

MODUL 02: ANATOMI VARIABEL & TIPE DATA

MASTERCLASS PYTHON DASAR

Memahami Mekanisme Penyimpanan Memori Komputer secara Dinamis.
Eksplorasi Aturan Penulisan Variabel, Karakteristik Tipe Data Primitif, serta Metode
Casting Konversi Nilai.

Kusuma Web Edu Series

Kelas Dasar Pemrograman — Modul 2 dari 10
Tanggal Rilis: 29 Juni 2026

Konsep Alokasi Memori & Aturan Penamaan Variabel

Di dalam bahasa Python, **variabel** bertindak sebagai label referensi yang menunjuk ke lokasi memori tempat suatu nilai disimpan. Tidak seperti bahasa pemrograman berjenis statis (seperti C++ atau Java), Python mengadopsi mekanisme alokasi memori dinamis.

1. Aturan Resmi Penamaan Variabel (Naming Convention)

Agar kode Anda valid dan mudah dipahami oleh komunitas developer global, ikuti aturan wajib berikut:

- Karakter pertama **harus** berupa huruf (a-z, A-Z) atau karakter garis bawah (**underscore** `_`).
- Karakter kedua dan seterusnya boleh berupa kombinasi huruf, angka (0-9), atau **underscore**.
- Nama variabel **tidak boleh** diawali oleh angka (misal: `1nilai` adalah ilegal).
- Bersifat **Case-Sensitive**: Variabel `Kusuma` dan `kusuma` dianggap terpisah di memori.
- Tidak boleh menggunakan kata kunci sistem (**Reserved Keywords**) seperti `if`, `class`, `import`, atau `print`.

2. Gaya Penulisan Profesional (PEP 8)

- **snake_case**: Digunakan untuk penulisan variabel standar (contoh: `nama_lengkap`, `total_biaya`).
- **UPPERCASE**: Digunakan untuk menyatakan nilai konstanta (contoh: `PI = 3.14`, `GRAVITASI = 9.8`).

```
1 # Contoh Deklarasi Variabel yang Baik
2 nama_siswa = "Rian Kusuma"
3 umur_siswa = 17
4 _id_database = 90211
```

Eksplorasi Tipe Data Primitif Python

Python memiliki empat tipe data dasar (primitif) utama yang mempresentasikan informasi di dalam program secara efisien.

1. Mengetahui Empat Pilar Tipe Data

- **String (str):** Tipe data teks literal yang dibungkus oleh tanda kutip tunggal (') atau ganda ("). Digunakan untuk menyimpan informasi tekstual.
- **Integer (int):** Bilangan bulat positif atau negatif tanpa pecahan desimal. Ukuran memori akan menyesuaikan secara dinamis tergantung besarnya angka.
- **Float (float):** Bilangan riil yang mengandung bagian desimal (pecahan). Penulisan desimal menggunakan tanda titik (.).
- **Boolean (bool):** Menyimpan status kebenaran logika absolut, hanya berisi dua kemungkinan nilai: `True` atau `False` (pastikan huruf kapital di awal).

2. Contoh Implementasi Struktur Kode

```
1 # Representasi tipe data di memori
2 judul_modul = "Variabel dan Tipe Data" # str
3 jumlah_halaman = 5                    # int
4 rating_modul = 4.95                   # float
5 apakah_aktif = True                   # bool
6
7 # Menampilkan nilai ke layar
8 print(judul_modul)
9 print(jumlah_halaman)
```

Dynamic Typing

Di Python, Anda tidak perlu menulis tipe data sebelum variabel. Interpreter Python secara otomatis menebak tipe datanya saat program dieksekusi berdasarkan nilai yang Anda masukkan.

Pemeriksaan & Konversi Tipe Data (Casting)

Sebagai bahasa dinamis, penting bagi kita untuk dapat memeriksa tipe data saat aplikasi berjalan (**runtime**) serta mengubah bentuknya ketika diperlukan untuk keperluan kalkulasi.

1. Memeriksa Tipe Data dengan `type()`

Kita dapat melacak tipe data aktual suatu objek menggunakan fungsi bawaan `type()`. Hal ini sangat penting untuk mendeteksi kesesuaian nilai sebelum diolah.

2. Mekanisme Casting (Konversi Tipe Data)

Casting adalah proses memaksa suatu nilai berubah menjadi jenis tipe data baru yang diinginkan. Python menyediakan fungsi casting bawaan:

- `str(nilai)`: Mengubah data angka menjadi string teks.
- `int(nilai)`: Mengonversi data string atau float menjadi bilangan bulat (pecahan desimal akan dipangkas).
- `float(nilai)`: Mengubah bilangan bulat atau string numerik menjadi pecahan desimal.

3. Implementasi Kode Pemeriksaan & Casting

```
1 # Demonstrasi Casting Data
2 teks_angka = "250"
3 angka_riil = 99.9
4
5 # Konversi string ke integer
6 nilai_baru = int(teks_angka)
7 print(type(nilai_baru)) # Output: <class 'int'>
8
9 # Konversi float ke integer (pembulatan ke bawah)
10 angka_bulat = int(angka_riil)
11 print(angka_bulat) # Output: 99
```

Studi Kasus: Program Konversi & Kalkulasi Dinamis

Mari kita buat program interaktif yang menggabungkan seluruh konsep variabel, pemeriksaan tipe data, dan casting untuk menyelesaikan suatu kalkulasi nyata.

1. Skenario Masalah

Fungsi `input()` selalu mengembalikan data bertipe string teks (`str`). Jika kita ingin melakukan operasi matematika pada input pengguna, kita harus melakukan casting eksplisit ke tipe data numerik (`int` atau `float`).

2. Kode Lengkap Program

```
1 # Skenario: Kalkulator Usia Sederhana
2 print("=== PROGRAM LAYANAN KUSUMA WEB ===")
3
4 # 1. Mengambil input tahun lahir (masih bertipe String)
5 tahun_lahir_str = input("Masukkan Tahun Lahir Anda: ")
6
7 # 2. Melakukan casting dari string ke integer
8 tahun_lahir = int(tahun_lahir_str)
9
10 # 3. Menghitung usia saat ini (asumsi tahun 2026)
11 tahun_sekarang = 2026
12 usia_saat_ini = tahun_sekarang - tahun_lahir
13
14 # 4. Menampilkan hasil dengan casting kembali ke string
15 print("-----")
16 print("Halo! Usia Anda saat ini adalah: " + str(usia_saat_ini) + " tahun.")
17 print("Tipe data variabel usia_saat_ini: " + str(type(usia_saat_ini)))
18 print("=====")
```

Rangkuman Modul 02

Anda sekarang memahami bagaimana variabel mengalokasikan data secara dinamis di memori, mengenali tipe data dasar, dan mengubah bentuknya secara aman. Langkah berikutnya pada Modul 03 adalah mempelajari cara melakukan manipulasi dan komparasi nilai menggunakan Operator!