). Sangat efisien untuk menyaring keunikan sekumpulan data.

### 3. Contoh Penerapan Kode Sintaksis

```
1 # Demonstrasi Tuple (Sifat Immutable)
2 koordinat_kantor = (-6.2001, 106.8166)
3 # koordinat_kantor[0] = -6.5000 -> Akan menghasilkan TypeError!
4
5 # Demonstrasi Set (Sifat Unik)
6 id_pengunjung = {101, 102, 101, 103}
7 print(id_pengunjung) # Output: {101, 102, 103} (Duplikat disaring)
```

## Menguasai Dictionary (Pasangan Key-Value)

Dictionary adalah struktur data yang menyimpan informasi dalam bentuk pemetaan pasangan kunci (*\*key\**) dan nilai (*\*value\**), mirip seperti kamus fisik. Kita tidak menggunakan indeks angka untuk mengakses nilai, melainkan menggunakan nama kunci yang kita buat sendiri.

### 1. Aturan Penulisan Dictionary

- Penulisan dibungkus menggunakan tanda kurung kurawal {}.
- Antara *\*key\** dan *\*value\** dipisahkan oleh simbol titik dua (:).
- Setiap nama *\*key\** dalam satu objek dictionary harus bersifat unik dan konsisten.

### 2. Contoh Implementasi Struktur Kode

```
1 # Membuat dictionary profil peserta kelas
2 peserta = {
3     "nama": "Andi Kusuma",
4     "kelas": "Python Masterclass",
5     "nilai_ujian": 92
6 }
7
8 # Mengakses nilai berdasarkan kunci 'nama'
9 print("Nama Peserta:", peserta["nama"])
10
11 # Menambahkan pasangan key-value baru secara dinamis
12 peserta["status"] = "Lulus"
13
14 print("Isi Data Peserta Terbaru:", peserta)
```

## Studi Kasus: Sistem Manajemen Inventori Produk

Mari kita bangun aplikasi mini pengolahan data inventori gudang produk Kusuma Web yang memanfaatkan gabungan fungsionalitas koleksi `List` dan `Dictionary`.

### 1. Skenario Desain Data

Kita akan menyimpan kumpulan data produk di dalam sebuah `List`. Di dalam list tersebut, setiap item produk secara detail dimodelkan sebagai objek `Dictionary` mandiri.

### 2. Kode Lengkap Program

```
1 # Inisialisasi Database Produk Toko Kusuma Web
2 inventori_toko = [
3     {"id": "KSM01", "nama": "Modul Python", "stok": 15},
4     {"id": "KSM02", "nama": "Ebook Django", "stok": 8}
5 ]
6
7 print("=== DAFTAR DETAIL INVENTORI KUSUMA WEB ===")
8
9 # Iterasi menampilkan isi database terstruktur
10 for produk in inventori_toko:
11     print(f"ID Produk: {produk['id']} | Nama: {produk['nama']} | Sisa Stok: {
12         produk['stok']} unit")
13
14 print("\n--- UPDATE DATA STOK PRODUK KSM01 ---")
15 # Mencari dan memperbarui nilai stok produk KSM01 secara langsung
16 for produk in inventori_toko:
17     if produk["id"] == "KSM01":
18         produk["stok"] = 20 # Memperbarui nilai stok di memori
19
20 print("Data berhasil diupdate!\n")
21 print(inventori_toko[0]) # Menampilkan hasil perubahan data produk pertama
```