

1. Struktur if di Kotlin

Struktur `if` di Kotlin mirip dengan bahasa pemrograman lain, tetapi dengan beberapa perbedaan unik.

a. `if` sebagai Pernyataan

Pada dasarnya, `if` digunakan untuk memeriksa kondisi tertentu. Jika kondisi terpenuhi, blok kode yang sesuai akan dijalankan.

```
if (condition) {  
    // Eksekusi jika kondisi true  
} else {  
    // Eksekusi jika kondisi false  
}
```

b. `if` sebagai Ekspresi

Salah satu fitur unik Kotlin adalah bahwa `if` dapat digunakan sebagai ekspresi, yang berarti ia mengembalikan nilai dan bisa digunakan untuk menginisialisasi variabel.

```
val max = if (a > b) a else b
```

Ini mirip dengan **ternary operator** (`? :`) di bahasa lain seperti Java, meskipun Kotlin tidak memiliki operator ternary secara eksplisit.

Contoh `if` Sederhana

```
fun main() {  
    val number = 10  
  
    if (number > 0) {  
        println("Angka positif")  
    } else if (number < 0) {  
        println("Angka negatif")  
    } else {  
        println("Angka nol")  
    }  
}
```

Contoh `if` sebagai Ekspresi

```
fun main() {  
    val a = 5  
    val b = 10  
  
    val max = if (a > b) a else b  
    println("Nilai maksimum adalah $max")  
}
```

2. Struktur when di Kotlin

when di Kotlin adalah struktur kontrol yang lebih kuat dibandingkan **switch** di bahasa lain seperti Java atau C. **when** dapat digunakan untuk memeriksa nilai yang berbeda atau kondisi yang lebih kompleks.

a. when sebagai Pengganti switch

when di Kotlin sering digunakan sebagai pengganti `switch` dan bisa menangani berbagai tipe data.

```
when (expression) {  
    value1 -> { // Blok kode jika value1 cocok }  
    value2 -> { // Blok kode jika value2 cocok }  
    else -> { // Blok kode jika tidak ada yang cocok }  
}
```

b. when sebagai Ekspresi

Seperti `if`, `when` juga bisa digunakan sebagai ekspresi dan mengembalikan nilai.

```
val result = when (expression) {  
    value1 -> result1  
    value2 -> result2  
    else -> defaultResult  
}
```

Contoh when Sederhana

```
fun main() {  
    val day = 3  
  
    when (day) {  
        1 -> println("Senin")  
        2 -> println("Selasa")  
        3 -> println("Rabu")  
        4 -> println("Kamis")  
        5 -> println("Jumat")  
        6, 7 -> println("Akhir Pekan") // Menangani beberapa kondisi  
        else -> println("Hari tidak valid")  
    }  
}
```

Contoh when sebagai Ekspresi

```
fun main() {  
    val number = 4  
  
    val result = when (number) {  
        1 -> "Satu"  
        2 -> "Dua"
```

```

        3 -> "Tiga"
        4 -> "Empat"
        else -> "Tidak dikenal"
    }

    println("Angka tersebut adalah $result")
}

```

c. when dengan Kondisi

Anda juga dapat menggunakan `when` dengan kondisi lebih kompleks, tidak terbatas pada nilai.

```

fun main() {
    val number = -5

    when {
        number > 0 -> println("Angka positif")
        number < 0 -> println("Angka negatif")
        else -> println("Angka nol")
    }
}

```

d. when tanpa Argumen

Jika Anda tidak perlu mengevaluasi ekspresi secara langsung, Anda bisa menggunakan `when` tanpa argumen.

```

fun main() {
    val input = readLine()

    when {
        input.isNullOrEmpty() -> println("Input kosong")
        input.length > 5 -> println("Input panjang")
        else -> println("Input valid")
    }
}

```

3. Perbandingan if vs when

- Gunakan `if` ketika Anda hanya perlu memeriksa satu atau dua kondisi sederhana.
 - Gunakan `when` ketika Anda memiliki banyak nilai atau kondisi yang berbeda untuk diperiksa, atau ketika Anda menginginkan cara yang lebih bersih dan terorganisir untuk menangani beberapa kondisi.
-

4. Contoh Kasus Penggunaan

a. Kasus dengan if

```
fun categorizeAge(age: Int) {  
    val category = if (age < 13) {  
        "Anak-anak"  
    } else if (age in 13..19) {  
        "Remaja"  
    } else if (age in 20..60) {  
        "Dewasa"  
    } else {  
        "Lansia"  
    }  
    println("Kategori umur: $category")  
}  
  
fun main() {  
    categorizeAge(25)  
}
```

b. Kasus dengan when

```
fun categorizeDay(day: String) {  
    when (day.lowercase()) {  
        "senin", "selasa", "rabu", "kamis", "jumat" -> println("Hari kerja")  
        "sabtu", "minggu" -> println("Akhir pekan")  
        else -> println("Hari tidak valid")  
    }  
}  
  
fun main() {  
    categorizeDay("Sabtu")  
}
```

5. Kesimpulan

- **if** digunakan untuk memeriksa kondisi sederhana dan bisa bertindak sebagai ekspresi untuk mengembalikan nilai.
- **when** adalah struktur yang lebih kuat untuk memeriksa beberapa kondisi atau nilai dan lebih fleksibel daripada switch di bahasa pemrograman lain.
- **when** juga bisa digunakan tanpa argumen untuk melakukan kondisi lebih kompleks, sedangkan if lebih sederhana dan cocok untuk kondisi sederhana.