

Tedie Muhammad Hanif

00000033597

Mobile Application Programming

2. Lakukan pengujian terhadap app anda dengan skenario di bawah ini:
 - a) Kosongkan Angka 1 dan Angka 2, lalu klik salah satu button
⇒ Terjadi force close app, dikarenakan kita harus memasukan angka
 - b) Kosongkan Angka 1 dan Angka 2, lalu klik salah satu button
⇒ Kasus masih sama kita harus memasukan angka di kedua slotnya
 - c) Isi 0 (nol) pada Angka 2, lalu klik button Bagi.
⇒ Infinity, karena nilai berapapun jika dibagi dengan angka nol (0) maka memiliki hasil infinity atau tak terhingga
3. Lakukan debugging pada program anda dengan meletakkan dua breakpoint bebas letaknya. Gunakan langkah-langkah step-over, step-into, step-out, dan resume. Tuliskan kesimpulan hasil debugging anda dan submit ke isian assignment di eLearning
 - a) Breakpoint

```
protected void onCreate(Bundle savedInstanceState) { savedInstanceState: null
    super.onCreate(savedInstanceState); savedInstanceState: null
    setContentView(R.layout.activity_main);
    angka1 = (EditText) this.findViewById(R.id.angka1); angka1: "androidx.appcompat.widget.AppCompatEditText{a413580 VFED..CL. ....I. 0,0-0,0 #7f08004b app:id/
    angka2 = (EditText) this.findViewById(R.id.angka2); angka2: "androidx.appcompat.widget.AppCompatEditText{d10f7b9 VFED..CL. ....I. 0,0-0,0 #7f08004c app:id/
    hasil = (TextView) this.findViewById(R.id.hasil); hasil: "com.google.android.material.textview.MaterialTextView{bc05675 V.ED.....ID 0,0-0,0 #7f0800b2
    btnTambah = (Button) this.findViewById(R.id.btnTambah);
    btnKurang = (Button) this.findViewById(R.id.btnKurang);
    btnKali = (Button) this.findViewById(R.id.btnKali);
    btnBagi = (Button) this.findViewById(R.id.btnBagi);

    btnTambah.setOnClickListener (new View.OnClickListener() { btnTambah: "com.google.android.material.button.MaterialButton{f4429ac VFED..C... ....I. 0,0-0,0 #
        @Override
        public void onClick(View v) { hitung ( operator: '+'); } });
    btnKurang.setOnClickListener (new View.OnClickListener() { btnKurang: "com.google.android.material.button.MaterialButton{9e8fb5f VFED..C... ....I. 0,0-0,0 #
        @Override
        public void onClick(View v) { hitung ( operator: '-'); } });

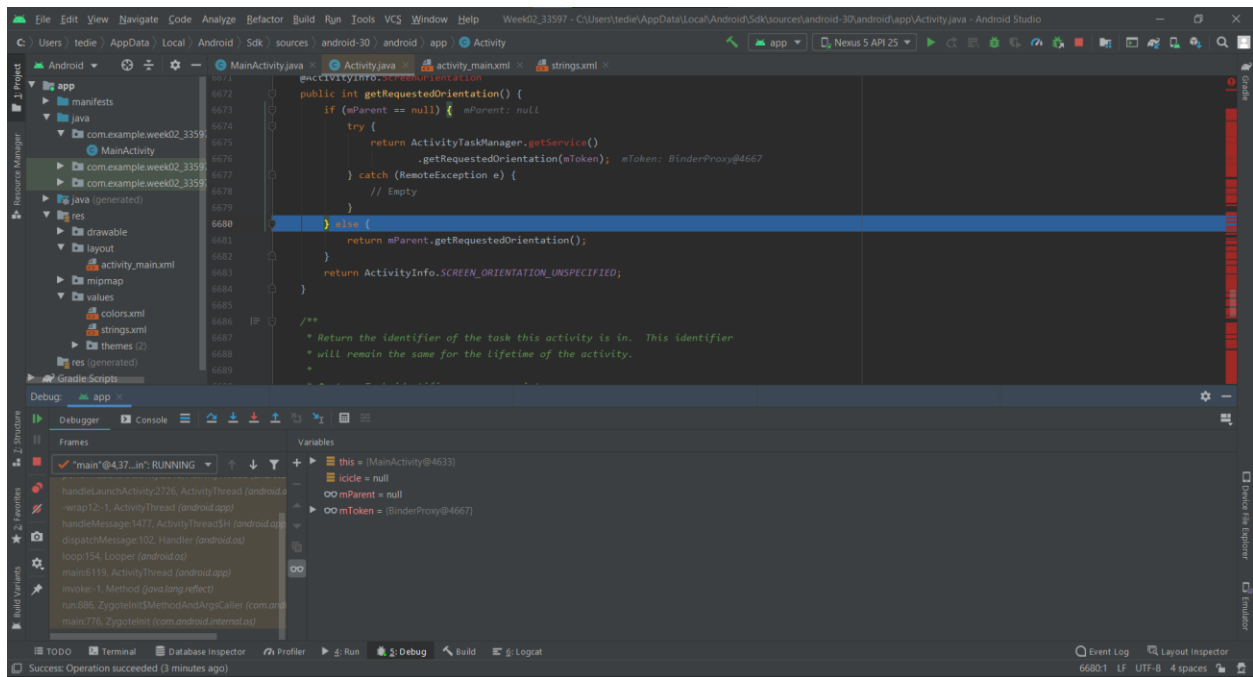
    btnKali.setOnClickListener (new View.OnClickListener() { btnKali: "com.google.android.material.button.MaterialButton{dca07fe VFED..C... ....I. 0,0-0,0 #7f08
        @Override
        public void onClick(View v) { hitung ( operator: '*'); } });

    btnBagi.setOnClickListener (new View.OnClickListener() { btnBagi: "com.google.android.material.button.MaterialButton{39b72b2 VFED..C... ....I. 0,0-0,0 #7f08
        @Override
        public void onClick(View v) { hitung ( operator: '/'); } });
    }

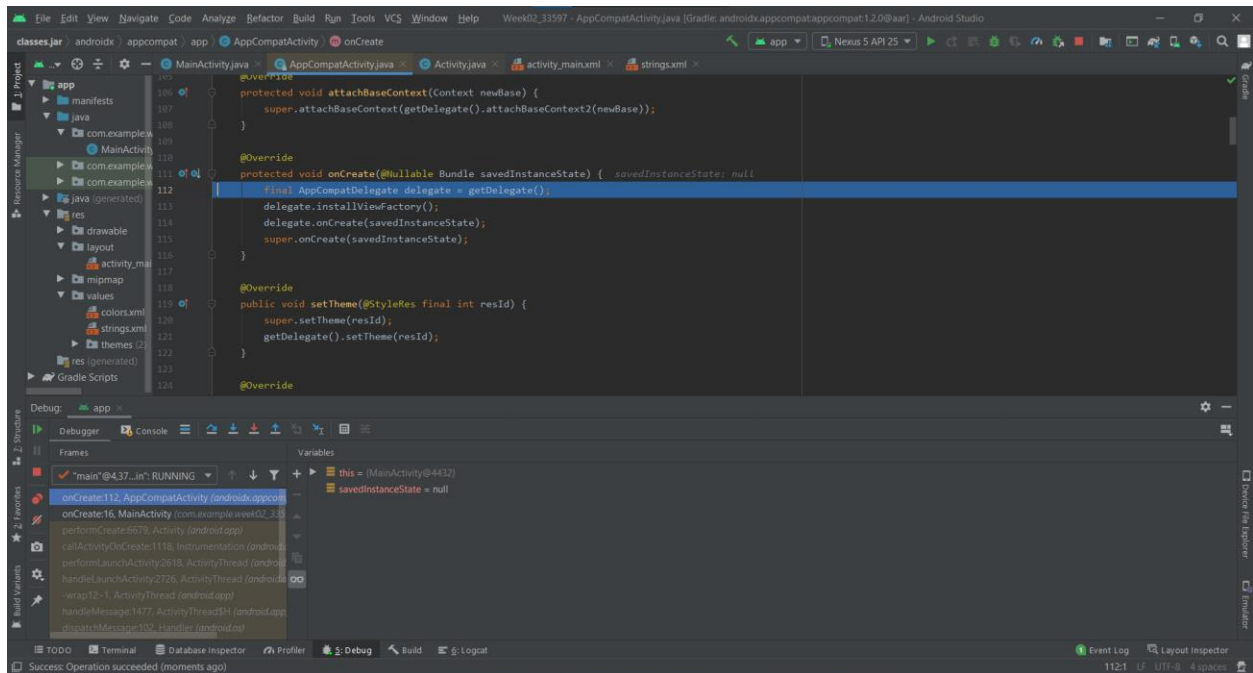
    protected void hitung(char operator){
        double operand1 = Double.parseDouble(angka1.getText().toString());
```

Breakpoint adalah tempat dalam kode yang Anda inginkan untuk menghentikan sementara eksekusi normal aplikasi untuk melakukan tindakan lainnya seperti memeriksa variabel-variabel atau mengevaluasi ekspresi, atau mengeksekusi kode setiap baris untuk menentukan penyebab kesalahan waktu proses.

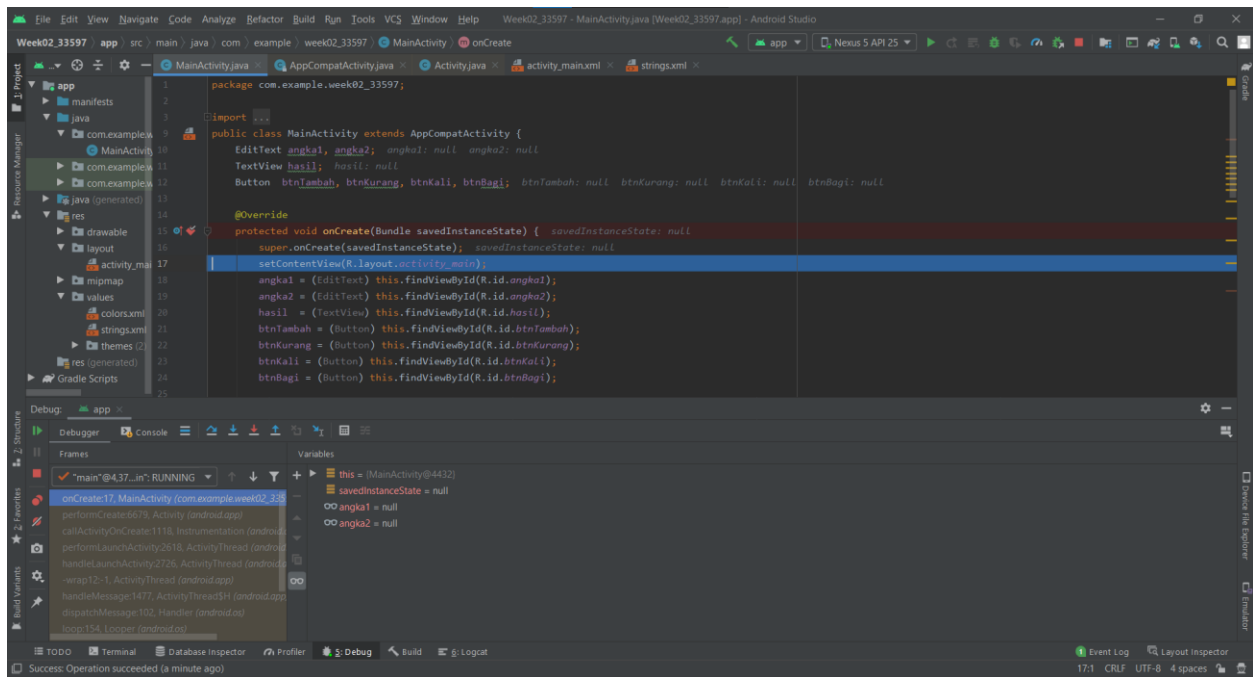
b) Step Over



c) Step Into



d) Step Out



Kesimpulan : Tumpukan eksekusi menampilkan semua kelas dan metode (bingkai) yang sedang dieksekusi hingga titik ini dalam aplikasi, dalam urutan terbalik (bingkai terbaru lebih dahulu). Saat eksekusi bingkai tertentu selesai, bingkai tersebut akan dimunculkan dari tumpukan dan eksekusi kembali ke bingkai berikutnya.