PBP

Modul 8

Hardware

PEMROGRAMAN BERBASIS PLATFORM

Oleh :

Cris Yustianto Putra Tangdialla

Program Studi Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

2022

Tujuan

1. Mengetahui hardware dan sensor apa saja yang dapat digunakan di Android Studio.
2. Menggunakan fitur hardware dan sensor yang ada pada device Android.

A. Camera

Popularitas kamera digital yang makin meningkat menyebabkan hampir semua ponsel cerdas memiliki kamera, tidak terkecuali dengan ponsel Android. Dengan Camera API yang disedikan oleh Android, pengguna dapat mengakses kamera dari perangkat ponsel untuk mengambil gambar. Android menyediakan kelas Camera sebagai API utama yang digunakan untuk mengontrol perangkat kamera. Kelas ini digunakan untuk mengambil gambar atau video ketika kita sedang membangun sebuah aplikasi kamera. Untuk meampilkan preview dari gambar yang ditangkap oleh kamera, Android menyediakan kelas SurfaceView. Kelasi ini digunakan untuk menampilkan apa yang dilihat kamera langsung kepada user.

Untuk dapat mengakses hardware kamera yang terpasang pada device, Anda perlu mendeklarasikan penggunaan fitur dan permintaan akses ke kamera, dengan menggunakan code <uses-feature android:name="android.hardware.camera" /> dan <uses-permission android:name="android.permission.CAMERA"/> kode ini diletakkan pada android manifest.

B. Sensor

Kebanyakan device Android memiliki sensor didalamnya yang mengukur gerakan, orientasi, dan berbagai kondisi lingkungan. Sensor-sensor ini dapat memberikan data mentah dengan tingkat presisi dan akurasi yang tinggi, dan sangat berguna jika ingin melakukan monitor terhadap pergerakan atau posisi device secara 3 dimensi, atau ingin mengukur perubahan lingkungan disekitar device.

Platform Android, mendukung 3 kategori sensor:

1. Motion sensors

Sensor ini mengukur gaya akselesasi dan gaya putar terhadap 3 sumbu. Kategori ini meliputi accelerometers, gravity sensors, gyroscopes,dan rotational vector sensors.

2. Environmental sensors

Sensor ini mengukur berbagai parameter lingkungan, seperti temperature dan tekanan udara, cahaya, dan kelembaban. Kategori ini meliputi barometers, photometers, dan thermometers.

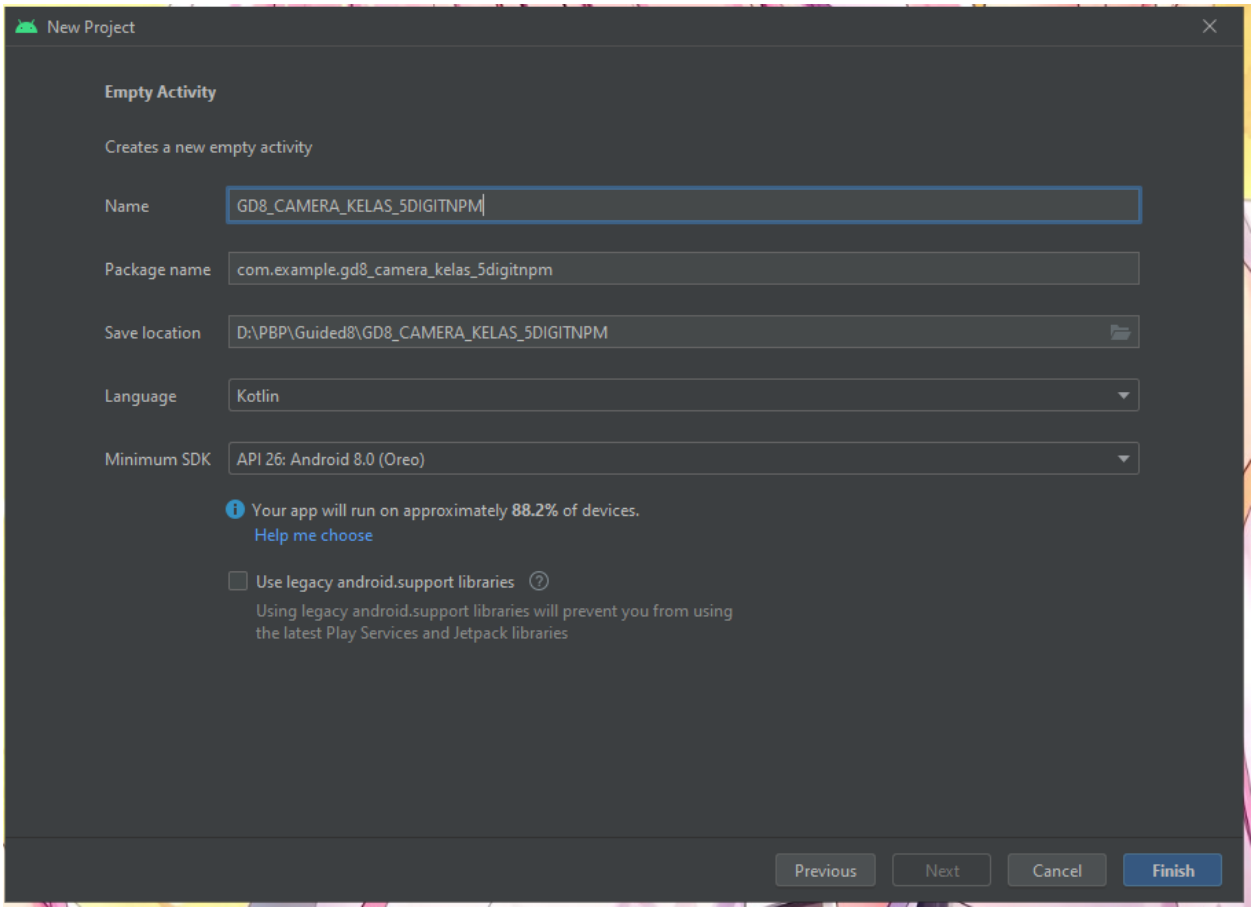
3. Position sensors

Sensor ini mengukur posisi fisik dari device. Kategori ini meliputi orientation sensors dan magnetometers.

Guided 1 – Camera

Pada guided 1 ini, kita akan mencoba mengakses hardware kamera yang terpasang dalam smartphone android anda dalam sebuah aplikasi android. Untuk melakukannya, silahkan ikuti langkah – langkah sebagai berikut:

1. Buatlah projek dengan ketentuan sebagai berikut:
   a. Application Name : GD8_Camera_KELAS_5DIGITNPM
   b. Minimum SDK: Android Oreo 26
   c. Jenis Activity: Empty Activity

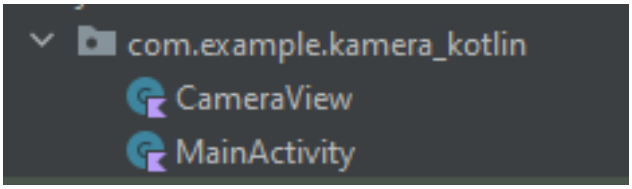


2. Kemudian buatlah tampilan layout activity_main.xml sebagai berikut:

   **buatlah icon_action_close pada drawable untuk menutup aplikasi anda.*

```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/FLCamera"/>

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imgClose"
        android:layout_gravity="center|bottom"
        android:background="@drawable/ic_baseline_close"/>

</FrameLayout>
```

3. Selanjutnya buatlah class baru dengan nama CameraView. Class ini berguna untuk menampilkan hasil capture kamera dan berguna untuk menjalankan hardware kamera.
4. Selanjutnya buka Class CameraView yang sudah dibuat dan modifikasi kodenya seperti



berikut :

```
class CameraView(context: Context?, private val mCamera: Camera) : SurfaceView(context),
    SurfaceHolder.Callback {
```

Class *CameraView* ini mengextends SurfaceView dan mengimplements interface SurfaceHolder.Callback.

5. Setelah itu, buatlah sebuah constructor class dan variable private seperti berikut :

```
    private val mHolder: SurfaceHolder

    init {
        mCamera.setDisplayOrientation(90)
        mHolder = holder
        mHolder.addCallback(this)
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS)
    }
```

6. Lalu implementasikan ketiga implement dari classs method tersebut :

```
override fun surfaceCreated(surfaceHolder: SurfaceHolder) {
    try {
        mCamera.setPreviewDisplay(mHolder)
        mCamera.startPreview()
    } catch (e: IOException) {
        Log.d( tag: "error",  msg: "Camera eror on SurfaceCreated" + e.message)
    }
}

override fun surfaceChanged(surfaceHolder: SurfaceHolder, i: Int, i1: Int, i2: Int) {
    if (mHolder.surface == null) return
    try {
        mCamera.setPreviewDisplay(mHolder)
        mCamera.startPreview()
    } catch (e: IOException) {
        Log.d( tag: "Error",  msg: "Camera error on SurfaceChanged" + e.message)
    }
}

override fun surfaceDestroyed(surfaceHolder: SurfaceHolder) {
    mCamera.stopPreview()
    mCamera.release()
}
```

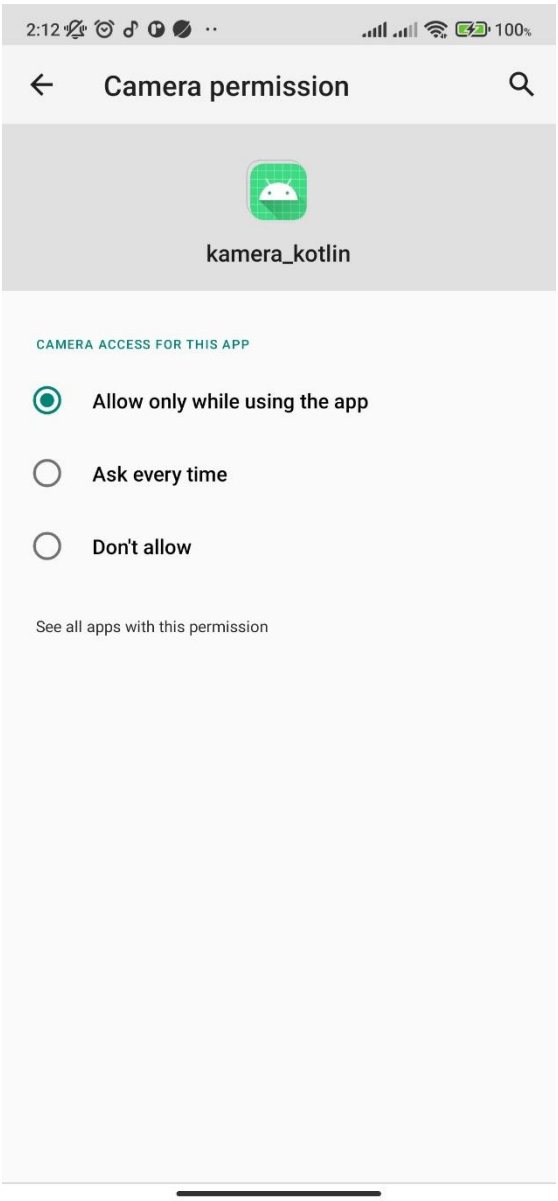
7. Selanjutnya pindah ke MainActivity.kt, tambahkan variable dan kemudian modifikasilah onCreate MainActivity.kt, seperti berikut :

```kotlin
package com.example.kamera_kotlin

import android.annotation.SuppressLint
import android.hardware.Camera
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.FrameLayout
import android.widget.ImageButton

class MainActivity : AppCompatActivity() {
    private var mCamera: Camera? = null
    private var mCameraView: CameraView? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        try {
            mCamera = Camera.open()
        } catch (e: Exception) {
            Log.d( tag: "Error",  msg: "Failed to get Camera" + e.message)
        }
        if (mCamera != null) {
            mCameraView = CameraView( context: this, mCamera!!)
            val camera_view = findViewById<View>(R.id.FLCamera) as FrameLayout
            camera_view.addView(mCameraView)
        }
        @SuppressLint("MissingInflatedId", "LocalSuppress") val imageClose =
            findViewById<View>(R.id.imgClose) as ImageButton
            imageClose.setOnClickListener { view: View? -> System.exit( status: 0) }
    }
}
```

8. Lalu tambahkan code permission di android manifest :

```
<uses-permission android:name="android.permission.CAMERA"/>
```

9. Setelah itu run aplikasi maka camera akan tampil dan jika icon X diklik maka camera akan ditutup. Gunakan Smartphone Android untuk mencobanya, bukan Emulator.



Guided 2 – Proximity

Pada guided 2 ini, kita akan mencoba mengakses Proximity yang terpasang dalam smartphone android anda dalam sebuah aplikasi android. Untuk melakukannya, silahkan ikuti langkah – langkah sebagai berikut:

1. Langkah awal buatlah projek dengan ketentuan seperti berikut :

a. Application Name : GD8_Proximity_KELAS_5 DIGITNPM

b. Minimum SDK : Android Oreo 26

c. Jenis Activity : Empty Activity

2. Kemudian buatlah tampilan layout activity_main.xml seperti berikut :

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!--on below line we are creating
        a text for heading of our app-->
    <TextView
        android:id="@+id/idTVHeading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:gravity="center"
        android:padding="4dp"
        android:text="QR Code Generator"
        android:textAlignment="center"
        android:textColor="@color/purple_200"
        android:textSize="18sp"
        android:textStyle="bold" />

    <!--Text view to display sensor status-->
    <TextView
        android:id="@+id/idTVSensorStatus"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:gravity="center"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="50dp" />

</RelativeLayout>
```

3. Selanjutnya pada class MainActivity.java, modifikasi code dan tambahkan 3 variable seperti berikut:

```
// on below line we are creating
// a variable for our text view.
lateinit var sensorStatusTV: TextView

// on below line we are creating
// a variable for our proximity sensor
lateinit var proximitySensor: Sensor

// on below line we are creating
// a variable for sensor manager
lateinit var sensorManager: SensorManager
```

4. Tambahkan method baru :

```
var proximitySensorEventListener: SensorEventListener? = object : SensorEventListener {
    override fun onAccuracyChanged(sensor: Sensor, accuracy: Int) {
        // method to check accuracy changed in sensor.
    }

    override fun onSensorChanged(event: SensorEvent) {
        // check if the sensor type is proximity sensor.

    }
}
```

5. Lalukan modifikasi seperti berikut :

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // on below line we are initializing our all variables.
    sensorStatusTV = findViewById(R.id.idTVSensorStatus)

    // on below line we are initializing our sensor manager
    sensorManager = getSystemService(Context.SENSOR_SERVICE) as SensorManager

    // on below line we are initializing our proximity sensor variable
    proximitySensor = sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY)

    // on below line we are checking if the proximity sensor is null
    if (proximitySensor == null) {
        // on below line we are displaying a toast if no sensor is available
        Toast.makeText(this, "No proximity sensor found in device..", Toast.LENGTH_SHORT).show()
        finish()
    } else {
        // on below line we are registering
        // our sensor with sensor manager
        sensorManager.registerListener(
```

```
            proximitySensorEventListener,
            proximitySensor,
            SensorManager.SENSOR_DELAY_NORMAL
        )
    }
}
```

6. Lalu tambahkan code untuk atur near dan far benda pada proximity

```
if (event.sensor.type == Sensor.TYPE_PROXIMITY) {
    if (event.values[0] == 0f) {
        // here we are setting our status to our textview..
        // if sensor event return 0 then object is closed
        // to sensor else object is away from sensor.
        sensorStatusTV.text = "<<<Near>>>"
    } else {
        // on below line we are setting text for text view
        // as object is away from sensor.
        sensorStatusTV.text = "<<<<Away>>>>"
    }
}
```

7. Jangan lupa tambahkan code ini di AndroidManifest kalian :

```
<uses-permission android:name="android.hardware.sensor.proximity"/>
```

8. Hasilnya seperti berikut : (dekatkan telapak tangan kalian ke *earpiece* untuk telepon)

<<<<Away>>>>

<<<Near>>>

Guided 3 – Accelerometer

Pada guided 3 ini, kita akan mencoba mengakses Accelerometer yang terpasang dalam smartphone android anda dalam sebuah aplikasi android. Untuk melakukannya, silahkan ikuti langkah – langkah sebagai berikut:

1. Langkah awal buatlah projek dengan ketentuan seperti berikut :
   a. Application Name : GD8_Proximity_KELAS_5 DIGITNPM
   b. Minimum SDK : Android Oreo 26
   c. Jenis Activity : Empty Activity
2. Tambahkan pada main_activity.xml seperti ini :

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    android:background="#212121"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv_square"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerInParent="true"
        android:background="#EF5350"
        android:gravity="center"
        android:text="Hello"
        android:textColor="@color/white"
        android:textSize="20sp" />


</RelativeLayout>
```

3. Pada Accelerometer, tambahkan variable private seperti berikut :

```
class MainActivity : AppCompatActivity(), SensorEventListener {

    private lateinit var sensorManager: SensorManager
    private lateinit var square: TextView
```

4. Pada xml yang sudah kita buat, kotak yang dibuat diharapkan bisa digerakkan dengan leluasa 360 derajat sesuai dengan arah sumbu pengarah dari sensor tersebut. Lakukan inisialisasi terlebih dahulu dengan membuat function setUpSensorStuff()

```
// Keeps phone in light mode
AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO)

square = findViewById(R.id.tv_square)

setUpSensorStuff()
```

5. Isilah code berikut pada function method setUpSensorStuff()

```
sensorManager = getSystemService(SENSOR_SERVICE) as SensorManager

// Specify the sensor you want to listen to
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)?.also
{ accelerometer ->
    sensorManager.registerListener(
        this,
        accelerometer,
        SensorManager.SENSOR_DELAY_FASTEST,
        SensorManager.SENSOR_DELAY_FASTEST
    )
}
```

6. Buatlah method function onSensorChanged() dan onAccuracyChanged()

```
override fun onSensorChanged(event: SensorEvent?) {
```

```
override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {
    return
}
```

7. pada onSensorChanged() sebelumnya, kita sudah memanggil sensor Accelerometer dengan tipe kecepatan sensor yang dibacanya yaitu delay cepat (0.5 seconds). Kemudian, accelerometer akan menggukan konsep 3 sumbu kutub yang diukur, yaitu sumbu x, y dan z. tapi karena kita menggunakan 2 sumbu saja yang akan pakai, yaitu x dan y jadi isilah code yang diisi di method onSensorChanged() seperti berikut :

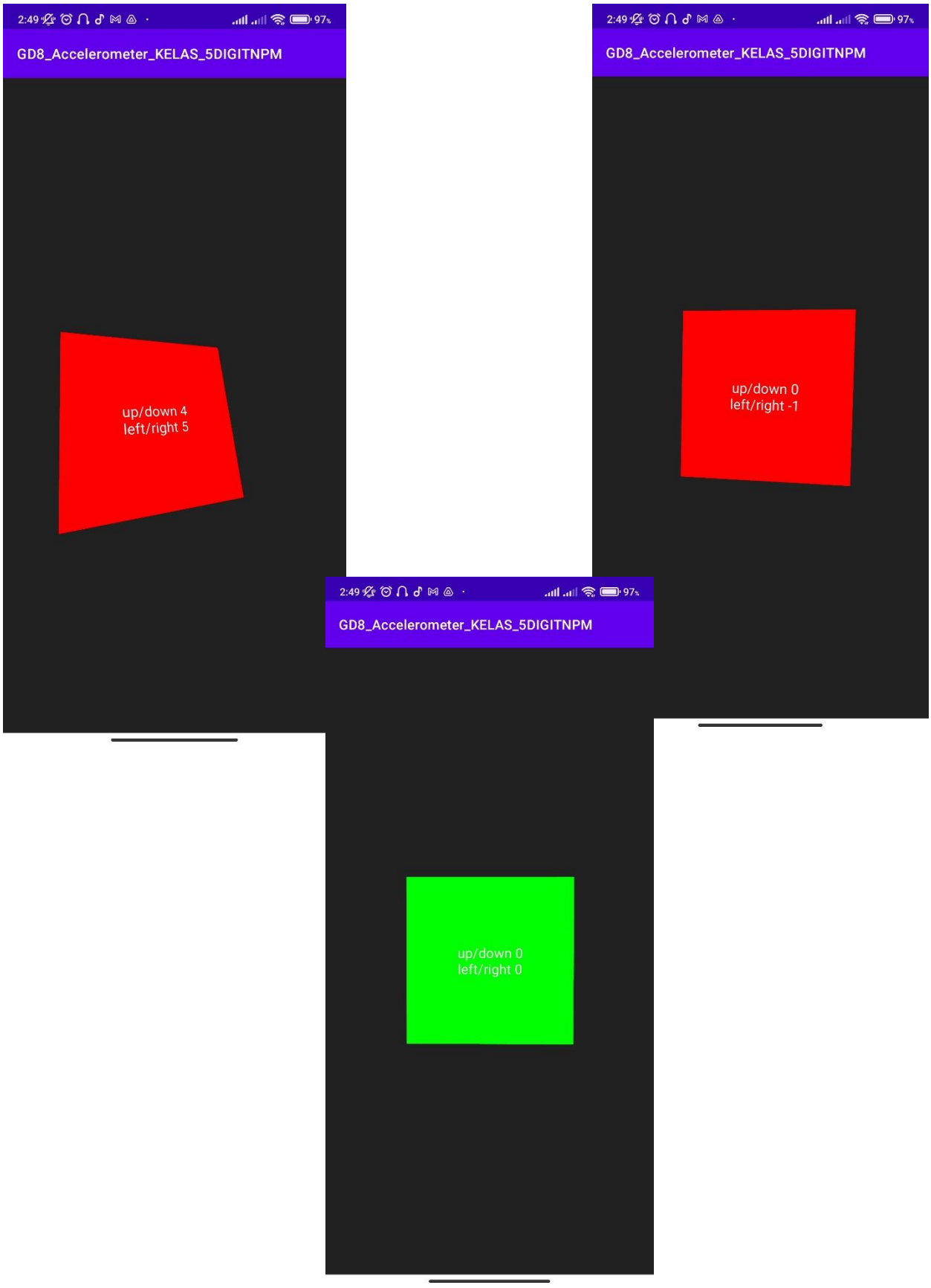
```
override fun onSensorChanged(event: SensorEvent?) {
    // Checks for the sensor we have registered
    if (event?.sensor?.type == Sensor.TYPE_ACCELEROMETER) {
        //Log.d("Main", "onSensorChanged: sides ${event.values[0]}
front/back ${event.values[1]} ")

        // Sides = Tilting phone left(10) and right(-10)
        val sides = event.values[0]

        // Up/Down = Tilting phone up(10), flat (0), upside-down(-
10)
```

```
        val upDown = event.values[1]

        square.apply {
            rotationX = upDown * 3f
            rotationY = sides * 3f
            rotation = -sides
            translationX = sides * -10
            translationY = upDown * 10
        }

        // Changes the colour of the square if it's completely flat
        val color = if (upDown.toInt() == 0 && sides.toInt() == 0)
Color.GREEN else Color.RED
        square.setBackgroundColor(color)

        square.text = "up/down ${upDown.toInt()}\nleft/right
${sides.toInt()}"
    }
```

8. Kemudian buatlah method function onDestroy() seperti berikut :

```
override fun onDestroy() {
    sensorManager.unregisterListener(this)
    super.onDestroy()
}
```

9. Hasilnya seperti berikut : (kalian goyangkan hp kalian bukan emulator)


2:49
97%
GD8_Accelerometer_KELAS_5DIGITNPM
up/down 4
left/right 5
2:49
97%
GD8_Accelerometer_KELAS_5DIGITNPM
up/down 0
left/right -1
2:49
97%
GD8_Accelerometer_KELAS_5DIGITNPM
up/down 0
left/right 0