

# DASAR PEMROGRAMAN KOTLIN

Modul 04: Alur Struktur Kontrol Percabangan (If-Else & When)

Disusun Oleh:

**Kusuma Web Academy**

PROTECTED WATERMARK - KUSUMA WEB

# Logika Percabangan: Statement vs Ekspresi

Di dalam Kotlin, struktur percabangan memiliki kedudukan yang lebih tinggi karena dapat digunakan sebagai penghasil nilai (\*ekspresi\*).

## ▶ Sebagai Statement (Pernyataan):

- ▶ Berfungsi seperti percabangan standar bahasa pemrograman tradisional untuk mengontrol alur eksekusi baris perintah berdasarkan kondisi boolean.

## ▶ Sebagai Expression (Ekspresi):

- ▶ Karena Kotlin tidak mendukung operator ternary (? :), fungsi penentuan nilai kondisional digantikan secara penuh oleh `if-else` sebagai ekspresi.
- ▶ Hasil evaluasi kondisi dapat langsung ditugaskan/disimpan ke dalam variabel tujuan secara instan.

# Struktur `when` yang Modern & Ringkas

Keyword `when` menggantikan sistem `switch-case` tradisional dari Java dengan paradigma pemrograman yang jauh lebih aman dan ekspresif.

## Kelebihan Utama `when`

Tidak membutuhkan kata kunci penutup `break` di setiap akhir cabang keputusan. Kompiler akan menghentikan evaluasi secara otomatis ketika kondisi pertama yang sesuai sudah berhasil dipenuhi.

## Format Penulisan:

- ▶ Masing-masing cabang dideklarasikan menggunakan relasi panah khusus (`->`).
- ▶ Wajib menyertakan cabang alternatif penutup `else` jika `when` digunakan sebagai ekspresi guna menjamin kepastian kembalian nilai.

# Pemeriksaan Lanjut pada Ekspresi when

Struktur `when` tidak hanya terbatas pada pencocokan nilai statis biasa, melainkan mendukung berbagai jenis evaluasi dinamis:

- ▶ **Multi-Value Checking (Banyak Nilai):**

- ▶ Mengevaluasi beberapa nilai kecocokan sekaligus di dalam satu cabang yang sama dengan memisahkan nilai menggunakan tanda koma (,).

- ▶ **Range Checking (Jangkauan Angka):**

- ▶ Mengevaluasi apakah suatu nilai numerik berada di dalam jangkauan angka tertentu menggunakan operator asimilasi `in` atau `!in`.

- ▶ **Type Checking (Kesesuaian Tipe):**

- ▶ Mengevaluasi kesesuaian tipe data objek secara langsung menggunakan operator identifikasi objek `is` atau `!is`.

# Praktik Implementasi Logika Percabangan

Contoh penulisan program percabangan lengkap untuk latihan di laboratorium:

```
fun main() {
    val nilaiTeori = 82

    // If-Else Bertindak Sebagai Ekspresi Pengembalian Nilai
    val statusKelulusan = if (nilaiTeori >= 75) "LULUS" else "REMEDIAL"

    // Evaluasi Menggunakan When-Expression
    val predikatHasil = when (nilaiTeori) {
        in 90..100 -> "A (Sangat Baik)"
        in 80..89  -> "B (Baik)"
        in 70..79  -> "C (Cukup)"
        else       -> "D (Butuh Perbaikan)"
    }

    println("Status Siswa: $statusKelulusan dengan Predikat:
            $predikatHasil")
}
```