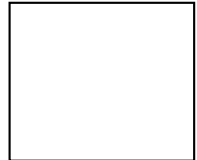




LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : OBSERVABLE
NAMA : PUTUT GURITNO BRAMANTYO
NIM : 145150207111061
KELOMPOK : 4
TANGGAL : 02/12/2020
ASISTEN : - IMAN HARIE NAWANTO
- ERSYA NADIA CANDRA



PROSEDUR 1 Membuat Observable sederhana yang meng-emit Next, complete, dan error

1. Salin kode ini ke dalam fungsi main, kemudian jalankan.

```
val observableSatu = Observable.create<String>{  
    it.onNext("Satu")  
    it.onNext("Dua")  
    it.onComplete()  
}  
val subscriberSatu = observableSatu.subscribeBy(  
    onNext = { println(it) },  
    onComplete = { println("Selesai") }  
)
```

2. Dari kode di atas dapat dilihat bahwa tipe dari variabel subscriberSatu ditentukan oleh tipe dari nilai balikan (return value) method subscribeBy. Dengan meletakkan cursor mouse di atas nama method subscribeBy, sebutkan tipe dari nilai balikan method tersebut:

Jawaban :
Disposable

3. Jalankan kode tersebut, kemudian tuliskan output yang ditampilkan.

Jawaban :

Menampilkan value dari it dan diakhiri dengan selesai yang menjadi tanda bahwa program telah selesai dijalankan

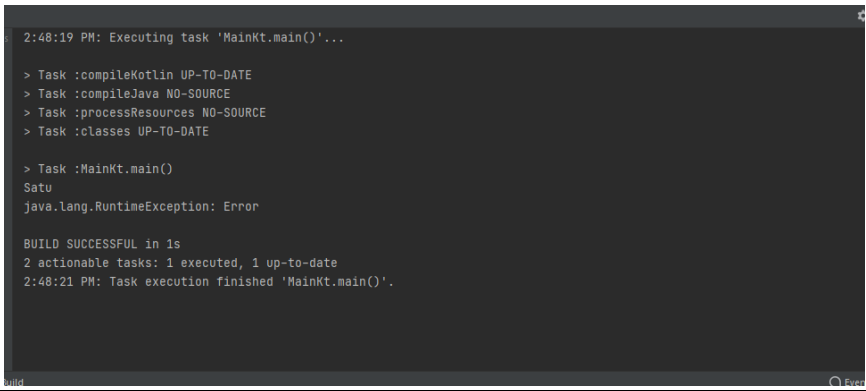
4. Sisipkan baris kode baru di antara onNext yang pertama dan kedua pada method create. Tambahkan pula baris baru setelah onComplete pada method subscribeBy. Sehingga kode akan seperti ini:

```
val observableSatu = Observable.create<String>{  
    it.onNext("Satu")  
    it.onError(RuntimeException("Error"))  
    it.onNext("Dua")  
    it.onComplete()  
}  
val subscriberSatu = observableSatu.subscribeBy(  
    onNext = { println(it) },  
    onComplete = { println("Selesai") },  
    onError = { println(it) }
```

)

5. Jalankan kode tersebut, kemudian tuliskan output yang ditampilkan.

Jawaban :



```
2:48:19 PM: Executing task 'MainKt.main()'...

> Task :compileKotlin UP-TO-DATE
> Task :compileJava NO-SOURCE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :MainKt.main()
Satu
java.lang.RuntimeException: Error

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
2:48:21 PM: Task execution finished 'MainKt.main()'.
```

6. Tuliskan kesimpulan apa yang dapat saudara tarik dari eksperimen tersebut.

Jawaban :

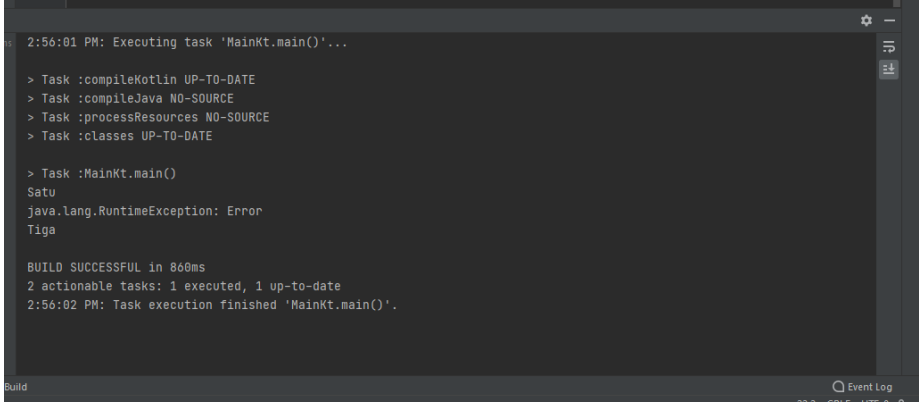
“Satu” tetap ditampilkan pada output. Namun karena “error” dideklarasikan setelah satu, maka sistem membaca bahwa setelah satu terjadi error dan “dua” dan “selesai” tidak ditampilkan

7. Tambahkan sebuah Observable beserta subscriber-nya seperti pada kode berikut:

```
val observableSatu = Observable.create<String>{
    it.onNext("Satu")
    it.onError(RuntimeException("Error"))
    it.onNext("Dua")
    it.onComplete()
}
val observableDua = Observable.create<String>{
    it.onNext("Tiga")
}
val subscriberSatu = observableSatu.subscribeBy(
    onNext = { println(it) },
    onComplete = { println("Selesai") },
    onError = { println(it) }
)
val subscriberDua = observableDua.subscribeBy(
    onNext = { println(it) }
)
```

8. Jalankan kode tersebut, kemudian tuliskan output yang ditampilkan.

Jawaban :



```
2:56:01 PM: Executing task 'MainKt.main()'...

> Task :compileKotlin UP-TO-DATE
> Task :compileJava NO-SOURCE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :MainKt.main()
Satu
Tiga
java.lang.RuntimeException: Error

BUILD SUCCESSFUL in 860ms
2 actionable tasks: 1 executed, 1 up-to-date
2:56:02 PM: Task execution finished 'MainKt.main()'.
```

9. Dapatkah saudara menemukan masalah yang ada pada observableDua? Sebutkan.

Jawaban :

Output dari observableDua tetap muncul karena dia bukan bagian dari observableSatu yang mendeklarasikan error

PROSEDUR 2: Mengelola subscriptions dengan CompositeDisposable

1. Tambahkan variabel bernama “subscriptions” dengan tipe CompositeDisposable untuk menampung seluruh subscriber yang ada.

```
fun main(args: Array<String>){  
  
    val subscriptions = CompositeDisposable()  
  
    val observableSatu = Observable.create<String>{
```

2. Dengan menggunakan method add pada objek CompositeDisposable, masukkan setiap subscriber ke dalam variabel subscriptions.

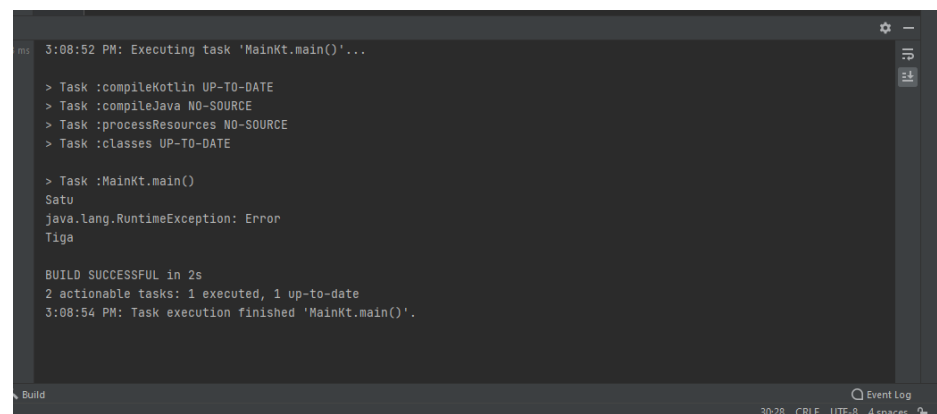
```
        val subscriberSatu = observableSatu.subscribeBy(  
            onNext = { println(it) },  
            onComplete = { println("Selesai") },  
            onError = { println(it) }  
        )  
        subscriptions.add(subscriberSatu)  
  
        val subscriberDua = observableDua.subscribeBy(  
            onNext = { println(it) }  
        )  
        subscriptions.add(subscriberDua)
```

3. Setelah itu panggil method dispose pada objek CompositeDisposable, yaitu variabel subscriptions, seperti pada kode berikut.

```
        val subscriberDua = observableDua.subscribeBy(  
            onNext = { println(it) }  
        )  
        subscriptions.add(subscriberDua)  
  
        subscriptions.dispose()
```

4. Jalankan kode tersebut, kemudian tuliskan output yang ditampilkan.

Jawaban :



```
3:08:52 PM: Executing task 'MainKt.main()'...  
  
> Task :compileKotlin UP-TO-DATE  
> Task :compileJava NO-SOURCE  
> Task :processResources NO-SOURCE  
> Task :classes UP-TO-DATE  
  
> Task :MainKt.main()  
Satu  
java.lang.RuntimeException: Error  
Tiga  
  
BUILD SUCCESSFUL in 2s  
2 actionable tasks: 1 executed, 1 up-to-date  
3:08:54 PM: Task execution finished 'MainKt.main()'.
```

5. Apakah ada perubahan dibandingkan output yang ditampilkan sebelumnya?

Jawaban :

Tidak ada perubahan output

6. Jelaskan perbedaan antara sebelum dan sesudah dikelola dengan CompositeDisposable

Jawaban :

PROSEDUR 3: Membuat Observable spesifik, yaitu Single

1. Sisipkan kode berikut ini untuk membuat Single beserta observer/subscriber-nya.

```
subscriptions.add(subscriberDua)

fun loadText(filename: String): Single<String> {
    return Single.create { emitter ->
        val file = File(filename)
        if(!file.exists()){
            emitter.onError(FileNotFoundException("File tidak ditemukan $filename"))
            return@create
        }
        val content = file.readText(Charsets.UTF_8)
        emitter.onSuccess(content)
    }
}

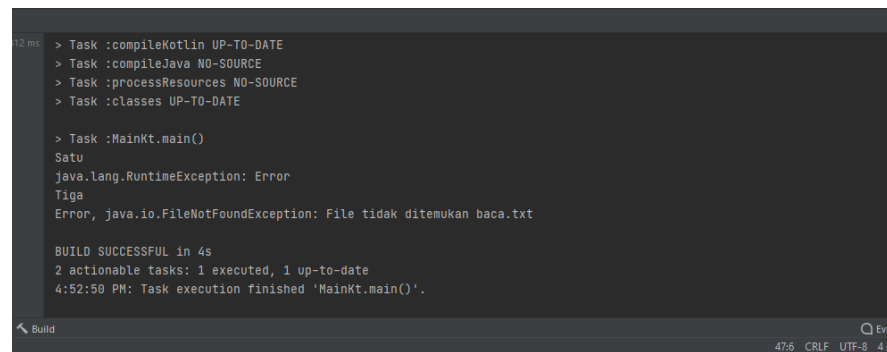
val subscriberText = loadText("baca.txt").subscribeBy(
    onSuccess = {println(it)},
    onError = {println("Error, $it")}
)

subscriptions.add(subscriberText)

subscriptions.dispose()
```

2. Jalankan kode tersebut, apakah output yang dikeluarkan oleh subscriberText?

Jawaban :



```
12 ms > Task :compileKotlin UP-TO-DATE
> Task :compileJava NO-SOURCE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :MainKt.main()
Satu
java.lang.RuntimeException: Error
Tiga
Error, java.io.FileNotFoundException: File tidak ditemukan baca.txt

BUILD SUCCESSFUL in 4s
2 actionable tasks: 1 executed, 1 up-to-date
4:52:50 PM: Task execution finished 'MainKt.main()'.

Build 47:6 CRLF UTF-8 4 s
```

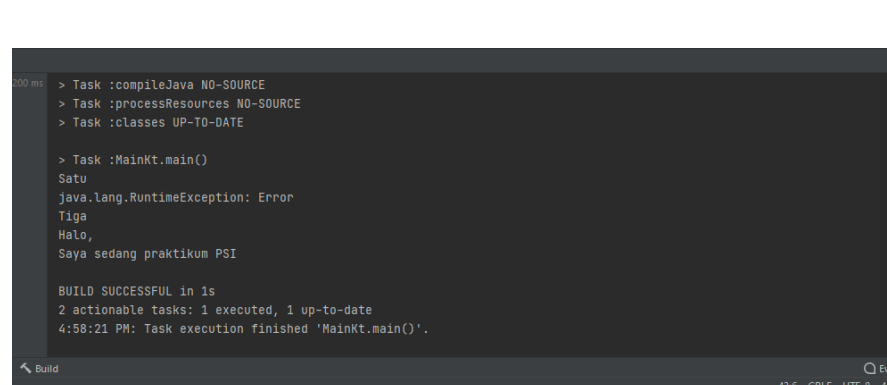
3. Pada root folder dari project (misal: C:\Users\user\IdeaProjects\observable), buatlah sebuah file teks baru dengan nama “baca.txt”. Kemudian isi dengan teks sbb:

Jawaban :

```
Halo,
Saya sedang praktikum PSI
```

4. Jalankan kode tersebut, apakah output yang dikeluarkan oleh subscriberText? Sertakan pula screenshot output secara keseluruhan.

Jawaban :



```
250 ms > Task :compileJava NO-SOURCE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :MainKt.main()
Satu
java.lang.RuntimeException: Error
Tiga
Halo,
Saya sedang praktikum PSI

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
4:58:21 PM: Task execution finished 'MainKt.main()'.

Build 43:6 CRLF UTF-8 4 s
```