

# DASAR PEMROGRAMAN KOTLIN

## Modul 11: Konsep Pewarisan Sifat (Inheritance) & Polimorfisme

Disusun Oleh:

**Kusuma Web Academy**

PROTECTED WATERMARK - KUSUMA WEB

# Konsep Pewarisan di Kotlin

Pewarisan (*\*Inheritance\**) membolehkan kelas anak (*\*subclass\**) merepresentasikan, mengambil, dan mengembangkan fungsionalitas dari kelas induk (*\*superclass\**).

## ▶ Secara Default Bersifat Final:

- ▶ Di Kotlin, seluruh kelas secara bawaan berstatus terkunci (*\*final\**) demi menjamin keamanan optimasi program.

## ▶ Keyword `open`:

- ▶ Untuk mengizinkan suatu Class atau metode agar bisa diwarisi dan diubah implementasinya, wajib didahului oleh penulisan kata kunci pembuka `open`.

## ▶ Keyword `override`:

- ▶ Digunakan pada kelas turunan untuk menuliskan kembali metode/fungsi yang diturunkan oleh kelas induk dengan logika spesifik yang baru.

# Memahami Polimorfisme secara Ringkas

Polimorfisme memiliki arti "banyak bentuk". Ini memungkinkan objek-objek dengan tipe kelas anak yang berbeda untuk dieksekusi melalui satu interface kelas induk yang sama.

## Konsep Utama Polimorfisme

Dengan polimorfisme, Anda bisa mendefinisikan array atau list bertipe superclass, namun mengisinya dengan berbagai instansi objek subclass yang beragam. Ketika metode yang sama dipanggil, masing-masing objek akan mengeksekusi logika internalnya sendiri.

# Praktik Implementasi Pewarisan Kelas

Contoh relasi inheritance lengkap di editor program:

```
// Superclass terbuka
open class AnggotaKusuma(val nama: String) {
    open fun aksesMateri() {
        println("$nama memiliki akses ke portal belajar standar.")
    }
}

// Subclass turunan
class VIPStudent(nama: String, val diskonKhusus: Int) : AnggotaKusuma(
    nama) {
    // Melakukan modifikasi method superclass
    override fun aksesMateri() {
        println("$nama memiliki akses belajar VIP tanpa batas, plus
            diskon kursus sebesar $diskonKhusus%.")
    }
}

fun main() {
    val siswaBiasa = AnggotaKusuma("Budi")
    val siswaVIP = VIPStudent("Budi", 10)
}
```