

DASAR PEMROGRAMAN KOTLIN

Modul 09: Pengenalan OOP - Class, Object, & Constructors

Disusun Oleh:

Kusuma Web Academy

PROTECTED WATERMARK - KUSUMA WEB

Konsep Dasar Pemrograman Berorientasi Objek (OOP)

OOP merupakan paradigma pemrograman yang berorientasi pada struktur representasi objek di dalam kehidupan nyata secara terorganisir.

▶ **Class (Kelas):**

- ▶ Berperan sebagai cetak biru konseptual (**blueprint**) atau template desain utama.
- ▶ Mendefinisikan status data (**properties**) dan tindakan perilaku (**methods/functions**).

▶ **Object (Objek):**

- ▶ Bentuk fisik nyata (**instance**) dari rancangan konseptual cetak biru Class yang menempati ruang memori komputer.
- ▶ Satu kelas dapat melahirkan objek-objek fisik tak terbatas dengan karakteristik data yang berbeda-beda.

Struktur Konstruktor Canggih di Kotlin

Kotlin menawarkan keunggulan deklarasi parameter pembangun objek yang jauh lebih ringkas dibanding bahasa pemrograman lain.

- ▶ **Primary Constructor (Konstruktor Utama):**

- ▶ Ditulis langsung di bagian kepala kelas, tepat di sebelah nama Class dideklarasikan. Properti variabel dapat dideklarasikan sekaligus di sini menggunakan `val` atau `var`.

- ▶ **init Block (Blok Inisialisasi):**

- ▶ Blok kode khusus yang otomatis dipicu pengekseskusion kodenya seketika setelah objek fisik berhasil terbentuk sempurna di memori.

- ▶ **Secondary Constructor:**

- ▶ Konstruktor alternatif tambahan apabila Anda membutuhkan variasi parameter inisialisasi yang berbeda.

Praktik Implementasi Blueprint OOP

Berikut demonstrasi pembuatan Class, Primary Constructor, init, dan pembuatan objek fisik:

```
// Class dengan Primary Constructor
class Laptop(val merk: String, var ramGB: Int) {
    // Blok inisialisasi otomatis
    init {
        println("Laptop baru merk $merk sukses terdaftar di sistem!")
    }

    // Perilaku / Method
    fun upgradeRAM(tambahan: Int) {
        ramGB += tambahan
        println("RAM dinaikkan! Kapasitas RAM saat ini: ${ramGB}GB")
    }
}

fun main() {
    // Membuat Objek Fisik (Tanpa kata kunci 'new')
    val laptopDeri = Laptop("Asus ROG", 16)
    laptopDeri.upgradeRAM(16)
}
```