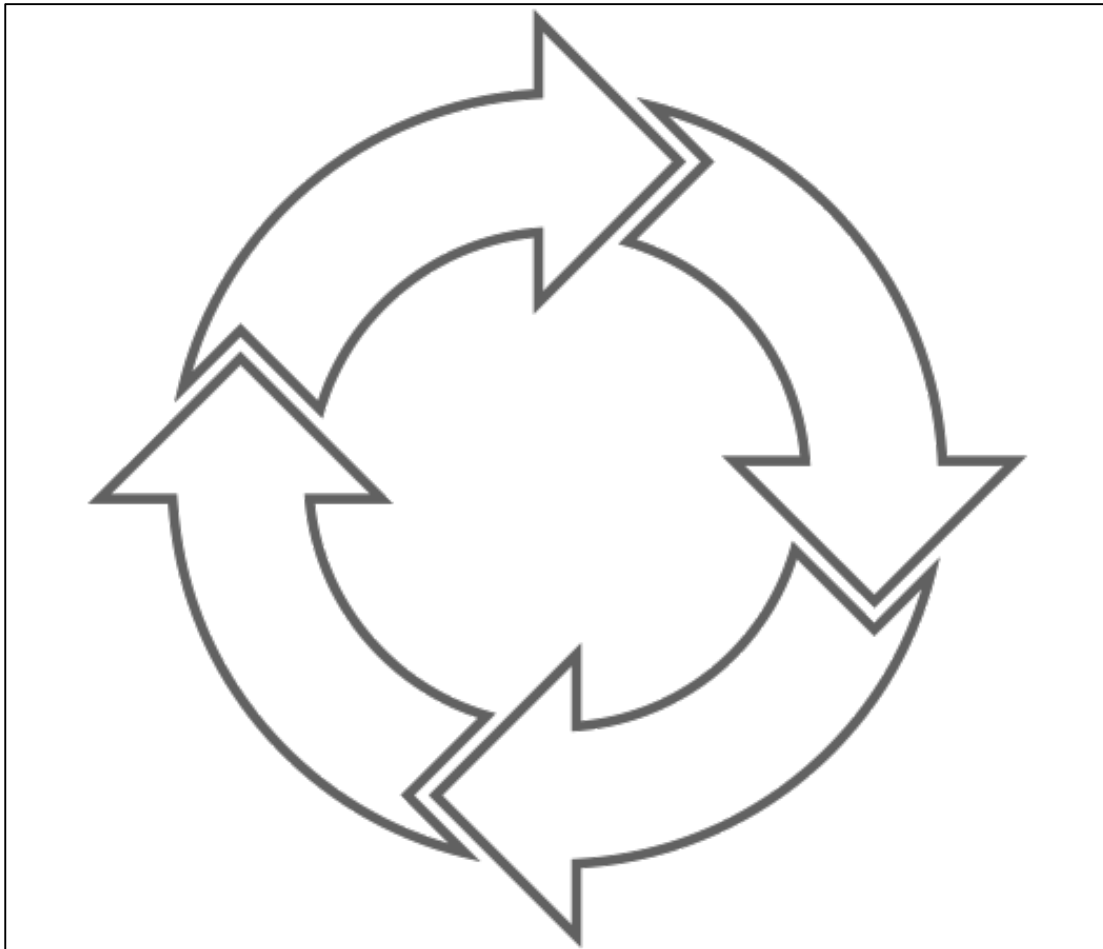


IS534 PEMROGRAMAN MOBILE

Praktikum 2

ACTIVITY LIFECYCLE DAN LOGCAT

Disusun Oleh: **Enrico Siswanto**

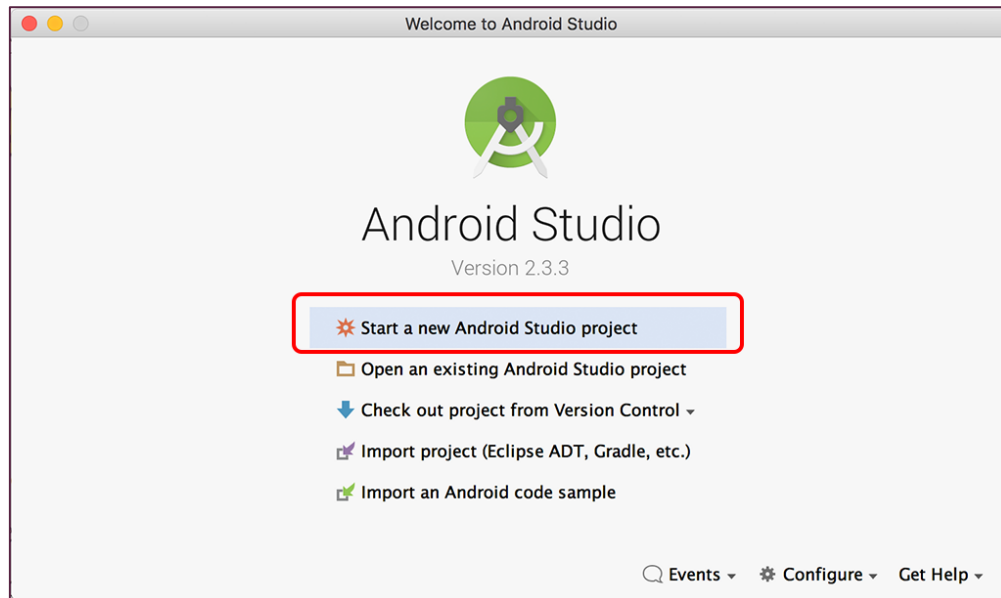


A. Tujuan Praktikum

1. Mengetahui activity lifecycle di dalam aplikasi Android yang dibuat
2. Mampu memanfaatkan logcat untuk mengidentifikasi activity lifecycle dan bug yang terjadi

B. Kegiatan Praktikum

1. Bukalah aplikasi Android Studio
2. Pilih Start a new Android Studio Project

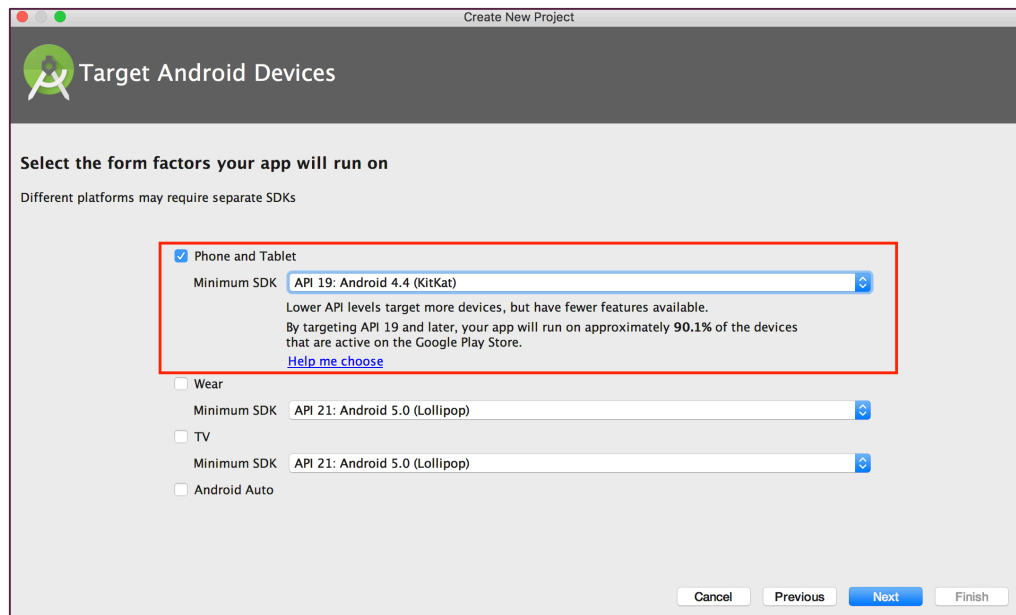


3. Kemudian pada dialog box yang muncul, masukkan informasi sebagai berikut:

- | | |
|---------------------|--|
| a. Application Name | : Lifecycle_Logcat |
| b. Company Domain | : Prak2_<NIM>.com |
| c. Project Location | : D:\Android_SI\<kode_kelas>\Prak2\<Application_Name> |

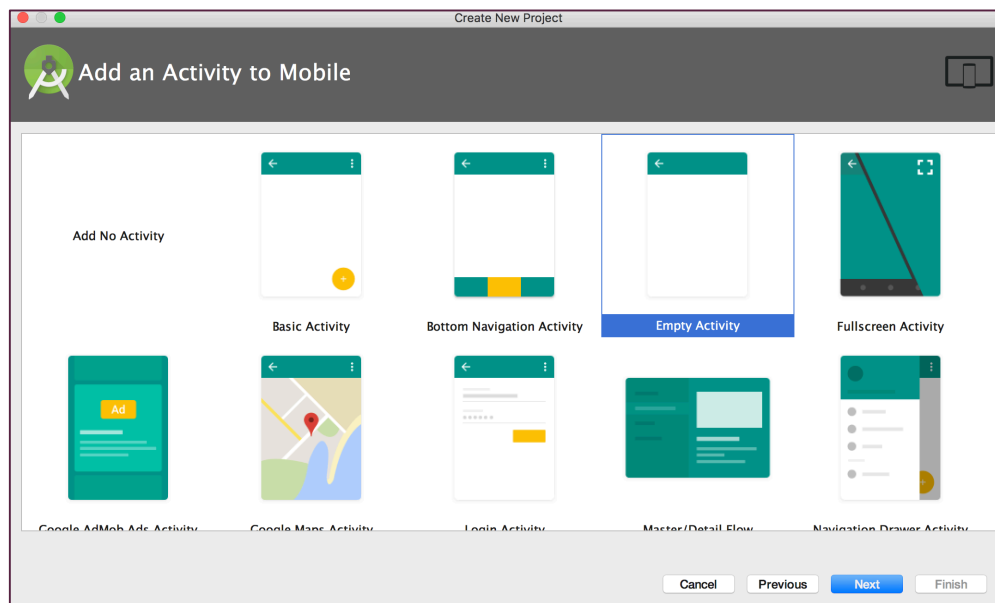
4. Klik Next

5. Pilih Phone and Tablet dan minimum SDK **API 19: Android 4.4 (KitKat)**



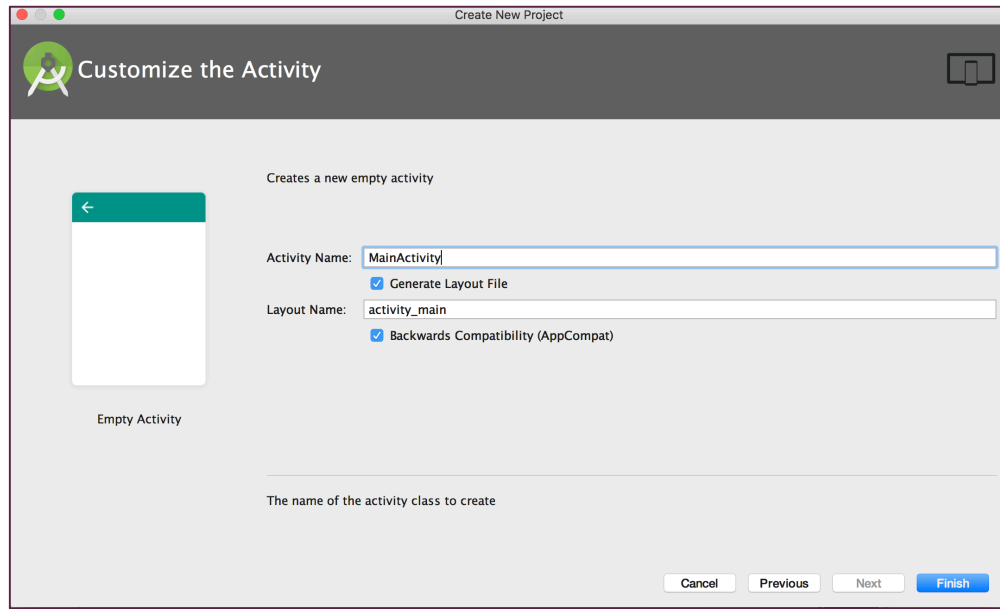
6. Klik Next

7. Akan muncul dialog box untuk memilih jenis activity yang akan dibuat. pilih **Empty Activity** dan klik Next



8. Kemudian pada dialog Customize Activity, beri nama Activity Name dengan nama **MainActivity** dan Layout Name dengan nama

activity_main



9. Klik Finish

10. Buka MainActivity.java, kemudian tambahkan

Log.i("posisi","onCreate") di dalam method onCreate dan lakukan import library Log seperti pada gambar.



11. Kemudian tambahkan method lainnya dengan menambahkan code berikut ini atau bisa digenerate otomatis dengan menekan Alt-Insert pada keyboard

```
@Override
protected void onStart() {
    super.onStart();
    Log.i("posisi", "onStart");
}

@Override
protected void onResume() {
    super.onResume();
    Log.i("posisi", "onResume");
}

@Override
protected void onPause() {
    super.onPause();
    Log.i("posisi", "onPause");
}

@Override
protected void onStop() {
    super.onStop();
    Log.i("posisi", "onStop");
}

@Override
protected void onRestart() {
    super.onRestart();
    Log.i("posisi", "onRestart");
}
```

```

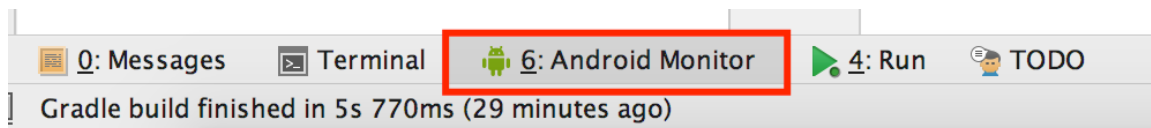
@Override
protected void onDestroy() {
    super.onDestroy();
    Log.i("posisi", "onDestroy");
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    Log.i("posisi", "onSaveInstanceState");
}

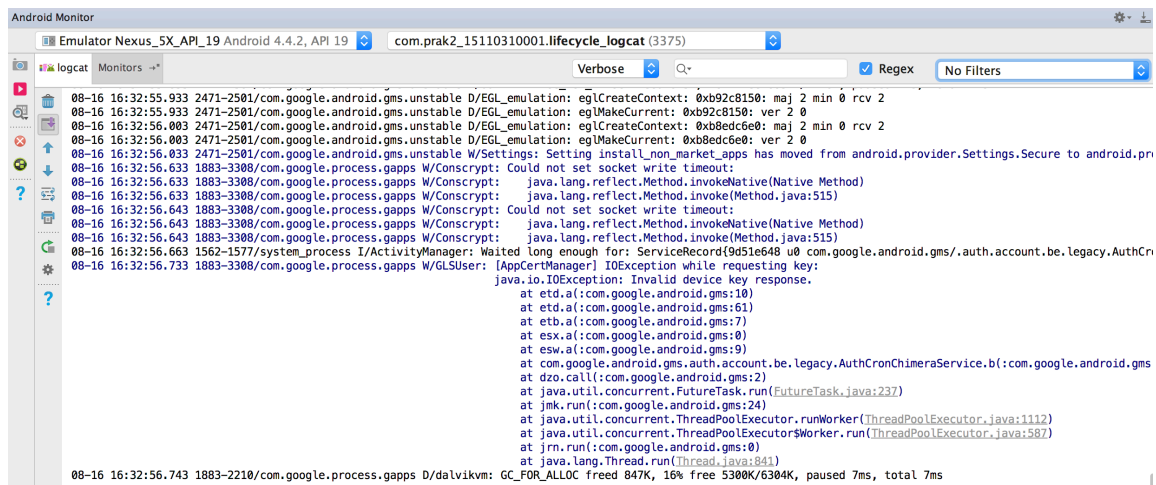
@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    Log.i("posisi", "onRestoreInstanceState");
}

```

12. Buka Window Logcat dengan cara mengklik Android Monitor pada bagian bawah aplikasi Android Studio

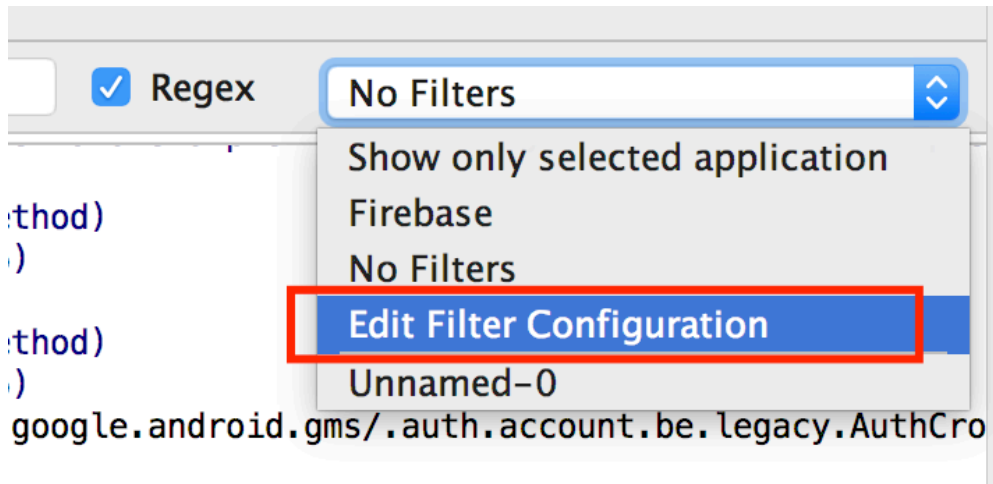


13. Coba jalankan aplikasi dengan mengklik tombol **Run**, dan secara default kita akan melihat semua informasi log dari aplikasi yang dijalankan secara realtime

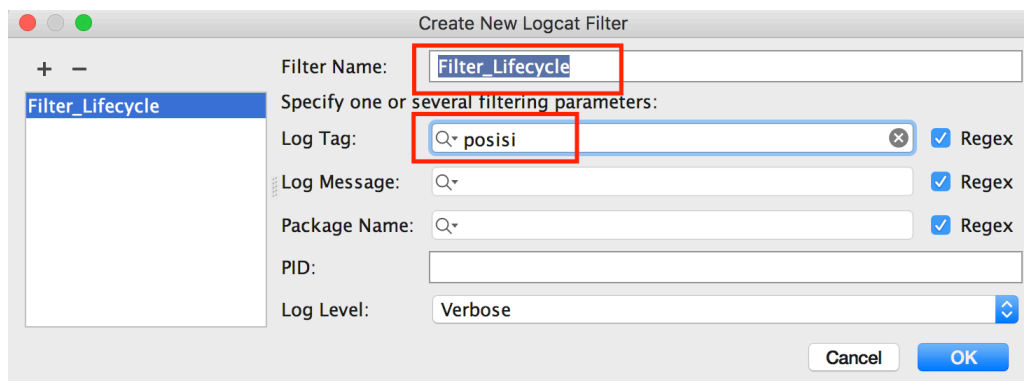


14. Untuk menampilkan hanya log yang kita butuhkan/buat, bisa dengan melakukan filter pada jendela Logcat

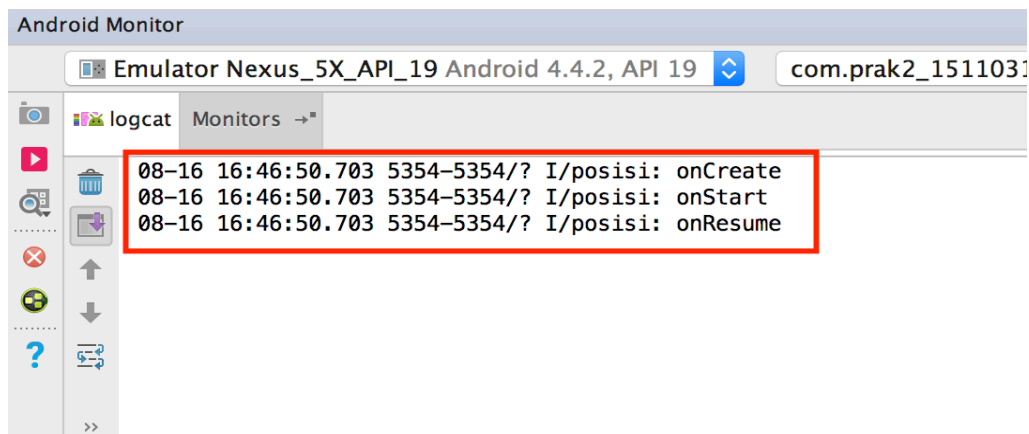
15. Pilih Dropdown list pada bagian kanan atas jendela Logcat, dan kemudian pilih **Edit Filter Configuration**



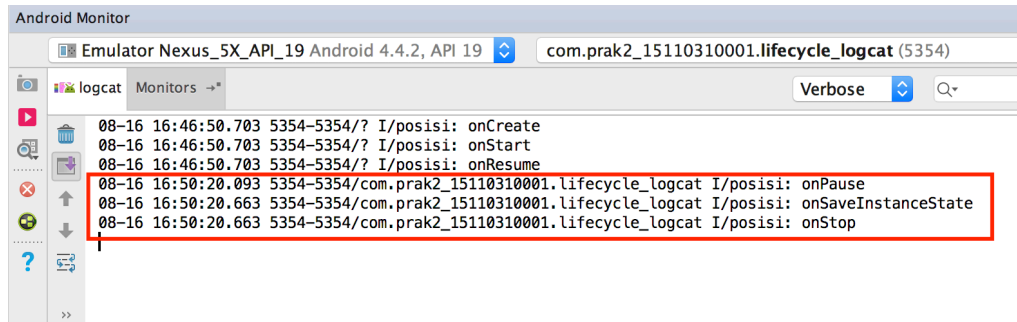
16. Buat filter baru dengan nama **Filter_Lifecycle**, kemudian kita filter log berdasarkan tag namanya dengan memberikan nama log tag yang sudah dibuat yaitu "**posisi**" dan berikan package name sesuai dengan package name aplikasi yang dibuat. Kemudian klik ok.



17. Jalankan kembali aplikasi yang dibuat, selanjutnya Logcat hanya akan menampilkan Log yang memiliki tag name "**posisi**"



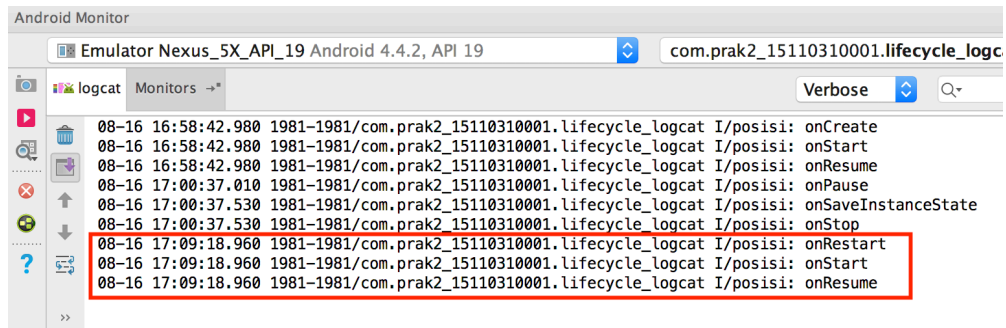
18. Kemudian lakukan eksperimen dengan menekan tombol Home pada device Android. Maka akan muncul status aplikasi tersebut sebagai berikut



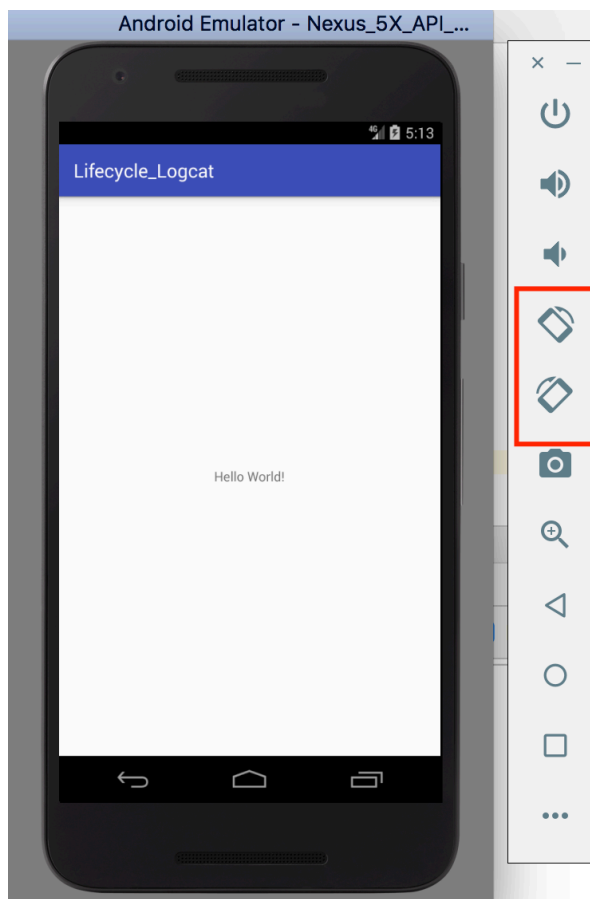
19. Tekan tombol **Overview/Recents Button** dan klik kembali aplikasi yang ada di list Overview



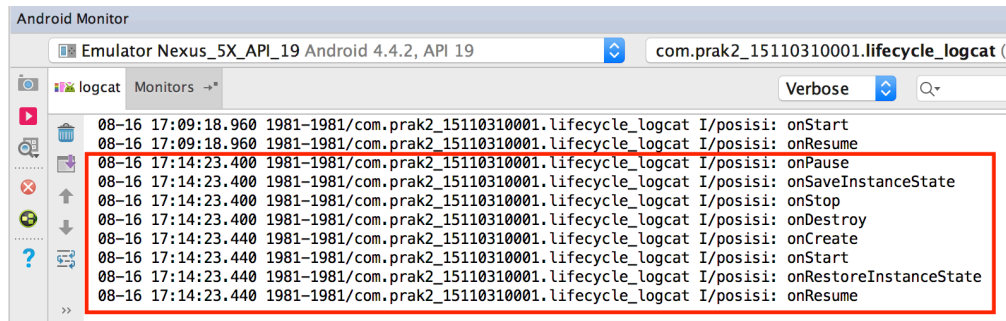
20. Cek kembali status jendela Logcat



21. Kemudian coba ubah orientasi layar dari Portrait ke Landscape dengan cara menekan tombol rotate right / left pada tombol di sebelah tampilan virtual device

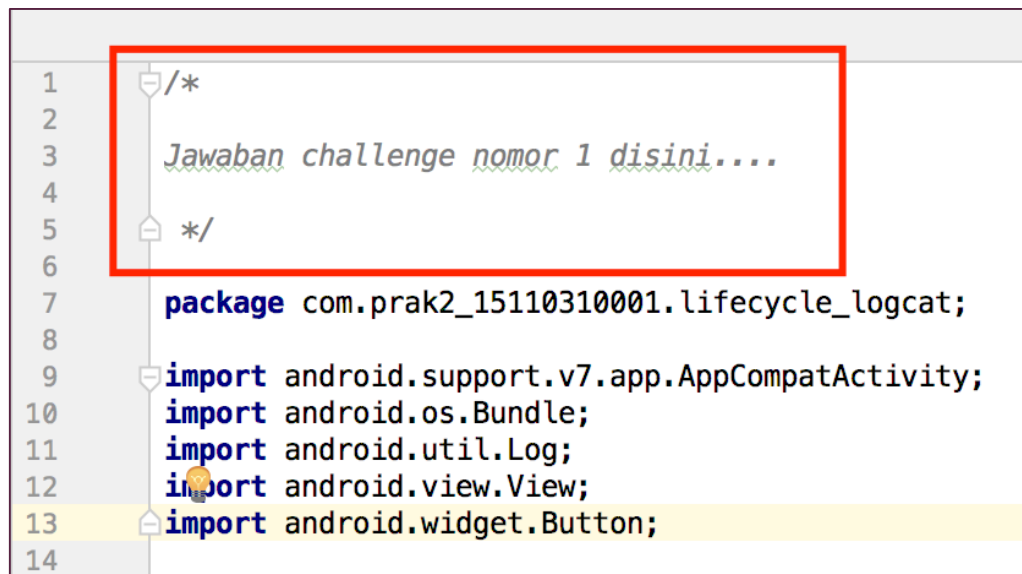


22. Cek kembali status jendela Logcat

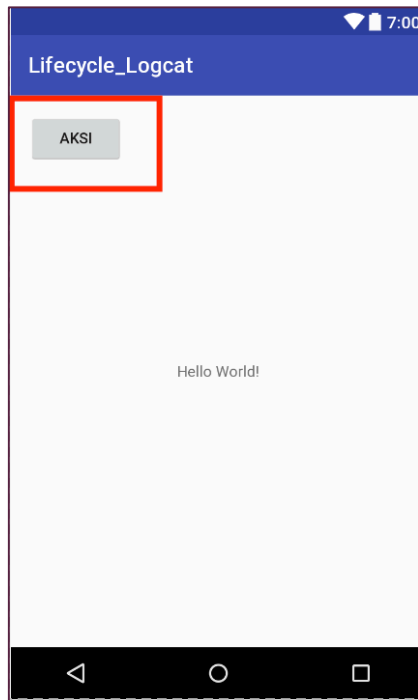


Challenge

1. Jelaskan mengapa activity lifecycle diatas dapat mengalami perubahan pada setiap action yang sudah dipraktikkan sebelumnya! Tuliskan jawaban Anda dengan memberikan **komentar** pada **MainActivity.java**!



2. Tambahkan satu button pada aplikasi Lifecycle_Logcat dan ganti ID button tersebut dengan **button1** dan beri nama button tersebut dengan **"Aksi"**



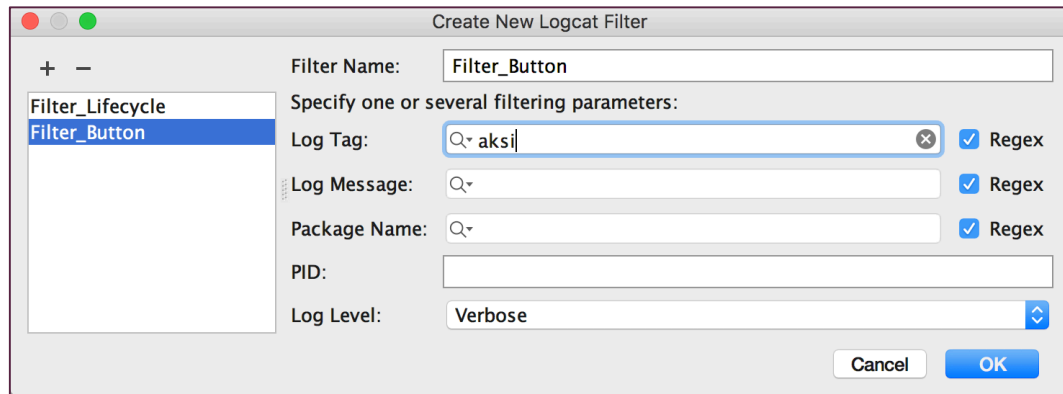
3. Masuk ke MainActivity.java dan tambahkan potongan code berikut di dalam method onCreate

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Log.i("posisi", "onCreate");

    Button tombol1 = (Button)findViewById(R.id.button1);
    tombol1.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v) {
            Log.i("aksi", "Tombol 1 diklik");
        }
    });
}
```

4. Buat filter baru di dalam jendela Logcat untuk menangkap Log yang dihasilkan jika button ditekan, beri nama filter tersebut dengan **"Filter_Button"**



5. Coba jalankan kembali aplikasi, dan kemudian tekan tombol Aksi dan lihat di dalam jendela Logcat. Apa log yang muncul? Copy hasil log yang muncul ke dalam komentar pada Challenge pertama