# Materi 4: Testing dan Debugging

## Pokok Bahasan

1. Debugging pada Android Studio
2. Melakukan Unit Testing pada Aplikasi

## Tujuan

Di Akhir pertemuan ini, diharapkan mahasiswa dapat mengenal cara melakukan debugging dan testing menggunakan JUnit pada saat membuat aplikasi Android.

## Penjelasan Singkat

Sebelum melangkah pada praktikum dimana anda akan mengerjakan modul-modul pada pertemuan ini. Anda perlu mengetahui dulu definisi dari Bugs, menurut pengertian yang diambil dari Wikipedia, pengertian Bug adalah kesalahan dalam program atau sistem komputer yang menyebabkan program menghasilkan hasil yang salah atau tidak terduga, atau berperilaku yang tidak diinginkan oleh pengguna. Bug ini dapat berupa:

1. Crashes
2. Exceptions
3. Freezes
4. Memory leaks

Hal ini dapat disebabkan karena beberapa hal, diantaranya sebagai berikut:

1. Desain dari sistem/ implementasi yang kurang tepat
2. Software fault
3. Hardware fault/limitation

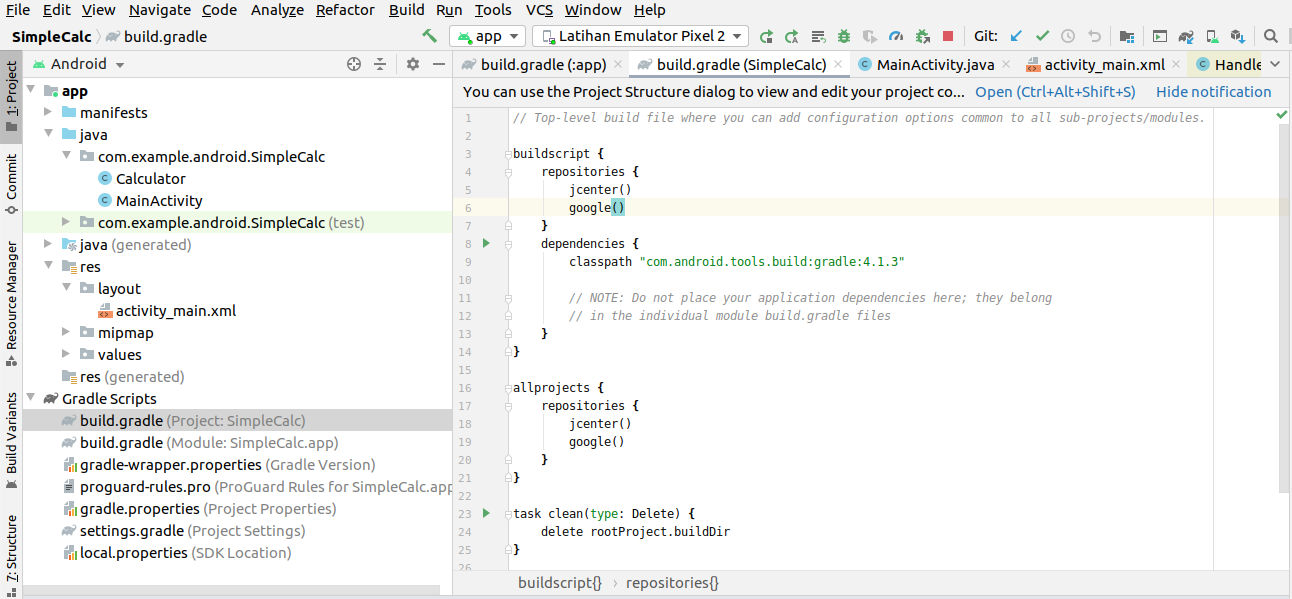
Debug merupakan sebuah cara yang digunakan untuk mencari dan memperbaiki error yang tidak diharapkan muncul pada aplikasi/ sistem, termasuk perilaku yang tidak diinginkan dari sebuah aplikasi. Pada Android Studio sudah terdapat tools untuk debugging aplikasi dimana mempunyai tujuan untuk:

1. Mengidentifikasi problem
2. Mencari dimana problem tersebut pertama kali muncul dan kemudian error tersebut dapat diperbaiki

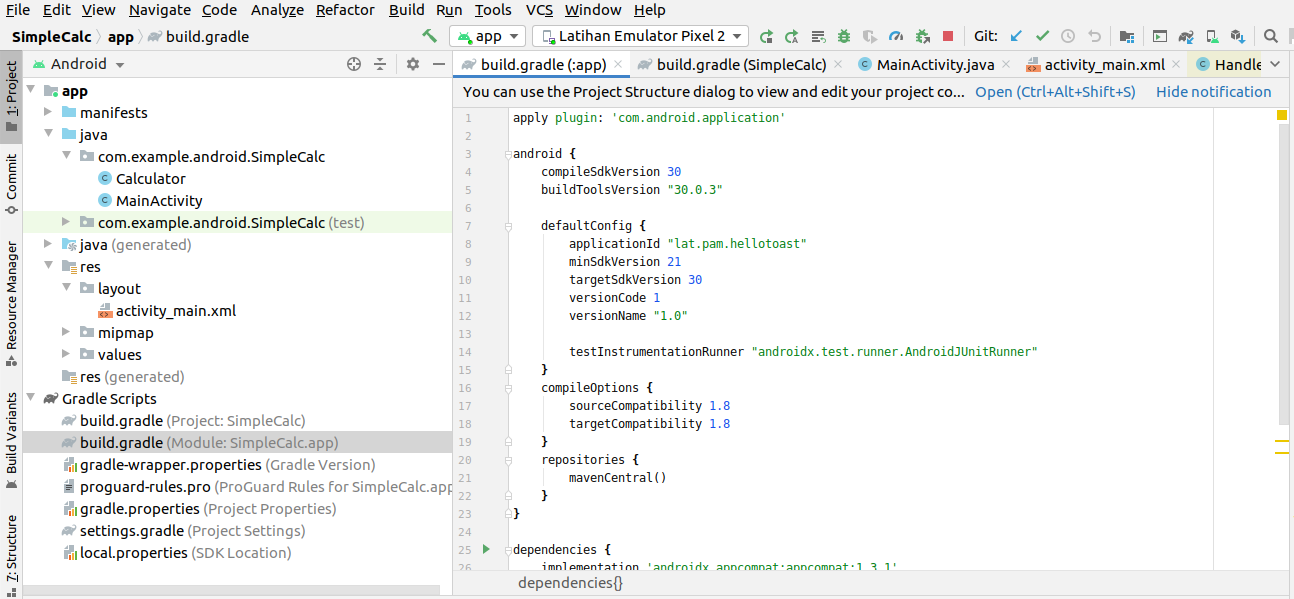
## Modul 1: Melakukan Debugging pada Android Studio

Pada modul pertama ini anda akan melakukan debugging pada Android Studio dengan menggunakan project Calculator yang sudah ada. Anda dapat mengunduh project pada tautan berikut ini [[link]](https://github.com/google-developer-training/android-fundamentals-starter-apps-v2/tree/master/SimpleCalc).

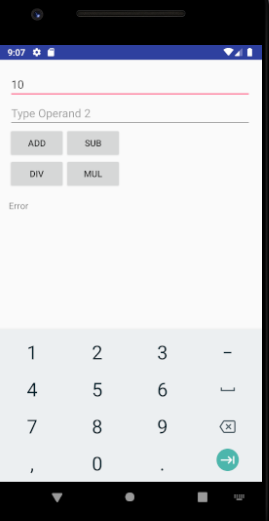
1. Pada tautan yang sudah disertakan tadi, anda dapat mengunduh project tersebut kemudian bukalah project SimpleCalc.
2. Anda dapat menyesuaikan project tersebut dengan melakukan setting versi Gradle terelebih dahulu pada file buld.gradle (Project: SimpleCalc), seperti pada tampilan berikut.



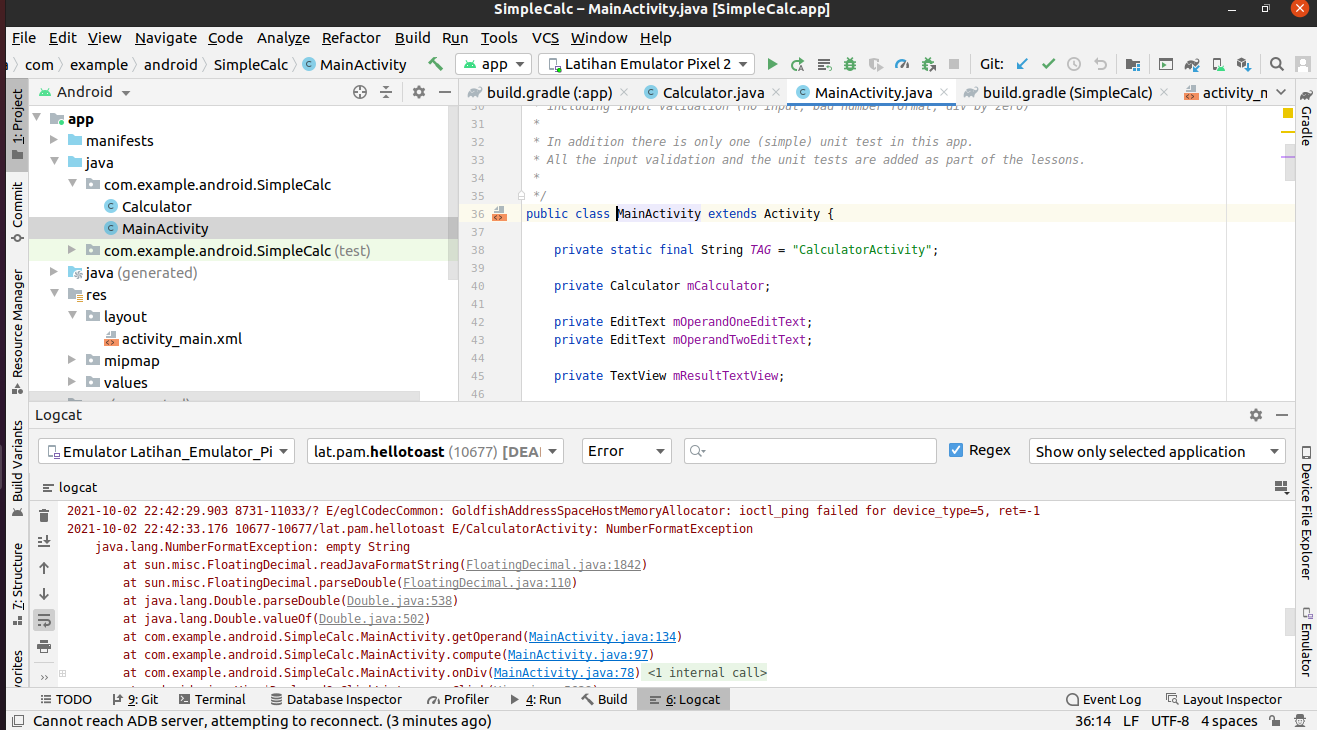
1. Selanjutnya, anda perlu menyesuaikan versi SDK, maka yang dapat dilakukan adalah melakukan setting pada compile dan target SDK. Sesuaikan dengan versi SDK yang digunakan pada Android Studio pada tampilan berikut.



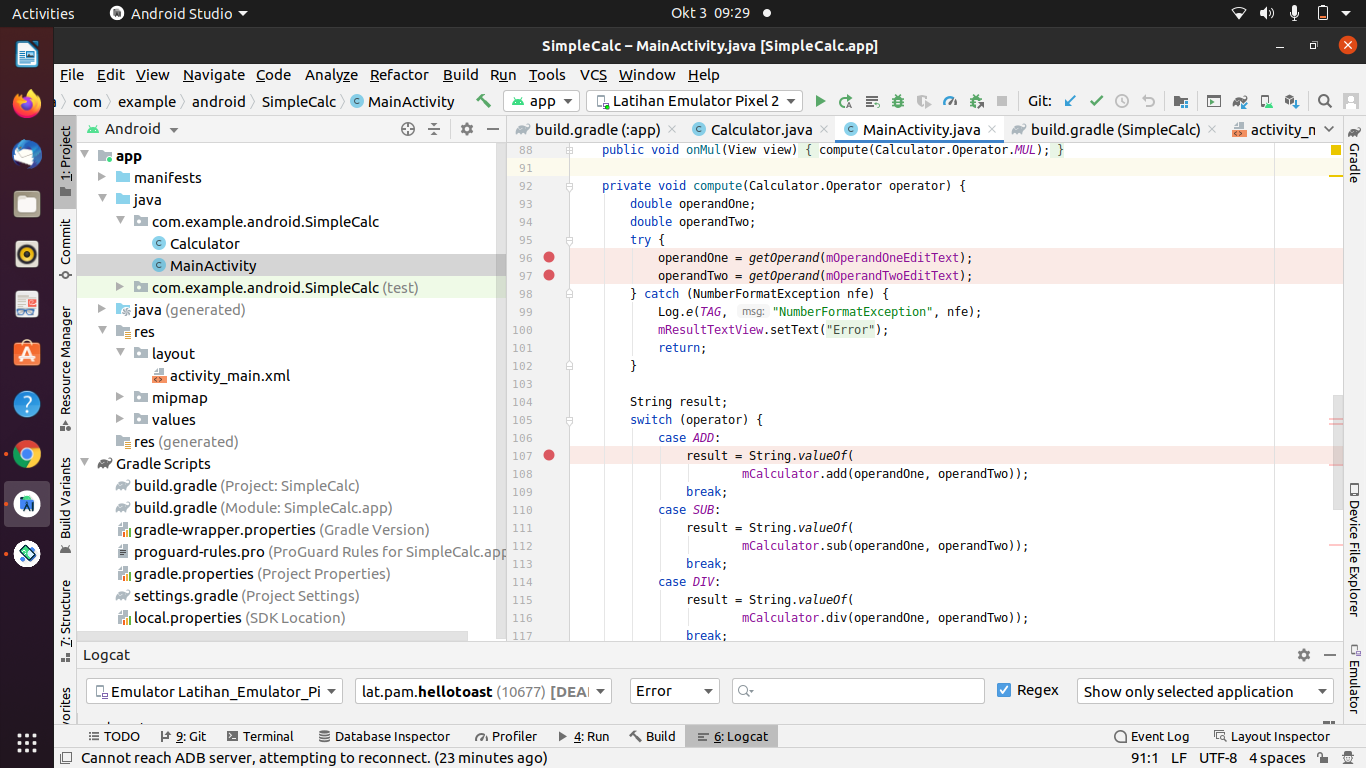
1. Lakukan build project kembali dengan cara memilih menu Build > Rebuild Project untuk memastikan tidak ada konfigurasi yang error.
2. Silahkan anda pahami kode program tersebut. Terdapat dua class pada java program yakni Calculator yang berisi method kalkulasi pada kalkulator yang terdiri dari Add (penambahan), Sub (pengurangan), Div (pembagian) dan Sub (perkalian). Kemudian satu class lagi (MainActivity) merupakan halaman utama dari program kalkulator tersebut.
3. Jalankan program tersebut pada emulator/ device dengan memasukkan nilai 10 untuk angka pertama dan kosongkan angka kedua. Selanjutnya, lakukan pembagian (Div) pada program kalkulator yang telah dibuat.



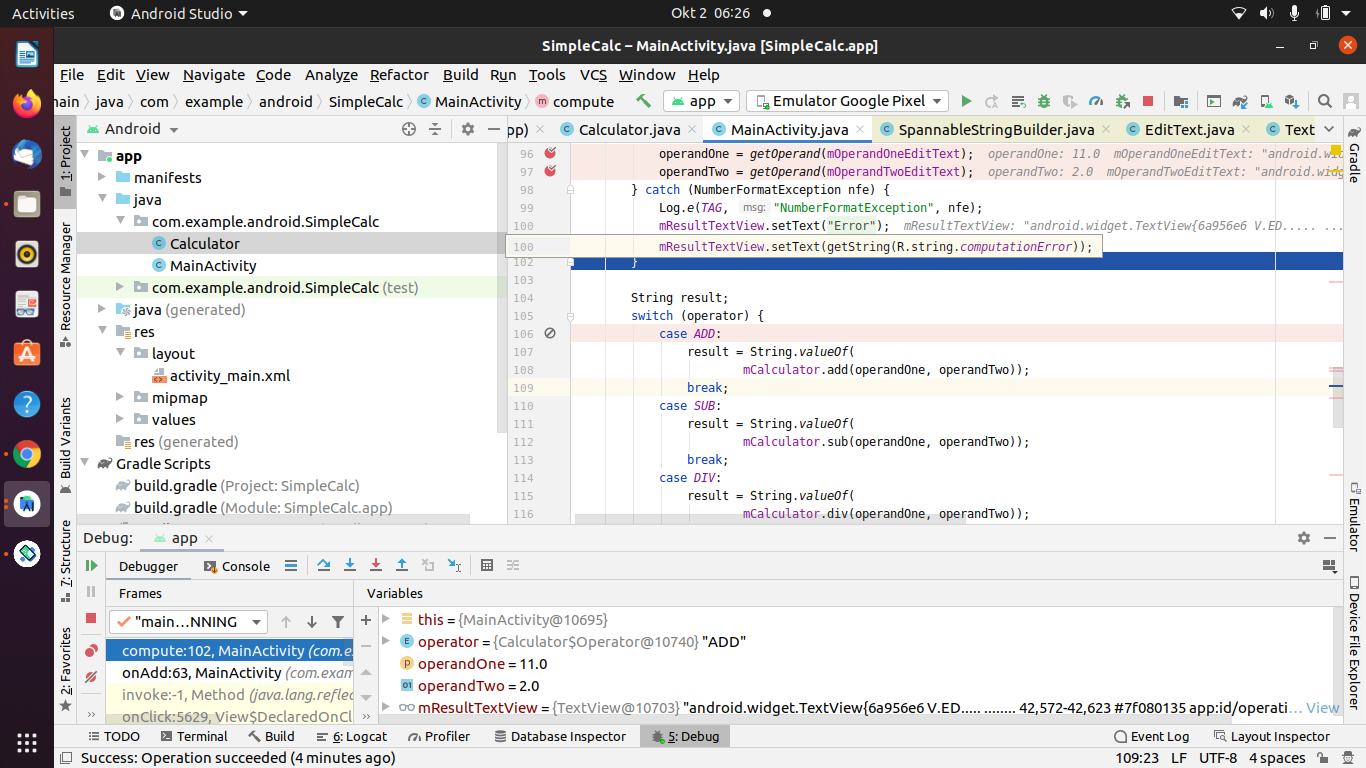
1. Pada program tersebut akan muncul error dan lihat Log Aplikasi tersebut pada logcat dimana terdapat error NumberFormatException seperti pada gambar dibawah ini.



1. Selanjutnya anda akan menjalankan aplikasi calculator tadi dalam mode Debug. Selanjutnya anda perlu menaruh breakpoints. Breakpoints digunakan untuk melihat eksekusi pada baris, method, atau melihat nilai dari variabel.
2. Tandai program tersebut dengan menaruh tombol merah pada baris program yang akan anda lihat. Kali ini anda akan menaruh breakpoints pada variabel **operandOne, operandTwo, resul**t dan baris **mResultTextView.setText(result)** seperti pada gambar dibawah ini.

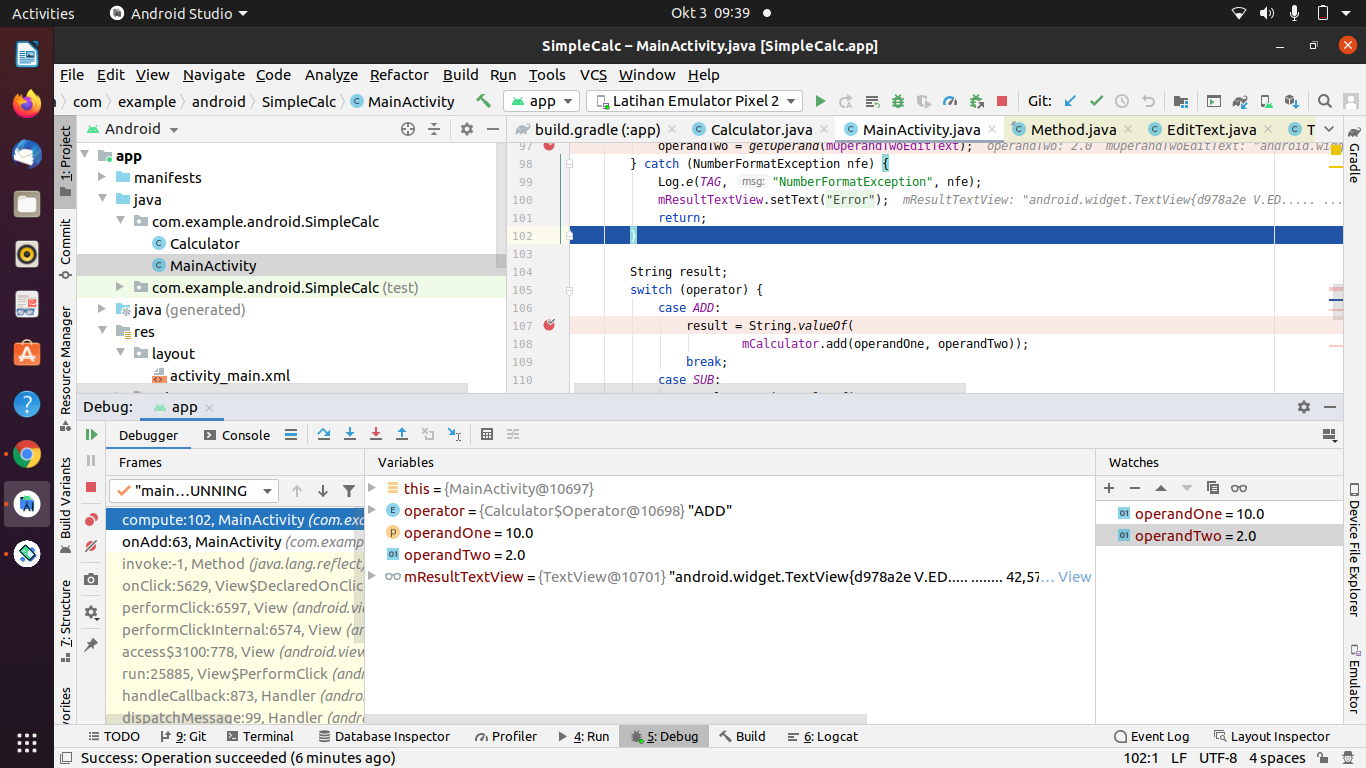


1. Jalankan mode debug tersebut dengan mengakses pada menu Run > Debug app atau menekan tombol serangga pada icon (dibawah menu bar).

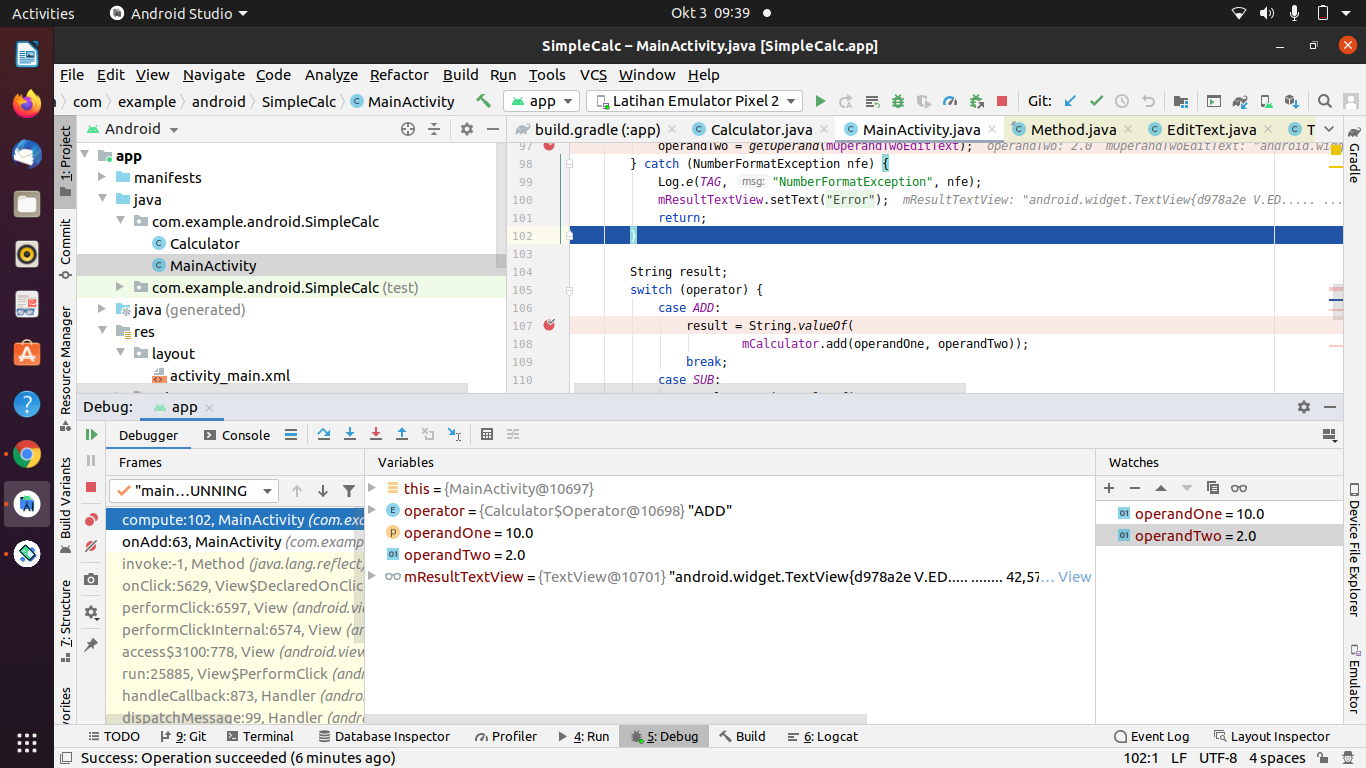


1. Terdapat tiga bagian mode yang dapat dieksplorasi yaitu:
   1. Frames: Untuk memperlihatkan Frames yang sedang dieksekusi dalam tumpukan frames
   2. Variables: Untuk memperlihatkan nilai dari variabel dari baris program yang sedang dijalankan
   3. Watches: Untuk menunjukkan nilai dari suatu baris program termasuk juga variabel yang sedang ingin dilihat

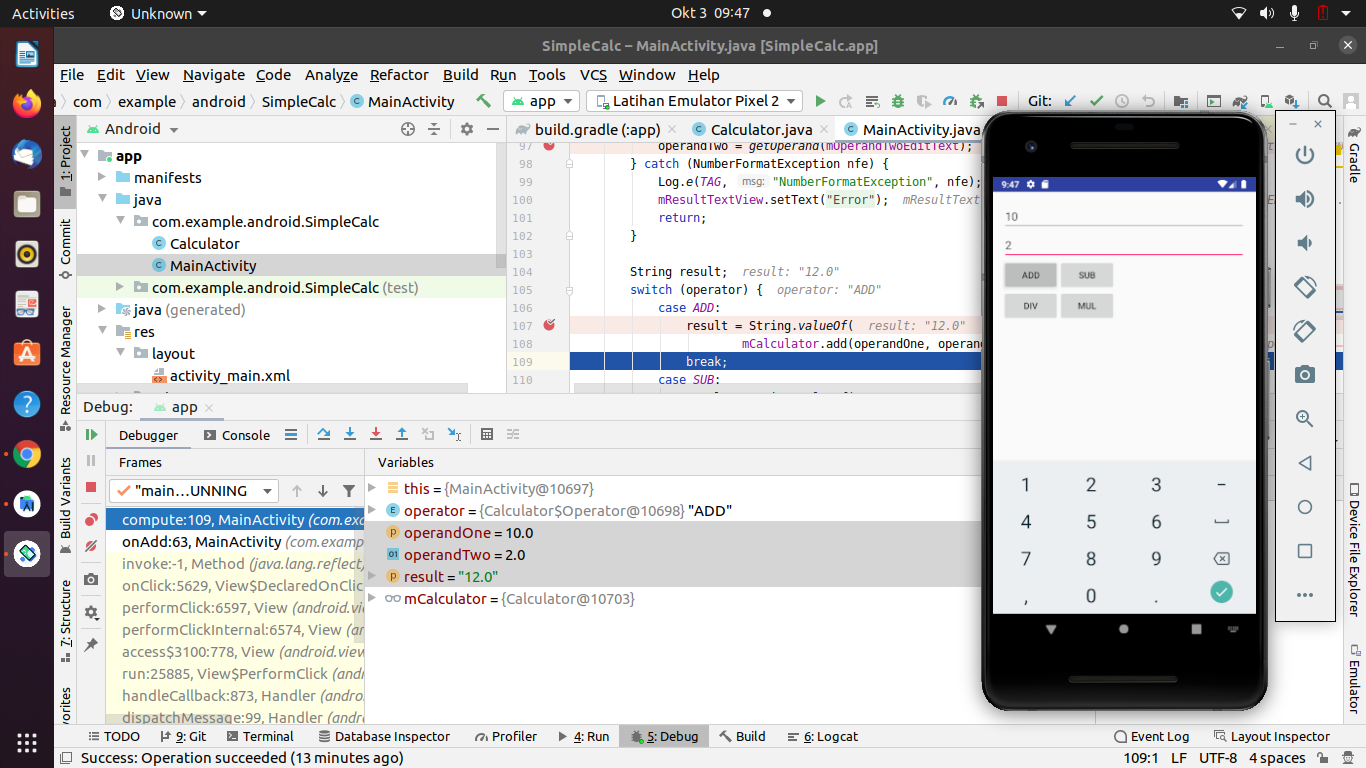
Mode tersebut dapat dilihat pada tampilan dibawah ini.



1. Jalankan setiap step pada program tersebut dengan menggunakan Step over dan Step Into untuk melihat langkah pada baris program yang dijalankan.



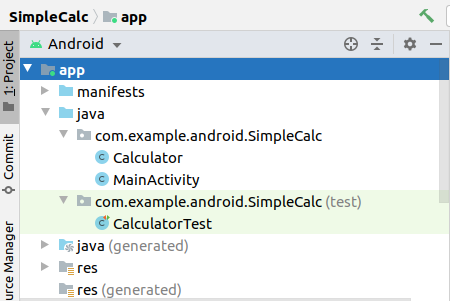
1. Pastikan nilai yang diinput sesuai dengan yang diinputkan, misalnya anda menginput nilai pada angka 1: 10, angka 2: 2. Sehingga menghasilkan nilai berikut.
   1. operatorOne = 10.0
   2. operatorTwo = 2.0
   3. Result = 12.0



## Modul 2: Melakukan Unit Testing pada Aplikasi

Pada modul kedua ini anda akan mencoba menjalankan unit testing dengan menggunakan Android Studio. Secara default, pada saat project tersebut dibuat sudah dibuatkan kode program sampel yang dapat digunakan untuk testing dan kita dapat mengubah kode program tersebut untuk kebutuhan testing yang dilakukan.

1. Gunakan project SimpleCalc yang digunakan pada project sebelumnya.
2. Jika kita membuka project tersebut maka kita akan temukan dua package :
   1. Main, digunakan untuk kode program utama yang dijalankan pada program
   2. Test, digunakan untuk kebutuhan testing pada aplikasi
3. Hal ini dapat kita lihat pada sidebar yang menampilkan isi dari project seperti pada tampilan berikut ini.



1. Didalam package test dapat kita temukan satu buah class CalculatorTest, anda dapat membuka class tersebut dimana terdapat:
   1. **Dependencies dari program** merupakan library yang digunakan seperti pada package **org.junit.**
   2. **@RunWith(JUnit4.class)** merupakan anotasi yang menandakan bahwa testing akan dijalankan pada class ini
   3. **@SmallTest** sebagai informasi yang disertakan pada JUnit bahwa method tersebut merupakan test method (JUnit 4)
   4. **setUp()** merupakan method yang digunakan untuk setup awal testing dan digunakan untuk menginisialisasi variabel atau objek yang digunakan untuk multiple test
   5. **addTwoNumbers()** merupakan method yang digunakan untuk testing
2. Untuk melakukan pengetesan anda dapat melihat kode eksisting yang sudah ada dengan melihat method **addTwoNumbers()** dimana dituliskan kode sebagai berikut.

*/\*\**

*\* Test for simple addition*

*\*/*

@Test

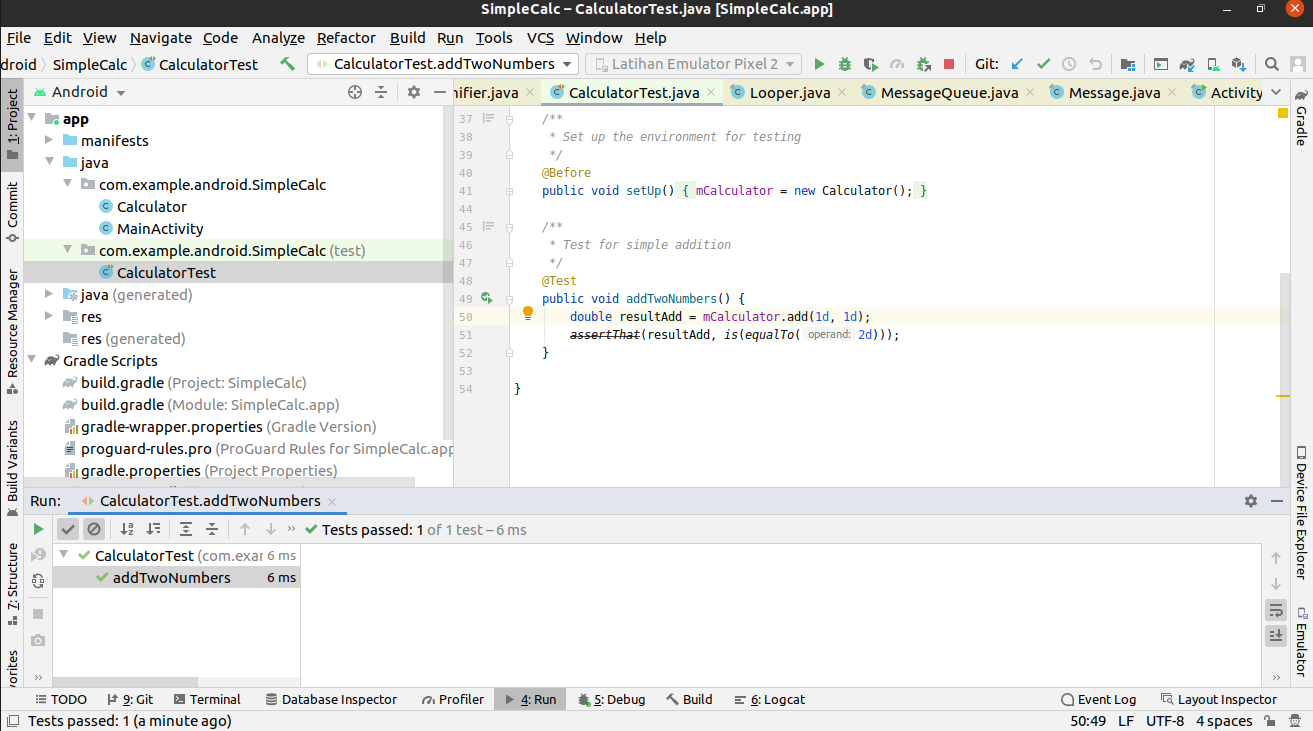
public void addTwoNumbers() {

double resultAdd = mCalculator.add(1d, 1d);

*assertThat*(resultAdd, *is*(*equalTo*(2d)));

}

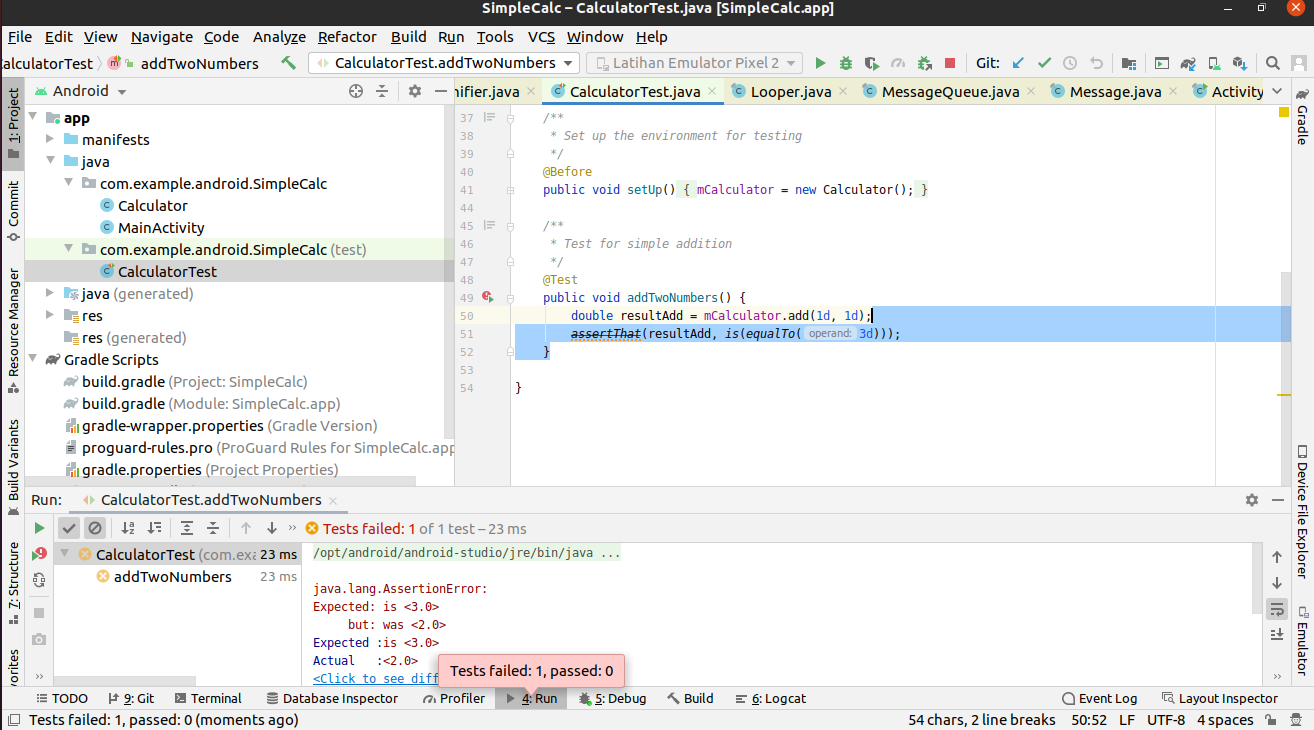
1. Jalankan program tersebut dengan melakukan Run pada Calculator Test.
2. Kode program tersebut menunjukkan bahwa hasil penambahan operand 1 dan operand 2 menghasilkan 2d. Jika sukses maka hasil test akan menunjukkan sebagai berikut.



1. Anda dapat mencoba dengan mengganti nilai yang dikeluarkan dengan mengganti kode program seperti dibawah ini.

*assertThat*(resultAdd, *is*(*equalTo*(3d)));

1. Jalankan program tersebut dengan melakukan Run pada Calculator Test.
2. Anda akan menemukan bahwa hasil unit test menunjukkan hasil yang tidak sesuai dan menunjukkan hasil Test failed seperti yang ditunjukkan pada tampilan berikut.



1. Selanjutnya anda akan menambahkan beberapa unit test lain dengan menambahkan method dari kondisi yang akan dilakukan pengujian. Tambahkan beberapa method dibawah ini.

@Test

public void addTwoNumbersNegative() {

double resultAdd = mCalculator.add(-1d, 2d);

*assertThat*(resultAdd, *is*(*equalTo*(1d)));

}

@Test

public void subTwoNumbers() {

double resultSub = mCalculator.sub(1d, 1d);

*assertThat*(resultSub, *is*(*equalTo*(0d)));

}

@Test

public void subWorksWithNegativeResult() {

double resultSub = mCalculator.sub(1d, 17d);

*assertThat*(resultSub, *is*(*equalTo*(-16d)));

}

@Test

public void mulTwoNumbers() {

double resultMul = mCalculator.mul(32d, 2d);

*assertThat*(resultMul, *is*(*equalTo*(64d)));

}

@Test

public void divTwoNumbers() {

double resultDiv = mCalculator.div(32d,2d);

*assertThat*(resultDiv, *is*(*equalTo*(16d)));

}

@Test

public void divTwoNumbersZero() {

double resultDiv = mCalculator.div(32d,0);

*assertThat*(resultDiv, *is*(*equalTo*(Double.*POSITIVE\_INFINITY*)));

}

1. Silahkan anda definisikan kira-kira method diatas ini untuk menguji nilai dari inputan yang seperti apa.
2. Jalankan program tersebut dengan melakukan Run pada Calculator Test. dan tunjukkan apakah method tersebut menghasilkan unit test yang sesuai ataupun tidak.