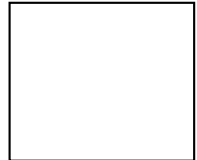




LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

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



PROSEDUR 1: Membuat PublishSubject

1. Salin kode ini ke dalam fungsi main.

```
fun main(args: Array<String>){  
    val subscriptions = CompositeDisposable()  
  
    val subject = PublishSubject.create<String>()  
  
    subject.onNext("Satu")  
    subject.onNext("Dua")  
    subject.onNext("Tiga")  
  
    val subscriberSatu = subject.subscribeBy(  
        onNext = {println("S1: $it")}  
    )  
    subscriptions.add(subscriberSatu)  
  
    subject.onNext("Empat")  
  
    val subscriberDua = subject.subscribeBy(  
        onNext = {println("S2: $it")}  
    )  
    subscriptions.add(subscriberDua)  
  
    subscriptions.dispose()  
}
```

2. Tuliskan output yang dicetak oleh semua subscriber. Adakah subscriber yang tidak mencetak apapun di layar? Jelaskan alasannya.

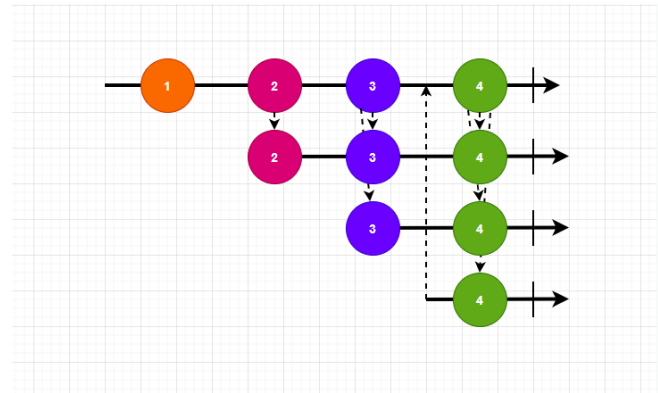
Jawaban :

```
24 | subscriptions.add(subscriberDua)  
ST ms |  
> Task :compileJava NO-SOURCE  
> Task :processResources NO-SOURCE  
> Task :classes UP-TO-DATE  
  
> Task :MainKt.main()  
S1: Empat  
  
BUILD SUCCESSFUL in 11s  
2 actionable tasks: 1 executed, 1 up-to-date  
5:52:14 PM: Task execution finished 'MainKt.main()'.  
Build | Event Log  
CRLF UTF-8 4 spaces
```

Subscriber satu tidak mencetak apapun karena yang dicetak hanya elemen terbaru pada subscribers yaitu empat

3. Gambarlah kode di atas dalam bentuk marble diagram

Jawaban :



PROSEDUR 2: Membuat BehaviorSubject

1. Pada kode sebelumnya, ubahlah tipe variabel subject yang tadinya PublishSubject menjadi BehaviorSubject, seperti kode berikut:

Jawaban :

```
val subscriptions = CompositeDisposable()

val subject = PublishSubject.create<String>()
```

Menjadi

```
val subscriptions = CompositeDisposable()

val subject = BehaviorSubject.create<String>()
```

2. Tuliskan output yang dicetak oleh semua subscriber. Adakah subscriber yang tidak mencetak apapun di layar? Jelaskan alasannya.

Jawaban :

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

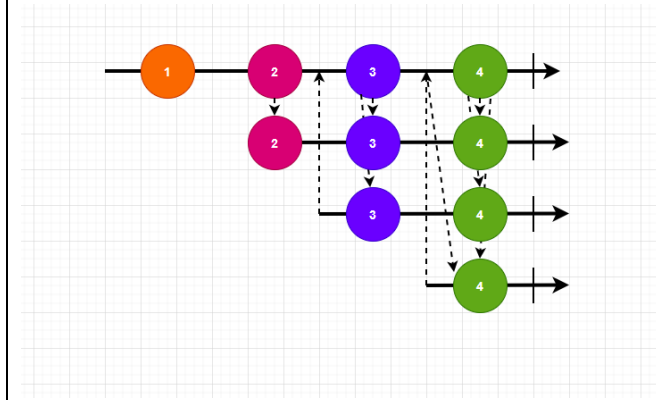
> Task :MainKt.main()
S1: Tiga
S1: Empat
S2: Empat

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
5:59:56 PM: Task execution finished 'MainKt.main()'.
```

Subscriber satu hanya mengeluarkan “tiga” karena behavior subject hanya mengembalikan nilai terakhir dari subject

3. Gambarlah kode di atas dalam bentuk marble diagram

Jawaban :



PROSEDUR 3: Membuat ReplaySubject

1. Pada kode sebelumnya, ubahlah tipe variabel subject yang tadinya BehaviorSubject menjadi ReplaySubject dengan buffer bernilai 2, seperti kode berikut:

Jawaban :

```
val subscriptions = CompositeDisposable()

val subject = BehaviorSubject.create<String>()
```

Menjadi

```
val subscriptions = CompositeDisposable()

val subject = ReplaySubject.createWithSize<String>(maxSize: 2)
```

2. Tuliskan output yang dicetak oleh semua subscriber.

Jawaban :

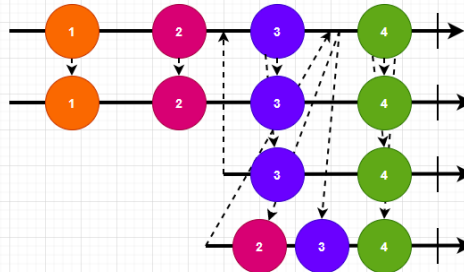
```
ms > Task :MainKt.main()
S1: Dua
S1: Tiga
S1: Empat
S2: Tiga
S2: Empat

BUILD SUCCESSFUL in 11s
2 actionable tasks: 2 executed
6:13:50 PM: Task execution finished 'MainKt.main()'.

Build 10:58 CRLF UTF-8 4 spa
```

3. Gambarlah kode di atas dalam bentuk marble diagram

Jawaban :



4. Pada contoh kode tersebut, berapakah nilai buffer terkecil sehingga semua nilai yang dipancarkan oleh subject dicetak oleh semua subscriber:

Jawaban : nilai buffernya 4 sehingga semua nilai bisa dicetak semua subscriber

```
fun main(args: Array<String>){
    val subscriptions = CompositeDisposable()
    val subject = ReplaySubject.createWithSize<String>(maxSize: 4)

    subject.onNext("Satu")
    subject.onNext("Dua")
    subject.onNext("Tiga")

    val subscriberSatu = subject.subscribeBy(
        onNext = {println("S1: $it")}
    )

    > Task :MainKt.main()
    S1: Satu
    S1: Dua
    S1: Tiga
    S1: Empat
    S2: Satu
    S2: Dua
    S2: Tiga
    S2: Empat

    BUILD SUCCESSFUL in 7s
    2 actionable tasks: 1 executed, 1 up-to-date
    6:40:52 PM: Task execution finished 'MainKt.main()'.
```

PROSEDUR 4: Membuat AsyncSubject

1. Pada kode sebelumnya, ubahlah tipe variabel subject yang tadinya ReplaySubject menjadi AsyncSubject, seperti kode berikut:

Jawaban :

```
val subscriptions = CompositeDisposable()
val subject = ReplaySubject.createWithSize<String>(maxSize: 2)
```

Menjadi

```
val subscriptions = CompositeDisposable()
val subject = AsyncSubject.create<String>()
```

2. Adakah output yang dicetak oleh subscriber? Jelaskan alasannya.

Jawaban :

Tidak ada output karena subject ini hanya mencetak nilai terakhir dari eksekusi dan nilai muncul ketika ada perintah eksekusi harus berhenti

3. Sisipkan sebaris kode yang dapat mengakibatkan semua subscriber mencetak output di layar. Tuliskan kode yang harus disisipkan pada posisi tersebut

Jawaban : penambahan dilakukan untuk menjadi tanda bahwa program telah selesai dieksekusi

```
        println("S1: Empat")
    }
    subscriptions.add(subscriberDua)
    subject.onComplete()
    subscriptions.dispose()
}
```

4. Tuliskan output yang dicetak oleh semua subscriber.

Jawaban :

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

> Task :MainKt.main()
S1: Empat
S2: Empat

BUILD SUCCESSFUL in 7s
2 actionable tasks: 2 executed
7:01:27 PM: Task execution finished 'MainKt.main()'.

Build
```

5. Gambarlah kode di atas dalam bentuk marble diagram.

Jawaban :

