

10. CONTOH APLIKASI REAKTIF

Pemrograman Sistem Interaktif



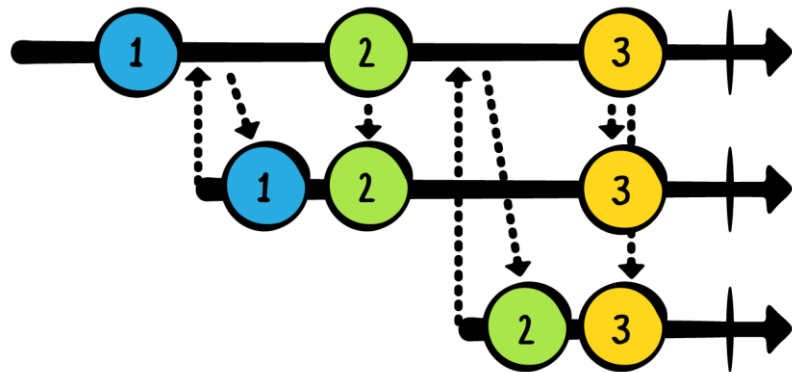
Capaian Pembelajaran

- Mampu menerapkan konsep sistem reaktif dengan pola perancangan Observer dan teknik pemrograman reaktif

Sub-Capaian Pembelajaran

- Mampu menerapkan komponen Subject pada pustaka Rx untuk aplikasi interaktif sederhana berbasis Android

BehaviorSubject



Events yang akan diterima *subscriber* dimulai dari event terbaru sebelum subscribe.

Objek BehaviorSubject Pada ViewModel

```
class PengunjungViewModel : ViewModel() {  
  
    ...  
  
    /* VERSI REAKTIF */  
  
    // Mendeklarasikan variabel untuk menyimpan jumlah pengunjung  
    private var jumlahReaktif: Int = 0  
  
    // Mendeklarasikan Subject untuk memancarkan jumlah pengunjung  
    private val jumlahSubject: BehaviorSubject<Int> = BehaviorSubject.createDefault(0)  
  
    // Untuk mengakses Subject dari Activity  
    fun getJumlahSubject(): BehaviorSubject<Int> {  
        return jumlahSubject  
    }  
  
    fun tambahReaktif(): Unit {  
        jumlahReaktif += 1  
        jumlahSubject.onNext(jumlahReaktif)  
    }  
  
    fun kurangReaktif(): Unit {  
        jumlahReaktif -= 1  
        jumlahSubject.onNext(jumlahReaktif)  
    }  
}
```

Subscriber Pada Activity

```
class MainActivity : AppCompatActivity() {  
    ...  
    /* REAKTIF */  
  
    // Mendeklarasikan TextView untuk tampilan Reaktif  
    var tvJumlahReaktif = findViewById<TextView>(R.id.jumlah_reaktif)  
  
    // Berlangganan setiap terjadi perubahan jumlah pengunjung dan bereaksi dengan mengubah nilai pada TextView  
    model.getJumlahSubject().subscribe {  
        tvJumlahReaktif.setText(it.toString())  
    }.addTo(disposable)  
  
    /* OPEARSI UMUM UNTUK PENAMBAHAN DAN PENGURANGAN */  
    findViewById<Button>(R.id.btn_tambah).setOnClickListener {  
        ...  
        // Operasi reaktif  
        model.tambahReaktif()  
    }  
    findViewById<Button>(R.id.btn_kurang).setOnClickListener {  
        ...  
        // Operasi reaktif  
        model.kurangReaktif()  
    }  
}  
override fun onDestroy() {  
    // Pastikan setelah selesai semua subscriber di-dispose agar tidak terjadi leak memory  
    disposable.dispose()  
    super.onDestroy()  
}  
}
```

SEKIAN

Terima Kasih