

# DASAR PEMROGRAMAN KOTLIN

## Modul 12: Abstract Class & Interface

Disusun Oleh:

**Kusuma Web Academy**

PROTECTED WATERMARK - KUSUMA WEB

# Struktur Arsitektur Abstract Class

Abstract Class bertindak sebagai rancangan struktural konseptual yang menampung fungsi abstrak tanpa tubuh implementasi nyata.

- ▶ **Tidak Bisa Diinstansiasi:**

- ▶ Anda tidak diizinkan membuat objek fisik langsung dari abstract class.

- ▶ **Method Abstrak:**

- ▶ Ditandai dengan kata kunci `abstract`. Fungsi abstrak ini **wajib** ditulis ulang (*\*override\**) oleh kelas anak yang mengimplementasikannya.

- ▶ **Method Biasa:**

- ▶ Bisa menampung fungsi biasa lengkap dengan tubuh perintah kodenya demi efisiensi kode terpusat.

# Kontrak Kerja Struktur: Interface

Interface menetapkan kontrak kelakuan (*\*behavior\**) yang wajib diimplementasikan oleh suatu kelas tanpa terikat relasi warisan vertikal tunggal.

## ▶ **Multiple Implementation (Implementasi Ganda):**

- ▶ Sebuah kelas hanya bisa mewarisi satu superclass saja, namun dibebaskan untuk mengimplementasikan antarmuka (*\*interface\**) dalam jumlah tak terbatas secara bersamaan.

## ▶ **Implementasi default:**

- ▶ Di Kotlin, fungsi di dalam interface diizinkan memiliki isi program default (*\*default body implementation\**).

# Praktik Implementasi Kontrak Abstraksi

Contoh penulisan integrasi abstract class dan interface di Kotlin:

```
interface Pembayar {
    fun bayarTagihan() // Kontrak fungsi murni
}

abstract class AkunBelajar(val ID: String) {
    abstract fun loginAccess() // Fungsi abstrak wajib di-override
}

class MuridEmas(ID: String) : AkunBelajar(ID), Pembayar {
    override fun loginAccess() {
        println("Murid $ID berhasil masuk melalui portal sistem.")
    }
    override fun bayarTagihan() {
        println("Tagihan sekolah Murid $ID sukses terbayar.")
    }
}

fun main() {
    val deri = MuridEmas("K-1002")
}
```