

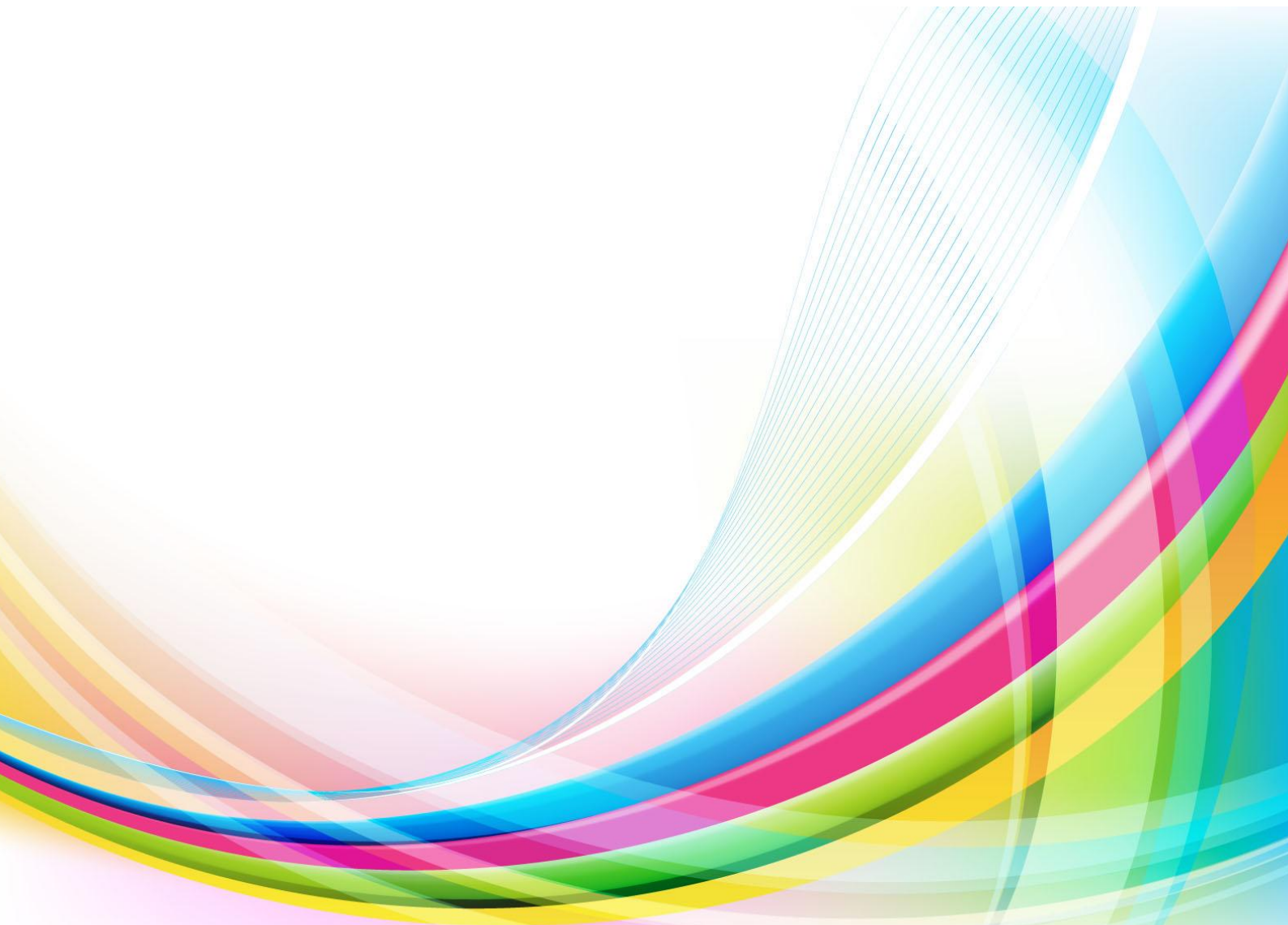


## **MODUL PRAKTIKUM 5**

## **FRAGMENT**

## **BAHASA PEMROGRAMAN II S1 SI**

**STEVI EMA WIJAYANTI**



## FRAGMENT

### Overview :

Pada modul ini akan dibahas mengenai Fragment serta implementasinya dalam Android Studio. Fragment merupakan salah satu komponen pada Android Studio dengan fungsi yang hampir sama seperti activity tetapi memiliki “*lifecycle*” atau siklus hidup yang berbeda

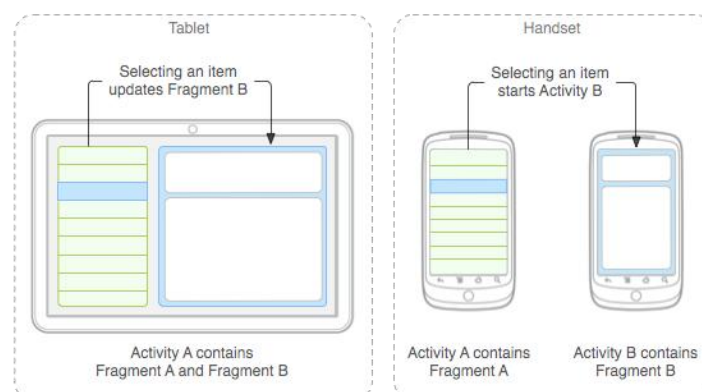
### Tujuan dari praktikum 5 :

1. Mengetahui definisi fragment dan siklus hidupnya (*life cycle*)
2. Mampu mendesain UI (*User Interface*) dan mengimplementasikan fragment pada Android Studio
3. Mengetahui perbedaan activity dan fragment serta penggunaannya dalam Android Studio

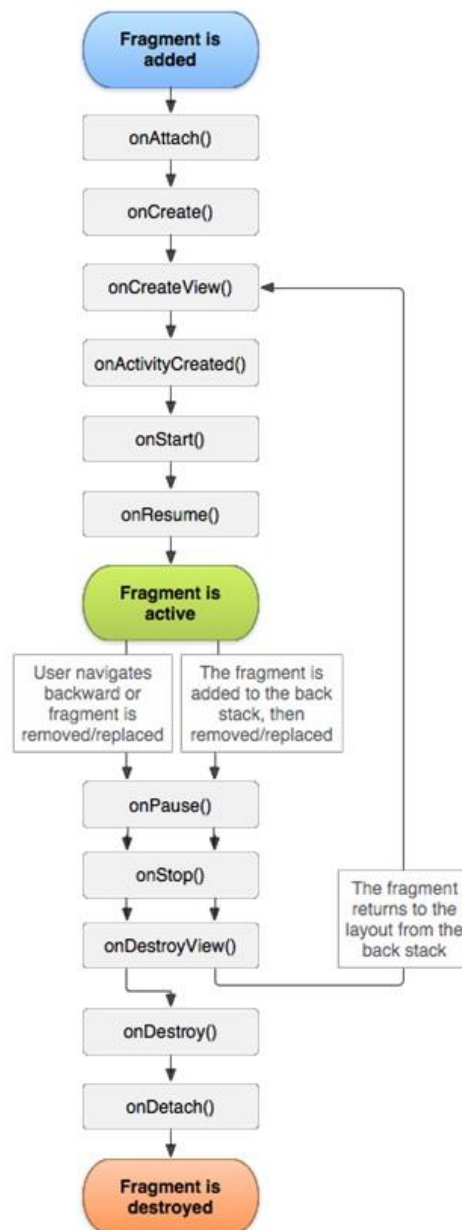
## DEFINISI FRAGMENT

Fragment merupakan salah satu komponen pada Android Studio dengan fungsi yang hampir sama seperti activity tetapi memiliki “*lifecycle*” atau siklus hidup yang berbeda. Fragment merupakan bagian dari sebuah activity karena fragment membutuhkan akses dari activity untuk dapat dijalankan atau dapat dikatakan bahwa Fragment semacam sub activity. Anda dapat menggabungkan beberapa Fragment dalam satu activity untuk membangun *User Interface* multi Panel dan menggunakan kembali Fragment tersebut ke dalam beberapa activity. Dengan Fragment memungkinkan seorang developer membuat lebih banyak “desain activity secara modular”. Fragment dalam Android dapat digunakan mulai dari versi Android Honeycombg (API versi 11). Kelebihan dari menerapkan Fragment antara lain :

1. Tidak perlu memasukkan nama file fragment ke dalam “AndroidManifest” yang diperlukan oleh activity.
2. Fungsi yang berada pada activity dapat langsung digunakan dalam fragment tersebut tanpa harus membuat ulang. Contoh pada saat **back**, fragment hanya perlu memanggil fungsi **getactivity**.



Siklus hidup (*life cycle*) dari Fragment dapat dilihat pada gambar berikut ini :



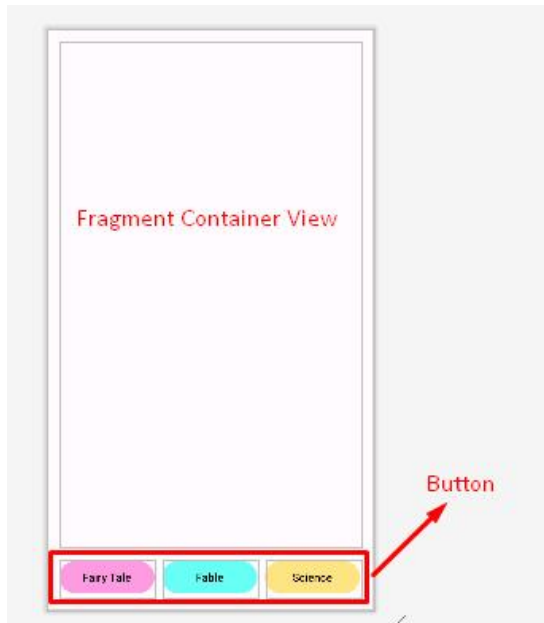
Gambar 1. Fragment Life Cycle

## IMPLEMENTASI FRAGMENT

Pada materi sebelumnya, jika kita ingin menampilkan activity maka hanya 1 activity saja yang dapat ditampilkan dalam layar tanpa dapat membaginya. Namun dengan adanya Fragment memungkinkan kita untuk dapat menampilkan beberapa bagian dalam 1 layar dan mengontrol bagian-bagian yang berbeda secara terpisah. Fragment memungkinkan kita dapat membentuk beberapa fragment dalam 1 activity karena Fragment memiliki event, layout, dan status mereka sendiri.

Langkah-langkah implementasi Fragment antara lain :

1. Buka kembali project yang telah anda buat pada pertemuan sebelumnya di Android Studio (**ProjectBPXXXX**).
2. Tambahkan 1 activity baru pada project anda, pilih **Empty Views Activity**, berikan nama **BookKidActivity**. (**Pastikan penulisan nama activity benar sesuai dengan gaya Camel Case**)
3. Buat desain layout seperti berikut ini :



XML code :

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BookKidActivity">

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragmentKidBook"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        app:layout_constraintBottom_toTopOf="@+id/buttonFable"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```

<Button
    android:id="@+id/buttonFairy"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginBottom="16dp"
    android:text="Fairy Tale"
    android:backgroundTint="#FF9BE3"
    android:textColor="@color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/buttonFable"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:text="Fable"
    android:backgroundTint="#68FFF6"
    android:textColor="@color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/buttonScience"
    app:layout_constraintStart_toEndOf="@+id/buttonFairy" />

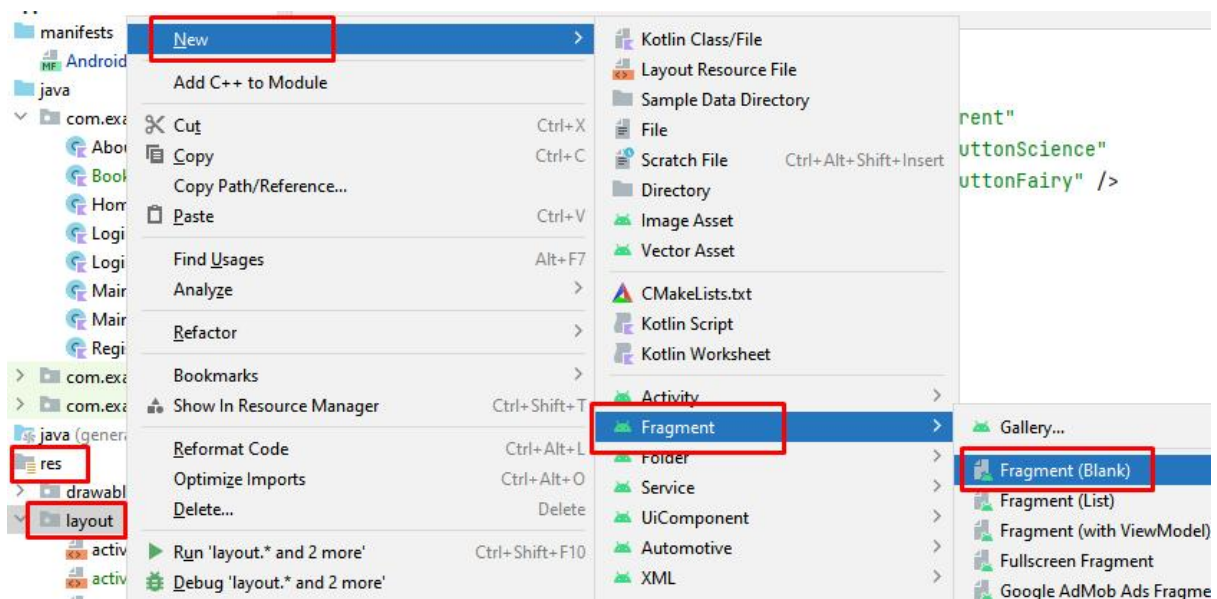
<Button
    android:id="@+id/buttonScience"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:text="Science"
    android:backgroundTint="#FFE484"
    android:textColor="@color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

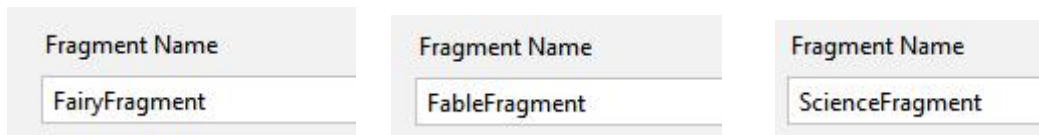
```

4. Langkah selanjutnya adalah membuat antar muka (UI) fragment untuk masing-masing button (Fairy Tale, Science dan Fable). Pada folder **res -> layout** silahkan buat fragment baru (pilih Blank agar fragment bisa didesain sendiri). Seperti pada contoh gambar di bawah ini :

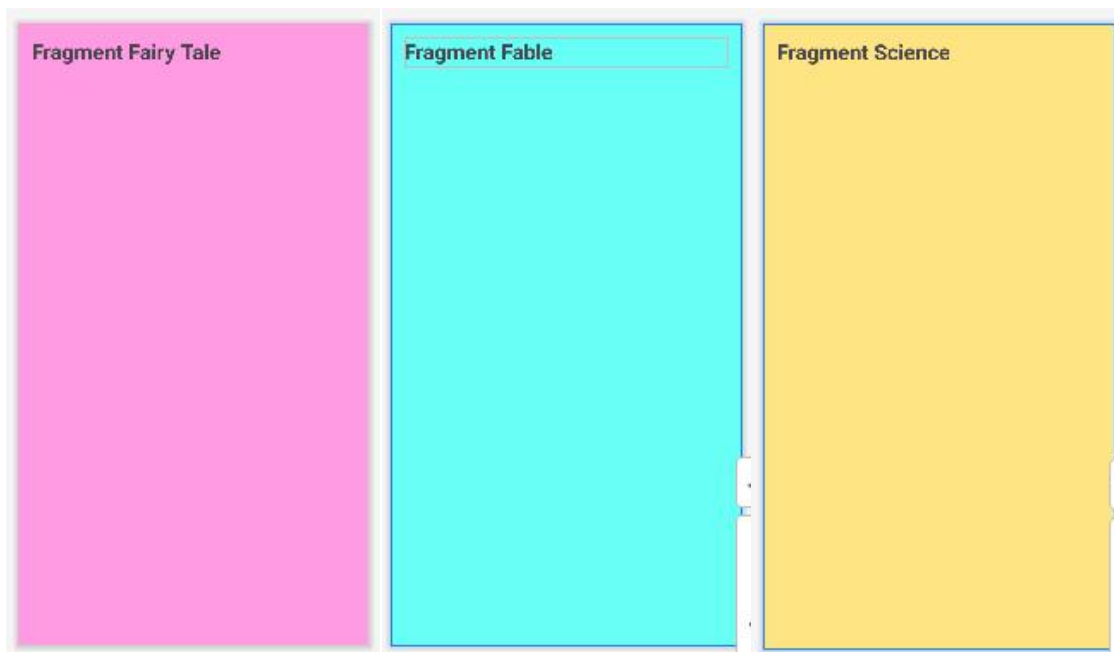




Buat 3 buah Blank Fragment : FairyFragment, FableFragment dan Science Fragment



5. Buka fairy\_fragment.xml, kemudian tambahkan sebuah TextView untuk memberikan keterangan bahwa Fragment tersebut adalah Fragment Fairy, dan lakukan hal yang sama untuk Fragment yang lain, sehingga contoh hasil desainnya dapat dilihat pada gambar dibawah ini :



6. Selanjutnya tambahkan source code pada **BookKidActivity.kt** untuk pemanggilan Fragment berdasarkan button yang di-klik. Langkah-langkahnya :

- a. Tambahkan deklarasi binding (binding adalah fitur yang memudahkan Anda menulis kode yang berinteraksi dengan tampilan) di bawah deklarasi class.

```
class BookKidActivity : AppCompatActivity() {
    lateinit var binding: ActivityBookKidBinding
    new *
```

- b. Buat fungsi `replaceFragment()` yang bertujuan untuk berganti fragment sesuai dengan button yang di-klik.

```
private fun replaceFragment(frg: Fragment) {
    val fragmentManager = supportFragmentManager
    val fragmentTrx = fragmentManager.beginTransaction()
    fragmentTrx.replace(R.id.fragmentKidBook, frg)
    fragmentTrx.commit()
}
```

- c. Bagian fungsi **onCreate()** tambahkan baris kode berikut ini :

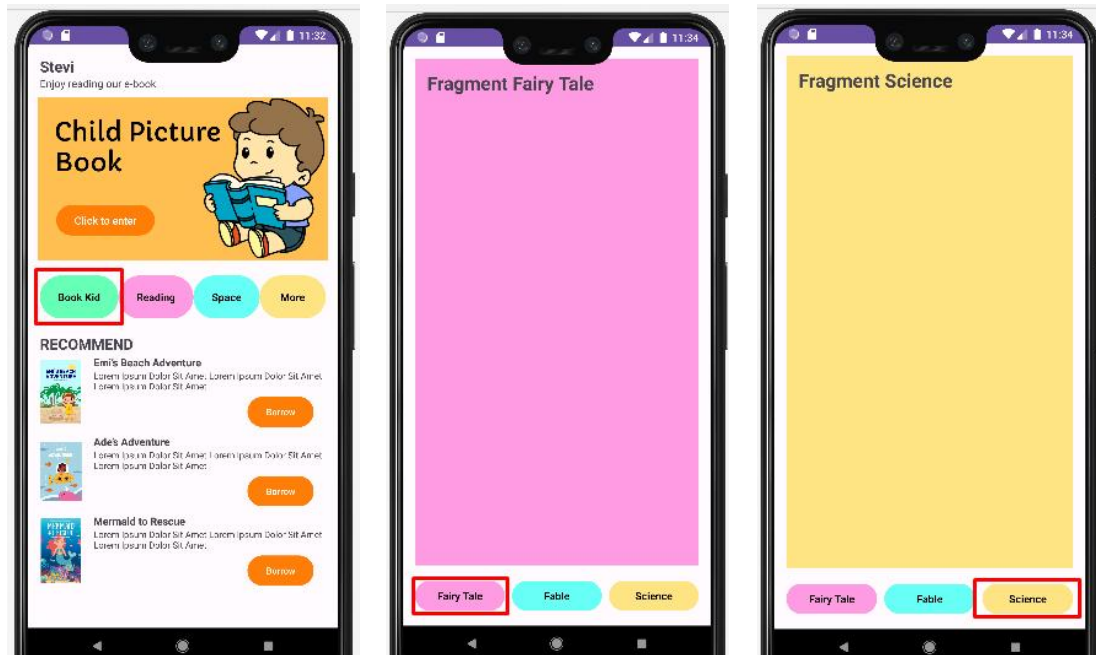
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityBookKidBinding.inflate(layoutInflater)
    setContentView(binding.root)

    //event saat button Fairy di klik
    binding.buttonFairy.setOnClickListener { it: View!
        replaceFragment(FairyFragment())
    }

    //event saat button Fable di klik
    binding.buttonFable.setOnClickListener { it: View!
        replaceFragment(FableFragment())
    }

    //event saat button Science di klik
    binding.buttonScience.setOnClickListener { it: View!
        replaceFragment(ScienceFragment())
    }
}
```

- d. Tambahkan intent untuk memanggil **BookKidActivity** pada bagian **HomeActivity** untuk menu button 1 (Book Kid).
- e. Pastikan semua baris kode yang dituliskan tidak ada kesalahan, dan jalankan aplikasi pada emulator. Jika semua source code-nya telah sesuai maka outputnya seperti berikut :



## TUGAS

Kerjakan tugas yang tertera di bawah ini :

1. Setelah mengimplementasi Fragment diatas, menurut anda apakah perlu Fragment ditambahkan dalam aplikasi yang rencananya akan anda buat ?
2. Berikan contoh lain berkaitan dengan penggunaan Fragment dalam aplikasi !