

BAB 2. PENGENALAN JAVA, XML, KOMPONEN DAN DASAR PEMBUATAN APLIKASI ANDROID

Daftar Isi

- A. Pengenalan Java, OOP, Good Programming Practice**
- B. XML Pada Android Studio**
- C. View, ViewGroup dan Komponen View**
- D. Contoh Praktek Mendesain Layout**
- E. Activity dan Intent**
- F. Trik Dasar Pemrograman di Android Studio**
- G. Contoh Aplikasi dengan Toast**
- H. Contoh Aplikasi Spinner**
- I. Contoh Aplikasi DatePicker**
- J. Font Awesome di Aplikasi Android**

© Copyright by Antonius – www.jasaplus.com All Rights Reserved

Barangsiapa didapati memperjual belikan materi training ini tanpa seizin dari pencipta maka pencipta berhak menuntut ganti rugi yang jumlahnya ditentukan oleh pencipta materi ini.

BAB 2. PENGENALAN JAVA, XML, KOMPONEN DAN DASAR PEMBUATAN APLIKASI ANDROID

A. Pengenalan Java, OOP, Good Programming Practice

Pada bagian ini kita akan mempelajari dasar java kemudian dilanjutkan pengenalan OOP pada java. Pada bagian terakhir akan membahas tentang good and bad programming practice di java.

A.1. Pengenalan JAVA

Java adalah bahasa pemrograman tingkat tinggi berorientasi objek.

Proyek java pertama kali diinisialisasi oleh Sun Microsystem pada tahun 1990 dengan projek yang dinamai green. Java diciptakan oleh developer Sun Microsystem : James Gosling, Mike Sheridan dan Patrick Naughton pada tahun 1991. Pada saat itu java masih bernama Oak hingga pada tahun 1995, Oak berganti nama menjadi java.

KARAKTERISTIK JAVA

Berikut ini beberapa karakteristik Bahasa Pemrograman Java

1. Lebih Simple dari C dan C++

Java merupakan bahasa pemrograman keluarga C. Di mana sintaks sintaks pada java lebih simple dari c dan c++.

2. Menekankan Konsep Pemrograman Berorientasi Objek

Java menekankan konsep pemrograman berorientasi. Pada java, base class (kelas dasar) untuk semua class dalam java adalah java.lang.Object. Semua kelas lain merupakan turunan dari java.lang.Object. Setiap kelas pada java memiliki fungsi yang kita sebut sebagai method (metode). Perhatian ! Pada java kita tidak menyebut fungsi seperti pada bahasa pemrograman lain, tetapi kita sebut sebagai method. Untuk detail lebih lengkap tentang object oriented programming di java bisa dilihat di bagian OOP (Object Oriented Programming) di java.

3. Terdistribusi

Pada java terdapat berbagai kelas dalam jumlah banyak yang mendukung java untuk berbagai macam komunikasi jaringan mulai dari ftp, http, wireless (2,4 GHz) dan lain sebagainya.

4. Robust

Java memiliki fitur yang kuat pada bagian exception handling (pengangguran kesalahan saat error).

5. Dapat Berjalan pada Bermacam Platform

Java dapat dijalankan pada beragam platform, mulai dari perangkat komputer, mobile dengan catatan platform tersebut sudah terinstall JVM (Java Virtual Machine)

6. Dinamis

Pada bahasa c++ jika terjadi perubahan pada parent class, maka perlu dilakukan kompilasi ulang secara menyeluruh, sedangkan Java menerapkan konsep interface sehingga tidak perlu dilakukan kompilasi ulang secara menyeluruh.

Untuk pengenalan java, kita mulai dengan membuat program java untuk print hello world di terminal.

HelloWorld.java :

```
class HelloWorld {  
    public static void main(String args[]) {  
        System.out.println("Hello World");  
    }  
}
```

Untuk kompile menjadi class ketik : javac HelloWorld.java

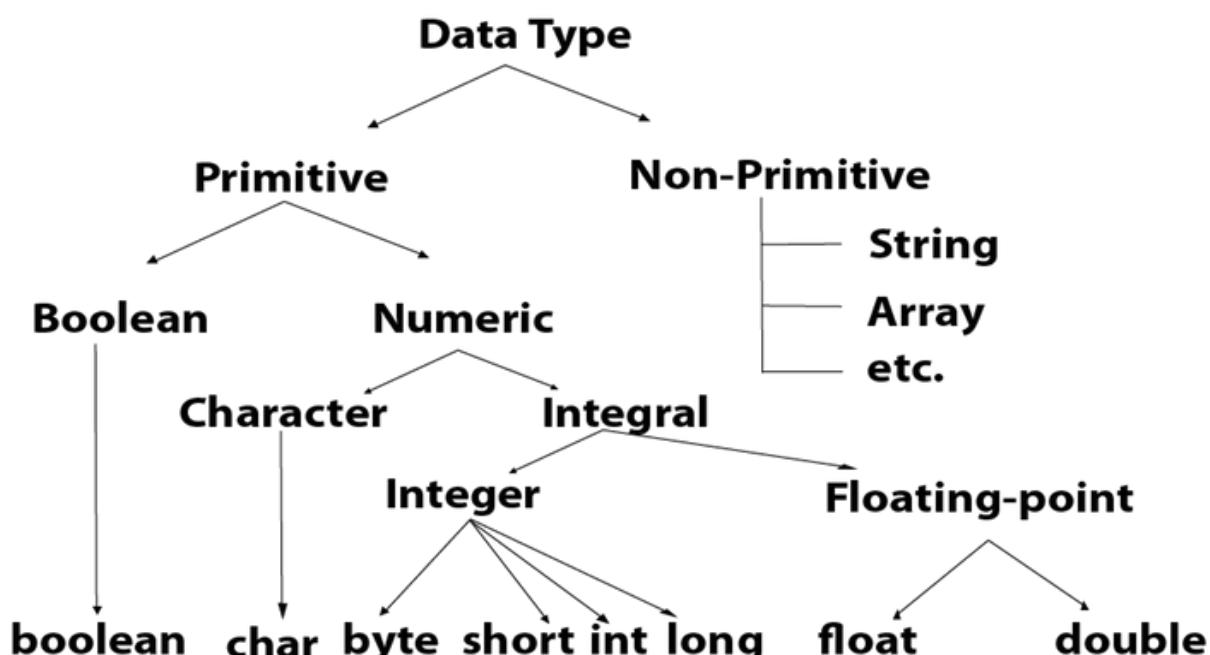
Selanjutnya agar jvm menjalankan kelas ketik : java HelloWorld

TIPE DATA , VARIABEL DAN ARRAY

Tipe Data

Pada java dikenal 2 macam tipe data yang menggolongkan variabel ke dalam 2 golongan utama yaitu : primitive dan non primitive.

Untuk lebih lengkapnya tentang penggolongan ini bisa dilihat pada gambar di bawah ini :



Variabel

Variabel adalah suatu penyimpanan yang bisa diisi nilai. Variabel bisa juga diartikan suatu nama yang merupakan penanda daerah di memori di mana daerah di memori ini digunakan sebagai tempat penyimpanan. Pada java dikenal 3 jenis variabel yang sering dipakai :

1. variabel lokal

merupakan variabel yang dideklarasikan pada tubuh suatu metode

2. variabel instance (non static)

variabel ini dideklarasi di dalam kelas tetapi di luar metode. Variabel ini biasa digunakan untuk menyimpan atribut masing masing objek dari kelas.

Contoh:

```
class Hacker {  
    int age;  
}
```

Pada contoh di atas variabel bertipe integer age, bisa berbeda beda untuk masing masing objek hasil instance.

3. variabel static

variabel ini dideklarasikan dengan keyword “static” variabel ini juga dikenal sebagai variabel kelas (class variable). Variabel ini adalah bersifat tetap dan tidak bisa berubah ubah.

Contoh:

```
class Hacker {  
    static String name;  
}
```

Pada contoh di atas variabel bertipe String bersifat tetap setiap kali proses instance dari kelas tersebut diinisialisasi.

Array

Array adalah penyimpanan yang digunakan untuk menyimpan beragam nilai dalam satu variabel.

Pola deklarasi array:

```
dataType[] nama_array;
```

Contoh array:

```
String[] robots;  
  
String[] Members = {"adi", "jose", "joseph"};  
  
double[] Nilai;
```

untuk membuat array, bisa juga menggunakan keyword new. Contoh:

```
String[] Exploits = new String[10];
```

OPERATOR PADA JAVA

Berikut ini adalah beberapa jenis operator pada java :

1. Operator penjumlahan (arithmetic)

operator ini digunakan untuk operasi matematika.

Contoh :

+

digunakan untuk penjumlahan, contoh $a + b$

-

digunakan untuk pengurangan

digunakan untuk perkalian

/

digunakan untuk pembagian

%

digunakan untuk modulus

++

digunakan untuk menambah nilai variabel dengan 1

--

digunakan untuk mengurangi nilai variabel dengan 1

2. Operator penunjukan (assignment)

=

digunakan untuk mengisi nilai dari operand kanan ke kiri

+=

digunakan untuk menambah operand kiri sejumlah operand kanan, hasil disimpan di operand kiri

-=

digunakan untuk mengurangi operand kiri sejumlah operand kanan, hasil disimpan di operand kiri

/=

digunakan untuk membagi operand kiri dengan operand kanan, hasil disimpan di operand kiri

%=

digunakan untuk operasi modulus operand kiri dengan operand kanan, hasil disimpan di operand kiri

&=

digunakan untuk operasi bitwise AND operand kiri dan operand kanan, hasil disimpan di operand kiri.

|=

digunakan untuk operasi bitwise AND operand kiri dan operand kanan, hasil disimpan di operand kiri.

^=

digunakan untuk operasi bitwise OR operand kiri dan operand kanan, hasil disimpan di operand kiri.

>>=

digunakan untuk operasi bitwise shift kanan operand kiri dan operand kanan, hasil disimpan di operand kiri.

<<=

digunakan untuk operasi bitwise shift kiri operand kiri dan operand kanan, hasil disimpan di operand kiri.

3. Operator perbandingan (comparison / relational)

merupakan operator yang digunakan untuk operasi perbandingan

==

digunakan untuk mebandingkan apakah operand kiri dan kanan sama, jika sama maka akan menghasilkan boolean true

!=

digunakan untuk mebandingkan apakah operand kiri dan kanan berbeda, jika berbeda maka akan menghasilkan boolean true

>

digunakan untuk mebandingkan apakah operand kiri nilainya di atas operand kanan, jika ya maka akan menghasilkan boolean true

<

digunakan untuk mebandingkan apakah operand kiri nilainya di bawah operand kanan, jika ya maka akan menghasilkan boolean true

>=

digunakan untuk mebandingkan apakah operand kiri nilainya di atas atau sama dengan operand kanan, jika ya maka akan menghasilkan boolean true

<=

digunakan untuk mebandingkan apakah operand kiri nilainya di bawah atau sama dengan operand kanan, jika ya maka akan menghasilkan boolean true

4. Operator Logika (logical)

merupakan operator yang digunakan untuk operasi logika

&&

digunakan untuk operator logika and

||

digunakan untuk operator logika or

!

digunakan untuk operator logika not

5. Operator Bitwise

merupakan operator yang digunakan untuk operasi bitwise

&

digunakan untuk operasi bitwise and operand kiri dengan kanan

|
digunakan untuk operasi bitwise or operand kiri dengan kanan

^
digunakan untuk operasi bitwise xor operand kiri dengan kanan

~
digunakan untuk operasi bitwise and operand kiri dengan kanan

<<
digunakan untuk operasi bitwise shift kiri di mana nilai operand kiri digeser ke kiri sejumlah nilai bit dari operator kanan

>>
digunakan untuk operasi bitwise shift kanan di mana nilai operand kiri digeser ke kanan sejumlah nilai bit dari operator kanan

>>>
digunakan untuk operasi bitwise shift kanan di mana nilai operand kiri digeser ke kanan sejumlah nilai bit dari operator kanan dan nilai yang sebelumnya diganti menjadi 0.

KEYWORD PADA JAVA

Keyword pada java merupakan kata yang tidak bisa digunakan sebagai identifier. Keyword tidak bisa digunakan untuk nama kelas, nama variabel, label dan nama method.

abstract

keyword ini digunakan untuk mendeklarasikan abstraksi di java. Abstract bisa diaplikasikan ke method dan class.

assert

pada java, assert bisa digunakan untuk debuggin, assert digunakan untuk menguji suatu kondisi apakah true atau false. Assert biasa digunakan untuk unit test.

boolean

jenis tipe data yang menyimpan nilai true atau false.

break

keyword untuk keluar dari pengulangan (loop)

byte

tipe data yang bisa menyimpan data yang berukuran 8 bit

case

keyword yang digunakan untuk kontrol switch, cases merupakan elemen dari switch di mana case akan menguji nilai yang diberikan switch dengan nilai dari setiap case.

catch

digunakan untuk menangkap saat terjadinya exception yang berasal dari rutin di dalam try.

char

tipe data yang bisa digunakan menyimpan karakter berukuran 16 bit yang jenisnya unsigned (hanya menyimpan nilai positif tidak bisa menyimpan nilai kurang dari nol). Unsigned pada char artinya nilai ascii dari karakter tidak boleh kurang dari 0.

class

digunakan untuk mendeklarasikan suatu class

continue

biasa digunakan saat looping (pengulangan) di java. Dengan keyword continue maka looping akan melompat langsung ke iterasi selanjutnya di dalam pengulangan, di mana rutin di bawah continue di dalam satu iterasi akan dilompati.

default

keyword ini sering digunakan saat kontrol switch, di mana jika seluruh case lain tidak terpenuhi dan terdapat default pada switch tersebut, maka statement di dalam blok default akan dieksekusi secara otomatis.

Selain digunakan pada switch, keyword default juga merupakan modifier untuk akses. Pada saat deklarasi suatu variabel atau method jika kita tidak memberikan modifier akses (contoh modifier akses : private, public) maka variabel atau method tersebut akan secara otomatis memiliki modifier akses default.

do

keyword ini biasa digunakan untuk pengulangan (looping) bersamaan dengan keyword while (do-while loop)

double

suatu tipe data di java di mana memori akan dialokasikan untuk menampung data berjenis floating point 64 bit.

else

digunakan untuk percabangan dari kondisional if.

enum

keyword ini digunakan untuk deklarasi suatu kelas enum. Kelas enum merupakan kelas yang khusus di java, di mana kelas ini berisi tipe data enum yang merupakan konstan.

extends

keyword ini mengindikasikan bahwa suatu kelas merupakan child class dari suatu kelas atau interface.

final

merupakan indikasi suatu variabel menyimpan nilai konstan (tidak bisa diubah), jika digunakan pada suatu method maka method tersebut tidak bisa dioverride

finally

merupakan keyword yang digunakan untuk suatu blok setelah struktur try-catch, digunakan untuk memastikan blok kode di dalam finally dieksekusi.

float

tipe data yang bisa digunakan untuk menyimpan angka floating point berukuran 32 bit

for

keyword ini digunakan untuk memulai pengulangan for loop

if

digunakan untuk menguji suatu blok statement (kondisional)

implements

keyword ini biasa digunakan bersamaan dengan suatu deklarasi kelas, keyword ini menunjukkan suatu kelas mengimplementasikan suatu interface

import

digunakan untuk mengimport referensi dari kelas lain

instanceof

keyword ini merupakan indikasi bahwa suatu objek merupakan instance dari suatu kelas atau mengimplementasikan suatu interface

int

jenis tipe data yang bisa digunakan menampung data berupa integer 32 bit bertanda (bisa minus)

interface

digunakan untuk deklarasi suatu interface

long

tipe data yang dapat digunakan untuk menampung data berupa integer 64 bit

native

dengan keyword ini artinya suatu method diimplementasikan menggunakan kode native (khusus untuk suatu platform)

new

digunakan untuk membuat objek baru dari suatu kelas

null

digunakan untuk indikasi suatu referensi yang tidak ada isi apapun (kosong)

package

digunakan untuk deklarasi suatu paket di java

private

merupakan access modifier yang mengindikasikan suatu method atau variabel hanya bisa diakses di dalam kelasnya sendiri

protected

merupakan suatu access modifier yang mengindikasikan suatu method atau variabel hanya bisa diakses oleh kelasnya sendiri, subclass dan kelas lainnya selama masih di dalam paket yang sama

public

merupakan suatu access modifier yang mengindikasikan suatu kelas atau interface atau method atau variabel bisa diakses dari manapun.

return

digunakan untuk mengembalikan kontrol program atau nilai kembali dari pemanggilan suatu method

short

suatu jenis tipe data yang digunakan untuk menyimpan data berupa integer 16 bit

static

keyword ini digunakan untuk pengaturan memori untuk alokasi variabel atau method di java, penggunaan static berguna untuk efisiensi penggunaan memori. Keyword ini digunakan untuk suatu variabel atau method.

Jika digunakan pada variabel maka variabel merupakan variabel untuk suatu kelas (class variable). Variabel ini adalah bersifat tetap dan tidak bisa berubah ubah. Variabel ini hanya dideklarasi sekali saat kelas diload di memori.

Jika digunakan pada method maka metode tersebut bisa digunakan untuk mengakses variabel static dan bisa mengubah nilai dari variabel static. Static method juga bisa dipanggil tanpa perlu membuat instance dari suatu kelas.

strictfp

keyword java yang digunakan untuk membatasi presisi dan pembulatan dari perhitungan tipe data floating point, digunakan untuk menjamin kode yang portable (mudah dimigrasikan ke arsitektur lain)

super

referensi untuk suatu base class dari suatu kelas, biasa digunakan di dalam method atau constructor

switch

digunakan untuk percabangan dengan switch. Keyword ini mengindikasikan percabangan berdasarkan nilai yang dites.

synchronized

digunakan untuk bagian atau metode yang kritis / rawan pada suatu kode multithreading

this

referensi ke objek saat ini di dalam suatu metode atau konstruktor

throw

digunakan untuk membuat suatu exception

throws

mengindikasikan exception yang bisa dimunculkan oleh suatu method

transient

menentukan bahwa suatu variabel bukan bagian dari objek yang bersifat tetap.

try

digunakan untuk memulai suatu blok kode yang akan dites untuk exception.

void

digunakan pada method yang tidak memiliki return value (nilai kembali)

volatile

keyword ini digunakan sebagai indikasi suatu variabel bisa berubah sewaktu waktu. Dengan volatile, maka suatu kelas menjadi thread safe, dengan keyword ini suatu metode atau instance dari kelas yang mengakses variabel yang menggunakan keyword ini bisa dieksekusi bersamaan pada beberapa thread tanpa masalah. Masalah di sini bisa berupa hanya bug biasa yang menyebabkan jalanya aplikasi menjadi tidak benar hingga bisa juga menyebabkan celah keamanan yang memanfaatkan celah bug pada aplikasi dengan multithreading yaitu race condition.

while

digunakan untuk memulai kontrol pengulangan dengan while

ALUR KONTROL PADA JAVA

Pada java dikenal berbagai macam alur kontrol jalanya aplikasi. Berikut ini adalah alur kontrol pada java :

1. PENGULANGAN (looping statement)

Pada suatu bahasa pemrograman, pengulangan (looping) digunakan untuk mengeksekusi suatu blok instruksi secara berulang ketika suatu kondisi terpenuhi. Ada 3 jenis pengulangan pada java : for, while dan do-while

for loop

Pola dari suatu pengulangan dengan for loop :

```
for (initialization condition; testing condition; increment/decrement) {  
    statement(s)  
}
```

contoh :

```
class ForSample {  
  
    public static void main(String[] args) {  
        for (int a = 0; a <= 10; a++) {  
            System.out.println("blok for");  
        }  
    }  
}
```

while loop

Pola dari suatu pengulangan dengan while loop :

```
while (boolean condition) {  
    statement(s)  
}
```

contoh :

```
class WhileSample {  
  
    public static void main(String[] args) {  
        int x = 1;  
  
        while (x <= 5) {  
            System.out.println("blok statemen");  
            x++;  
        }  
    }  
}
```

do-while loop

Pola dari suatu pengulangan dengan do-while loop :

```
do {  
    statement(s)  
}  
while (condition);
```

contoh :

```
class DoWhileSample {  
  
    public static void main(String[] args) {  
        int a = 1;  
  
        do {  
            System.out.println("blok do");  
            a++;  
        }  
        while (a < 10);  
    }  
}
```

2. PENGUJIAN KONDISI (decision making statement)

terdiri dari : if dan switch

if

Digunakan untuk mengeksekusi blok kode jika kondisi terpenuhi. Pola dari pengujian kondisi dengan if :

```
if (some expression is true) {  
    do something;  
}
```

contoh :

```
/*  
IfSample.java  
*/  
class IfSample {
```

```

public static void main(String[] args) {
    int a = 10;
    int b = 20;

    if (b > a) {
        System.out.println("b > a");
    }
}

```

if-else

Merupakan varian dari if statement. Digunakan untuk melakukan pengujian suatu kondisi, jika kondisi true maka lakukan eksekusi, jika kondisi tidak terpenuhi, maka jalankan statement di dalam blok kode else. Pola pengujian kondisi dengan if-else :

```

if (some expression is true) {
    do something;
}
else {
    do something else;
}

```

contoh :

```

/*
IfElseSample.java
*/
class IfElseSample {

    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        if (b > a) {
            System.out.println("b > a");
        }
        else {
            System.out.println("a > b");
        }
    }
}

```

if-else-if

Merupakan varian dari pengujian kondisi dengan if di mana pengujian kondisi lebih dari satu. Pola dari if-else-if:

```

if (condition1) {
    statement(s)
} else if (condition2) {
    statement(s)
} else {
    statement(s)
}

```

Contoh :

```
/*
IfElseIfSample.java
*/
class IfElseIfSample {

    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        if (b > a) {
            System.out.println("b > a");
        }
        else if (a > b) {
            System.out.println("a > b");
        }
        else {
            System.out.println("match");
        }
    }
}
```

switch

Digunakan untuk memilih salah satu dari blok kode yang memenuhi persyaratan.

Pola :

```
switch(expression) {
    case x:
        statement(s)
        break;
    case y:
        statement(s)
        break;
    default:
        statement(s)
}
```

contoh :

```
class SwitchSample {
    public static void main(String[] args) {
        int a = 2;

        switch (a) {
            case 1:
                System.out.println("1");
                break;
            case 2:
                System.out.println("2");
                break;
            case 3:
                System.out.println("3");
        }
    }
}
```

```
        break;
        case 4:
            System.out.println("4");
            break;
        case 5:
            System.out.println("5");
            break;
    }
}
}
```

MODIFIER PADA JAVA

Pada bagian ini kita akan mempelajari modifier pada java. Modifier adalah keyword pada java yang digunakan untuk memodifikasi arti dari suatu definisi. Secara garis besar, modifier pada java dapat digolongkan ke dalam 2 macam :

1. Modifier untuk Akses

Modifier ini digunakan untuk mengatur skope dari suatu class, constructor, method, variable. Ada 4 modifier akses pada java : default, private, protected, public

default

keyword default merupakan modifier untuk akses. Pada saat deklarasi suatu variabel atau method jika kita tidak memberikan modifier akses maka variabel atau method tersebut akan secara otomatis memiliki modifier akses default. Secara default, suatu method atau variabel yang aksesnya default hanya bisa diakses di dalam 1 paket aplikasi yang sama.

private

merupakan access modifier yang mengindikasikan suatu method atau variabel hanya bisa diakses di dalam kelasnya sendiri

protected

merupakan suatu access modifier yang mengindikasikan suatu method atau variabel hanya bisa diakses oleh kelasnya sendiri, subclass dan kelas lainnya selama masih di dalam paket yang sama

public

merupakan suatu access modifier yang mengindikasikan suatu kelas atau interface atau method atau variabel bisa diakses dari manapun.

2. Modifier Non Akses

Modifier ini adalah modifier yang tidak berhubungan dengan skope akses. Modifier non akses pada java antara lain : static, final, abstract, synchronized, transient, volatile, native

static

keyword ini digunakan untuk pengaturan memori untuk alokasi variabel atau method di java, penggunaan static berguna untuk efisiensi penggunaan memori. Keyword ini digunakan untuk suatu variabel atau method.

Jika digunakan pada variabel maka variabel merupakan variabel untuk suatu kelas (class variable). Variabel ini adalah bersifat tetap dan tidak bisa berubah ubah. Variabel ini hanya dideklarasi sekali saat kelas diload di memori.

final

merupakan indikasi suatu variabel menyimpan nilai konstan (tidak bisa diubah), jika digunakan pada suatu method maka method tersebut tidak bisa dioverride

abstract

keyword ini digunakan untuk mendeklarasikan abstraksi di java. Abstract bisa diaplikasikan ke method dan class.

synchronized

digunakan untuk bagian atau metode yang kritis / rawan pada suatu kode multithreading

transient

menentukan bahwa suatu variabel bukan bagian dari objek yang bersifat tetap.

volatile

keyword ini digunakan sebagai indikasi suatu variabel bisa berubah sewaktu waktu. Dengan volatile, maka suatu kelas menjadi thread safe, dengan keyword ini suatu metode atau instance dari kelas yang mengakses variabel yang menggunakan keyword ini bisa dieksekusi bersamaan pada beberapa thread tanpa masalah. Masalah di sini bisa berupa hanya bug biasa yang menyebabkan jalanya aplikasi menjadi tidak benar hingga bisa juga menyebabkan celah keamanan yang memanfaatkan celah bug pada aplikasi dengan multithreading yaitu race condition.

native

dengan keyword ini artinya suatu method diimplementasikan menggunakan kode native (khusus untuk suatu platform)

A.2. OOP (Object Oriented Programming) di Java

OOP (object oriented programming) merupakan konsep / paradigma pemrograman yang berorientasi pada objek . Tujuan dari OOP adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Suatu aplikasi yang menerapkan konsep OOP bisa memiliki 4 karakteristik dasar OOPS : abstraction, inheritance, encapsulation dan polymorphism.

Konsep OOP pada java :

1. Class

class merupakan suatu model / cetak biru / kerangka dasar yang berisi kumpulan atribut, metode dalam satu unit.

Sebagai contoh analogi : family Felidae atau kucing merupakan kelas yang mendefinisikan family kucing berupa karakteristik yang dimiliki secara garis besar dari objek objek di dalamnya.

Pada java, nama kelas utama dan nama file adalah sama. Contoh : class Bird maka akan menggunakan filename Bird.java.

Komponen yang bisa terdapat pada definisi suatu kelas :

Modifier

Pada bagian modifier ini menentukan jenis hak akses suatu kelas. Pada saat kelas dideklarasikan dari kelas lain, maka kelas tersebut hanya bisa dideklarasikan dengan 2 jenis modifier yaitu : public dan default (tidak ada modifier access yang dideklarasikan).

Jika kelas tersebut dideklarasikan sebagai kelas utama (nama kelas sama dengan nama file) maka kelas tersebut bisa dideklarasikan dengan 4 macam modifier di bawah ini:

1. private

hanya bisa diakses di dalam kelas itu sendiri

2. default

bisa diakses hanya dari kelas kelas dalam satu paket saja

3. protected

hanya bisa diakses dari kelas kelas dalam satu paket dan hanya dari sub class

4. public

bisa diakses dari manapun

Nama Class

Nama dari kelas yang akan dideklarasikan biasanya diawali huruf besar

Super Class

Nama kelas induk jika ada

Body

Bagian tubuh dari kelas merupakan bagian yang diapit tanda kurawal : {}

Contoh java class:

```
/*
java class example for course material
*/
class Bird {
    String wakeup_time = "06:00 am";
    String sleep_time = "06:00 pm";

    public void WakeUp() {
        System.out.println("\nChirp chirp I wake up at " + wakeup_time);
    }

    public void Sleep() {
        System.out.println("\nChirp chirp ... sleep at " + sleep_time);
    }

    public static void main(String args[])
    {
        Bird bird = new Bird();
        bird.WakeUp();
        bird.Sleep();
    }
}
```

PENJELASAN

Pada contoh di atas, kita membuat kelas baru dengan nama Bird. Kelas tersebut memiliki 3 metode yaitu : WakeUp, Sleep dan main. Metode main di sini wajib ada jika kelas tersebut akan dijalankan secara langsung.

2. Interface

Java memperkenalkan konsep “interface”. Pada saat deklarasi suatu kelas yang mengimplementasi interface, jika implementasi dari metode tidak ada maka kelas tersebut harus didefinisikan sebagai abstract class. Interface memiliki kemiripan dengan kelas (class), konsep interface ditujukan untuk kebutuhan “abstraction” di aplikasi java.

- Interface memiliki kemiripan dengan kelas di mana suatu interface bisa memiliki metode dan variabel
- Metode yang dideklarasi di dalam suatu interface tidak memiliki implementasi, sedangkan metode yang dideklarasi di dalam kelas bisa memiliki implementasi

contoh deklarasi kelas dengan implementasi interface :

```
/*
AngryBird.java
java interface example
*/
interface bird {
    void fly();
}

class AngryBird implements bird {
    public void fly() {
        System.out.println("\nfly implementation\n");
    }

    public void echo() {
        System.out.println("\nchirp chirp\n");
    }

    public static void main (String[] args) {
        AngryBird ab = new AngryBird();
        ab.fly();
        ab.echo();
    }
}
```

PENJELASAN

Pada contoh di atas, kita membuat interface baru dengan nama bird dengan metode tanpa implementasi : fly.

Selanjutnya di bawahnya kita membuat kelas baru yang mengimplementasikan interface bird dengan nama kelas AngryBird. Kelas AngryBird memiliki implementasi metode fly.

3. Object

object merupakan perwujudan (turunan) dari class, setiap object akan mempunyai attribute dan method yang dimiliki oleh class-nya.

Contoh sederhana untuk menjelaskan object, contoh : kelas kucing memiliki objek : anggora, kucing kampung, harimau, leopard dan lain sebagainya yang termasuk ke dalam kelas “kucing”

Contoh deklarasi objek :

```
/*
java class example for course material
*/
class Kucing {
    String name;
    int age;

    /* class constructor */
    public Kucing(String name_param, int age_param) {
        this.name = name_param;
        this.age = age_param;
    }

    public void Meow() {
        System.out.println("meow...");
    }

    public void PrintMyProperties() {
        System.out.println("My name is " + this.name);
        System.out.println("My age is " + this.age);
    }

    public void Breed() {
        System.out.println("Give birth");
    }

    public void Defecate() {
        System.out.println("Discharge");
    }

    public void Urinate() {
        System.out.println("pee piss");
    }

    public static void main(String args[])
    {
        Kucing kampung = new Kucing("Pussy", 5);
        Kucing anggora = new Kucing("Katy", 7);

        kampung.PrintMyProperties();
        anggora.PrintMyProperties();

        kampung.Breed();
        anggora.Urinate();

        kampung.Defecate();
        anggora.Meow();
    }
}
```

PENJELASAN

Pada contoh di atas ada deklarasi 2 objek yaitu kampung dan anggora. Kedua objek ini masing masing merupakan instance dari kelas Kucing.

Karena sama sama objek yang berasal dari kelas Kucing, maka baik kampung maupun anggora memiliki metode bawaan yang sama sama berasal dari kelas Kucing.

Perbedaan objek anggora dan kampung adalah saat pemberian parameter yang menyebabkan karakteristik objek anggora dan kampung berbeda.

4. Encapsulation

Encapsulation merupakan pembungkusan atribut, metode ke dalam satu unit tunggal dan membatasi akses pada beberapa komponen pada objek.



Sebagai analogi, suatu kapsul merupakan satu kesatuan yang terdiri dari berbagai macam jenis obat di dalamnya. Pada enkapsulasi, data di dalam suatu kelas tersembunyi di dalam kelas lainnya. Enkapsulasi dicapai dengan cara mendeklarasikan variabel di dalam kelas dengan modifier private dan menulis metode public untuk set dan get nilai variabel tersebut.

Berikut ini contoh konsep enkapsulasi:

Hacker.java:

```
/*
Hacker.java
example for course material
*/
public class Hacker {
    private String Name;
    private int SkillCounts;

    public void Properties() {
        System.out.println("Name : " + Name);
        System.out.println("Skill Count : " + SkillCounts);
    }

    public String getName() {
        return Name;
    }

    public void setName(String nama) {
        Name = nama;
    }

    public void SetSkillCounts(int jum) {
        SkillCounts = jum;
    }

    public int getSkillCounts() {
        return SkillCounts;
    }
}
```

Coder.java :

```
/*
Coder.java
example for course material
*/
public class Coder {
    public static void main (String[] args) {
        Hacker joanna = new Hacker();
        joanna.setName("Joanna");
        joanna.SetSkillCounts(7);

        joanna.Properties();
    }
}
```

Carder.java :

```
/*
Carder.java
example for course material
*/
public class Carder {
    public static void main (String[] args) {
        Hacker leroux = new Hacker();
        leroux.setName("Paul");
        leroux.SetSkillCounts(2);

        leroux.Properties();
    }
}
```

PENJELASAN

Pada kelas Hacker.java, enkapsulasi dicapai dengan membuat metode get dan metode set untuk private variable di dalam kelas.

5. Inheritance

Inheritance merupakan proses mewariskan karakteristik dari suatu kelas untuk kelas turunannya. Untuk mewariskan metode dan atribut dari kelas parent maka saat deklarasi child kelas, kita gunakan keyword “extends”

Berikut ini contoh konsep inheritance :

Tanaman.java :

```
/*
Tanaman.java
example for course material
*/
class Tanaman {

    public static void Tumbuh() {
        System.out.println("Tumbuh");
    }
}
```

```
public static void Layu() {
    System.out.println("Layu");
}
}
```

Samsit.java :

```
/*
Samsit.java
example for course material
*/
class Samsit extends Tanaman {
    public static void main (String[] args) {
        Tumbuh();
        Layu();
    }
}
```

Antanan.java :

```
/*
Antanan.java
example for course material
*/
class Antanan extends Tanaman {
    public static void main (String[] args) {
        Tumbuh();
        Layu();
    }
}
```

PENJELASAN

Pada contoh di atas, kelas Antanan dan kelas Samsit merupakan kelas yang mewariskan metode dari parent class Tanaman yaitu : Tumbuh() dan Layu().

6. Polymorphism

Secara sederhana polymorphism adalah konsep di mana suatu aplikasi bisa mengeksekusi satu aksi dengan berbagai cara yang berbeda. Biasanya polymorphism terjadi ketika kita memiliki banyak kelas yang terkait satu sama lain melalui inheritance dari parent class.

Untuk lebih lengkapnya, polymorphism dapat didefinisikan tergantung konteks penggunaanya dalam program java, berikut ini beberapa definisi polymorphism berdasarkan konteks penggunaan

- Polymorphism merupakan kemampuan objek-objek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.

- Polymorphism juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
- Polymorphism dapat berarti banyak bentuk, maksudnya yaitu kita dapat menimpa (override) suatu method, yang berasal dari parent class (super class) dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda (Method overriding)

Berikut ini contoh sederhana konsep polymorphism :

Vehicle.java :

```
class Vehicle {
    public void PropertiesDisplay() {
        System.out.println("general vehicle property");
    }
}
```

MotorCycle.java :

```
class MotorCycle extends Vehicle {

    public void PropertiesDisplay() {
        System.out.println("I have 500cc");
    }

    public static void main(String args[])
    {
        MotorCycle motor = new MotorCycle();

        motor.PropertiesDisplay();
    }
}
```

Bike.java :

```
class Bike extends Vehicle {

    public void PropertiesDisplay() {
        System.out.println("I do not have machine, running by gear");
    }

    public static void main(String args[])
    {
        Bike obj = new Bike();

        obj.PropertiesDisplay();
    }
}
```

PENJELASAN

Pada contoh di atas kita memiliki kelas generik untuk kendaraan dengan metode dasar PropertiesDisplay yang akan menampilkan property generic dari kendaraan.

Pada kelas turunan MotorCycle dan Bike metode PropertiesDisplay akan merespon dengan berbeda pada metode PropertiesDisplay. Pada kelas MotorCycle dan kelas Bike metode PropertiesDisplay akan dioverride sehingga melakukan cara berbeda untuk metode PropertiesDisplay.

7. Abstraction

Abstraksi adalah proses pengabstraksi atau menyembunyikan kerumitan dari suatu proses (dalam hal ini program). Dengan abstraksi maka kerumitan implementasi bisa disembunyikan dari user. Abstraksi bisa dicapai dengan 2 cara yaitu menggunakan abstract class dan menggunakan interface. Dengan interface, abstraksi bisa dicapai 100%.

Perbedaan antara abstraction dan encapsulation adalah tujuan dari kedua konsep tersebut. Abstraction bertujuan untuk menyembunyikan kerumitan proses sedangkan encapsulation bertujuan untuk membatasi akses ke atribut (variabel atau method) di dalam class.

Jenis abstraksi :

1. Abstraksi Data (Data Abstraction)
2. Abstraksi Kontrol (Control Abstraction)

Untuk implementasi konsep abstraction bisa dilakukan dengan membuat abstract class atau dengan menggunakan interface.

Abstraction dengan Abstract Class

Kelas yang dideklarasi dengan keyword “abstract” merupakan kelas abstract. Suatu kelas abstract bisa memiliki metode abstract dan concrete method. Metode abstract tidak memiliki body dan implementasi sedangkan metode concrete memiliki body dan implementasi. Berikut ini contoh abstraction dengan cara menggunakan abstract class.

Animal.java:

```
/*
Animal.java
example for course material
*/
public abstract class Animal {
    public abstract void Sound();
}
```

Cat.java :

```
/*
Cat.java
example for course material
*/
public class Cat extends Animal {
    public void Sound() {
        System.out.println("meow meow meow");
    }

    public static void main(String []args) {
        Cat kattie = new Cat();
        kattie.Sound();
    }
}
```

Dog.java :

```
/*
Dog.java
example for course material
*/
public class Dog extends Animal {
    public void Sound() {
        System.out.println("guk guk guk");
    }
    public static void main(String []args) {
        Dog doggy = new Dog();
        doggy.Sound();
    }
}
```

PENJELASAN

Pada contoh di atas kita memiliki kelas parent : Animal, dimana kelas tersebut dideklarasi sebagai abstract class. Dua kelas selanjutnya adalah Dog dan Cat dimana dog dan cat adalah inheritance dari kelas Animal. Pada class Animal kita gunakan metode abstract Sound untuk deklarasi suara hewan akan tetapi karena metode dideklarasi secara abstract maka tidak ada implementasi detail untuk metode Sound ini, karena masing masing child class : Cat dan Dog memiliki implementasi sendiri sendiri untuk metode sound ini jadi kerumitan implementasi metode sound tidak perlu dituliskan di kelas parent Animal sehingga cukup membuat abstract method Sound tanpa implementasi pada kelas Animal.

Abstraction dengan Interface

Selain menggunakan abstract class, abstraksi juga bisa dilakukan dengan menggunakan interface. Berikut ini contoh abstraction dengan menggunakan interface

Animal.java :

```
/*
Animal.java
example for course material
*/
public interface Animal {
    void Pee(int peetime);
    void Sound();
}
```

Dog.java :

```
/*
Dog.java
example for course material
*/
public class Dog implements Animal {
    int peetime = 0;

    public void Pee(int num) {
        peetime = num;
        System.out.println("pee time :" + peetime);
    }
}
```

```

public void Sound() {
    System.out.println("guk guk");
}

public static void main(String []args) {
    Dog doggie = new Dog();
    doggie.Sound();
    doggie.Pee(3);
}
}

```

Cat.java :

```

/*
Cat.java
example for course material
*/

public class Cat implements Animal {
    int peetime = 0;

    public void Pee(int num) {
        peetime = num + 1;
        System.out.println("pee time :" + peetime);
    }
    public void Sound() {
        System.out.println("meow meow");
    }
    public static void main(String []args) {
        Cat kattie = new Cat();
        kattie.Sound();
        kattie.Pee(3);
    }
}

```

PENJELASAN

Pada contoh di atas, kita memiliki interface dengan nama Animal. Selanjutnya kita membuat 2 kelas lainnya yang mengimplementasi interface Animal, yaitu Cat dan Dog.

Pada bagian interface terdapat 2 metode yaitu Pee dan Sound, di mana kedua metode di interface tidak memiliki badan (tidak memiliki implementasi). Implementasi detail untuk metode Pee dan metode Sound pada interface Animal didefinisikan masing masing di class Cat dan class Dog.

A.3. Good Programming Practice pada Java

Pada saat mempelajari suatu bahasa pemrograman, pastikan anda mencari tahu tentang good programming practice dan bad programming practice pada bahasa pemrograman tersebut. Ini akan meminimalisir kesalahan dalam pemrograman di kemudian hari.

Pada bagian ini, kita akan mempelajari contoh contoh good programming practice pada java. Berikut ini adalah contoh contoh good programming practice pada java :

1. Usahakan Membuat Class Kecil

Fokuskan untuk membuat class yang kecil tetapi fokus untuk melakukan pekerjaan yang sedikit sesuai fungsional class dan bisa bekerja dengan baik.

2. Usahakan Menamakan Metode Menjadi Lebih Mudah Dibaca.

Contoh :

```
public void Ares()
```

dan

```
public void AdvancedReasoning()
```

Penamaan metode pertama lebih singkat akan tetapi akan lebih sulit dipahami oleh developer lain dalam satu tim, Jika anda berada pada tim developer maka akan lebih baik memberikan penamaan yang mudah dibaca oleh anggota tim lain.

3. Usahakan Membuat Metode Kecil

Usahakan membuat suatu method fokus khusus hanya untuk 1 jenis pekerjaan, ini akan mempermudah pada saat debugging aplikasi dan untuk memahami jalanya aplikasi anda di kemudian hari.

4. Penggunaan Komentar

Komentar bisa dimanfaatkan untuk membuat source code Anda lebih mudah dimengerti ketika suatu saat Anda membaca source code Anda.

5. Gunakan Error Handling

Pada java, akan sangat baik jika anda memasang error handling pada rutin rutin yang berpotensi crash, error. Ini akan mempermudah pada saat proses debug aplikasi.

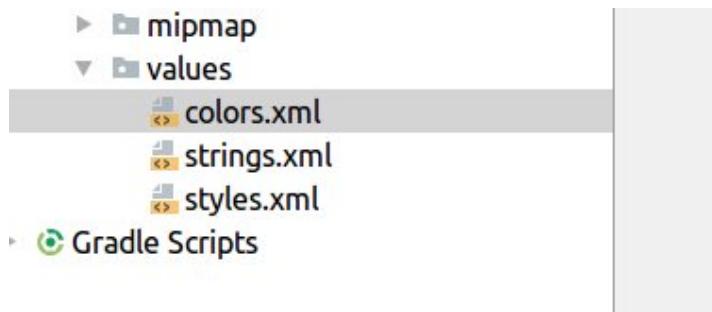
B. XML Pada Android Studio

Xml dirancang untuk menyimpan dan mengirimkan data. Berikut ini contoh xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
    <food>
        <name>Baju</name>
        <price>100</price>
        <description>
            Pakaian untuk manusia
        </description>
    </food>
    <food>
        <name>Celana</name>
        <price>50</price>
        <description>
            Bagian bawah pakaian
        </description>
    </food>
</breakfast_menu>
```

xml (Extensible Markup Language) pada android studio digunakan untuk mendefinisikan layout yang akan digunakan pada aplikasi android; menyimpan data berupa string maupun array; mendefinisikan komponen komponen di dalam layout; hingga mendefinisikan android manifest.

Berikut ini contoh penggunaan xml untuk mendefinisikan asset di android berupa : warna, string dan style:



Contoh mendefinisikan warna :

colors.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

Contoh mendefinisikan string:

strings.xml:

```
<resources>
    <string name="app_name">ToastSample</string>
</resources>
```

Contoh mendefinisikan styles:

styles.xml:

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

xml pada android studio bisa juga digunakan untuk mendefinisikan tata letak (layout) dari activity (satu layar penuh user interface).

C. View, ViewGroup dan Komponen View

C.1. View

View adalah komponen yang disediakan android untuk kita mendesign layout aplikasi. View bisa dianggap daerah kotak yang mana akan kita isi komponen komponen user interface di dalamnya.

Berikut ini beberapa komponen view yang sering digunakan pada saat pembuatan aplikasi android:

1. Button

Digunakan untuk view berupa tombol.

2. EditText

Digunakan untuk view berupa kotak teks yang bisa dierit

3. TextView

Merupakan elemen user interface yang digunakan untuk menampilkan teks

4. ImageView

Digunakan untuk menampilkan resource berupa image

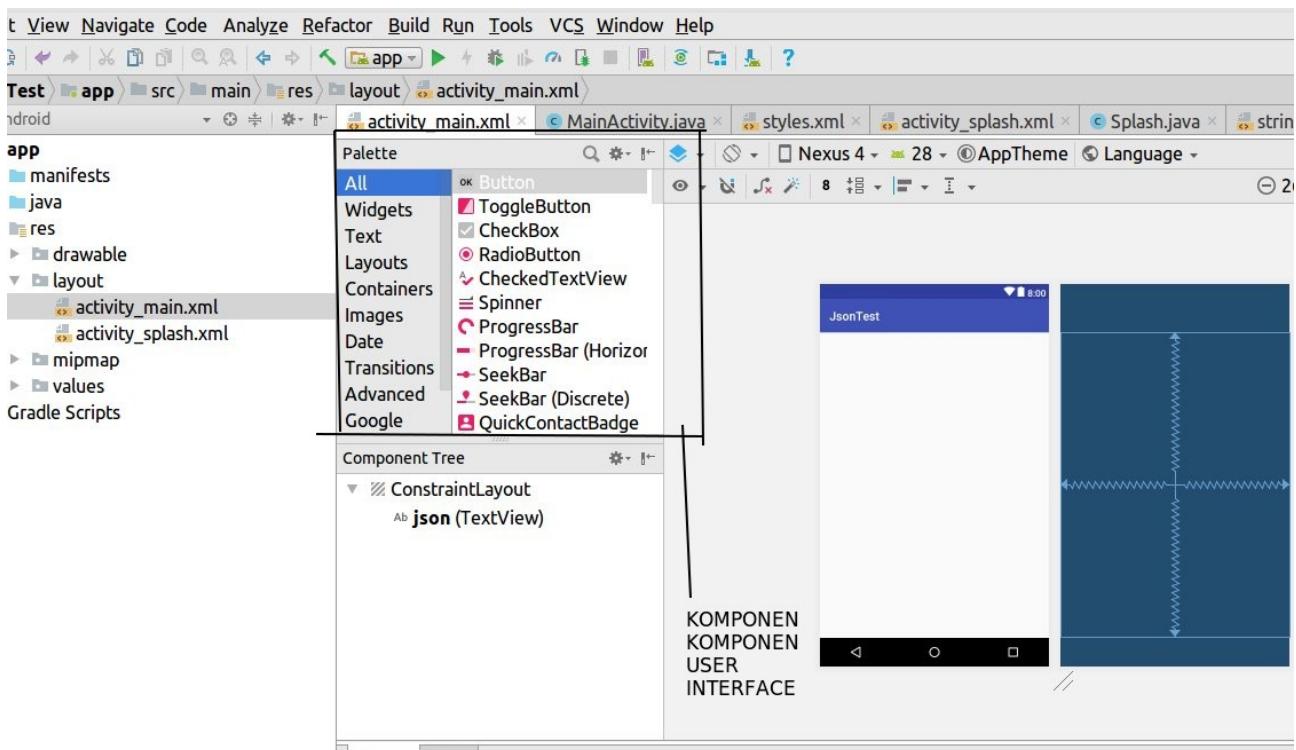
5. Space

Digunakan untuk memberikan area spase kosong pada layout

7. Checkbox

Digunakan untuk menampilkan pilihan centang

Untuk daftar lengkap bisa dicek di Android Studio



C.2. ViewGroup

ViewGroup adalah jenis view khusus yang bisa berisi komponen view lain di dalamnya. Beberapa contoh ViewGroup : LinearLayout, ScrollView, FrameLayout.

Suatu layout adalah subclass dari ViewGroup dengan tujuan khusus untuk mengorganisir dan memposisikan view-view lain di dalamnya.

Activity di Android menggunakan layout sebagai container untuk view lain di dalamnya, bahkan juga bisa berisi layout lain. Layout yang paling umum dipakai diantaranya adalah LinearLayout, RelativeLayout, dan FrameLayout.

Berikut ini penjelasan tentang beberapa tipe viewgroup layout yang sering dipakai :

1. LinearLayout

LinearLayout akan menampilkan dan memposisikan elemen di dalamnya dalam satu arah, secara horizontal atau vertikal. Arahnya dapat diatur melalui atribut android:orientation.

Untuk lebih jelasnya, berikut ini gambaran LinearLayout dengan orientation horizontal



LinearLayout dengan orientation vertical



2. RelativeLayout

Pada RelativeLayout, kita memposisikan view di dalamnya relatif terhadap view lain atau si parent view-nya.

RelativeLayout adalah jenis layout yang memiliki karakteristik dalam menempatkan view secara relatif. Relatif disini berarti posisi dari setiap view bergantung kepada view yang lain.



3. ConstraintLayout

ConstraintLayout memungkinkan kita untuk mengatur posisi objek menggunakan *constraint* dengan objek lainnya sesuai keinginan kita. Ini hampir mirip dengan RelativeLayout karena semua tampilan ditata berdasarkan hubungan antara satu objek dengan yang lain, tetapi lebih fleksibel daripada RelativeLayout dan lebih mudah digunakan dengan Editor Layout Android Studio. Dengan fleksibilitas pada constraint layout, kita tidak perlu menggunakan nested layout saat mendesign layout yang kompleks. Constraint Layout lebih fleksibel dan lebih mudah didesign melalui layout design editor karena dengan constraint layout, kita bisa menambahkan berbagai constraint connection pada suatu view. Pada constraint layout, kita perlu membuat minimal 2 constraint / 2 connection handle pada suatu komponen.

Berikut ini beberapa perbedaan fitur pada constraint layout dan relative layout:

1. Pada constraint layout terdapat beberapa tambahan atribut dimana kita bisa mengatur tata letak suatu view / komponen secara absolute pada layar berdasarkan koordinat x dan y, melalui atribut :
koordinat x bisa diatur dengan :

tools:layout_editor_absoluteX

koordinat y bisa diatur dengan :

tools:layout_editor_absoluteY

di mana nilai dari x dan y bisa berupa dp. Fitur tersebut tidak ada pada relative layout.

2. Adanya Constraint Bias

Fungsi bias ini digunakan untuk penyesuaian posisi komponen ketika suatu komponen memiliki 2 tali constraint pada sisi kiri dan kanan atau pada sisi atas dan bawah, bias ini digunakan untuk adjustment position.

3. Pada constraint layout, saat menggunakan design editor, kita bisa membuat constraint pada masing masing view / komponen.

Ada 2 jenis constraint yang bisa kita atur :

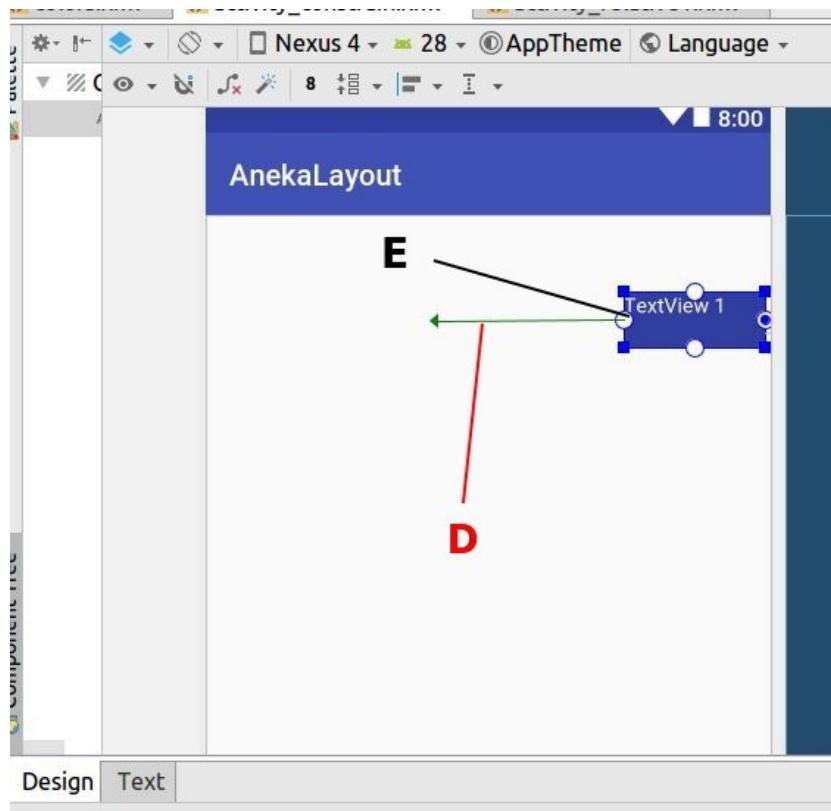
1. Constraint (connection handle) untuk pengaturan tata letak

Pada constraint layout, di mana minimal kita harus membuat 2 constraint untuk koneksi dengan komponen lain. Untuk mengatur constraint ini, handlenya terdapat pada bagian bagian samping dari suatu objek

2. Constraint untuk pengaturan ukuran (resize handle)

Pada constraint layout terdapat handle untuk resize sama seperti pada layout lainnya.

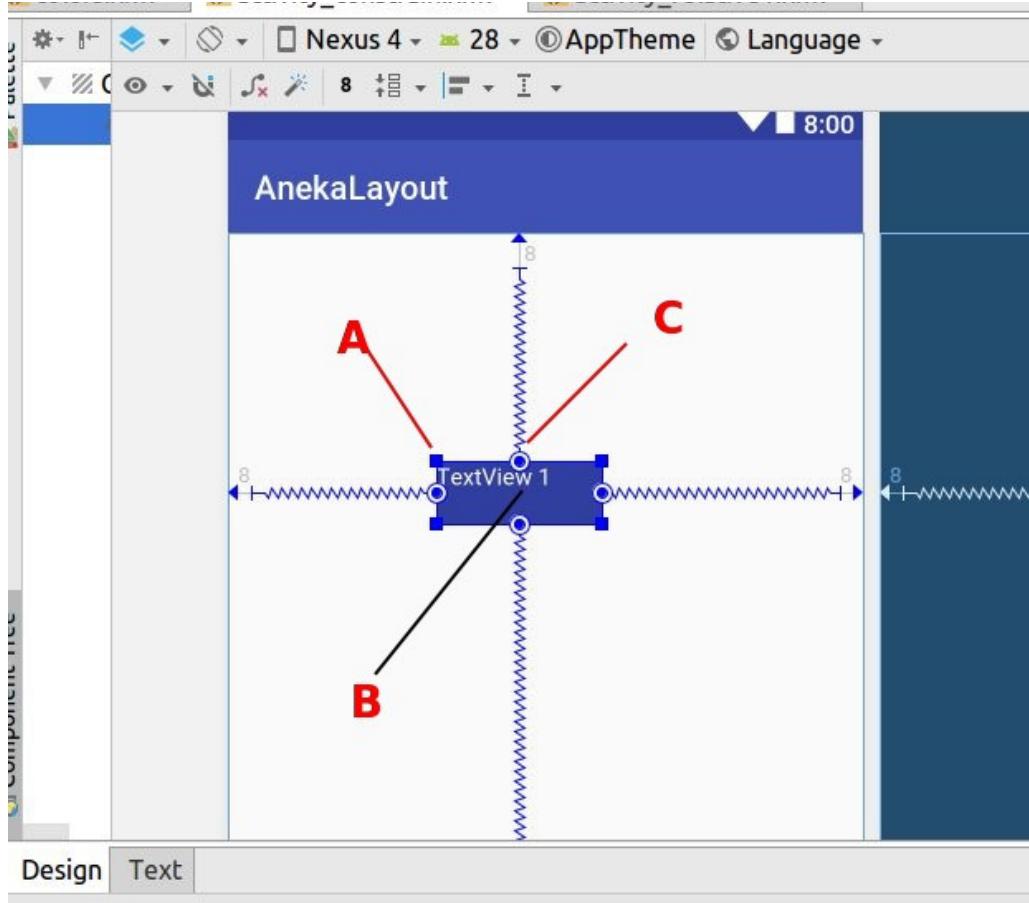
Untuk lebih jelasnya perhatikan design visual berikut ini, pada gambar di bawah ini, kita akan membuat tali constraint untuk patokan posisi komponen pada layout, pada contoh di bawah ini, pada akhirnya kita akan membuat 4 tali constraint untuk masing masing sisi komponen.



Misal kita menaruh 1 komponen berupa textView pada design visual

E : merupakan handle yang berguna untuk membuat constraint, di mana pada textView di atas terdapat 4 handle pada masing masing sisi untuk membuat constraint (tali), minimal kita harus membuat 2 constraint untuk pengaturan posisi komponen

D : seret dan tarik tali ke salah satu patokan, misal ke pinggir layout (bisa juga ke komponen lain jika ada).



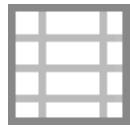
A : Digunakan untuk meresize ukuran komponen

B : Klik dan seret untuk pengaturan posisi komponen, pastikan sebelumnya sudah membuat minimal 2 constraint di salah satu sisi komponen agar komponen bisa dirender dengan benar.

C : jika sudah ada constraint sebelumnya, maka jika diklik pada titik handle yang sudah ada constraint tersebut akan hilang.

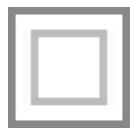
4. TableLayout

Table Layout adalah Layout yang digunakan untuk membangun user interface (tampilan antar muka) aplikasi android dengan berdasarkan Baris dan Kolom. Layout ini digunakan untuk keperluan tertentu saja karena kebanyakan pembangunan tampilan antar muka lebih sering memanfaatkan Relative Layout dan Linear Layout.



5. FrameLayout

FrameLayout adalah suatu wadah di layar yang bisa kita gunakan untuk menampilkan satu view saja. View yang kita tambahkan ke FrameLayout akan selalu di-anchor ke pojok kiri atas dari layout.



C.3. Atribut View

Berikut ini beberapa atribut yang sering digunakan pada definisi suatu View :

android:id

id dari view, id ini adalah unik untuk masing masing komponen. Suatu id selalu diawali dengan @+.

Tanda @+ digunakan saat definisi awal dari suatu komponen. Tanda + digunakan untuk memberikan informasi pada android sdk aapt (android packaging tool) bahwa komponen ini adalah resource baru.

android:gravity

Digunakan untuk posisi layout pada bidang x-y. Atribut ini biasa digunakan pada parent View. Property dari gravity bisa berupa : center, center_vertical, center_horizontal, left, right, top, bottom, dll.

android:ignoreGravity

Dengan attribut ini maka suatu view tidak akan dipengaruhi oleh android gravity.

android:layout_gravity

Atribut ini digunakan biasa digunakan pada ViewGroup (child view). Gambaran Perbedaan android:layout_gravity dan android:gravity adalah :

- atribut android:layout_gravity digunakan oleh child view di dalam suatu parent view , atribut ini akan menginformasikan parent view bagaimana child view harus diposisikan di dalam parent view.

- atribut android:gravity digunakan oleh parent view, di mana atribut ini akan menginformasikan child view untuk posisinya masing masing di dalam parent view.

android:layout_width

Ukuran width dari layout. Properti yang digunakan adalah : match_parent dan wrap_content.

android:layout_height

Ukuran tinggi dari layout.

android:layout_marginTop

Space ekstra pada atas view

android:layout_marginBottom

Space ekstra pada bawah view

android:layout_marginLeft

Space ekstra pada kiri view

android:layout_marginRight

Space ekstra pada kana view

android:layout_weight (Digunakan pada LinearLayout)

Properti ini digunakan untuk presisi pembagian jatah layar untuk suatu view.

android:paddingLeft

Digunakan untuk padding kiri di dalam suatu view

android:paddingRight

Digunakan untuk padding kanan di dalam suatu view

android:paddingTop

Digunakan untuk padding atas di dalam suatu view

android:paddingBottom

Digunakan untuk padding bawah di dalam suatu view

C.4. Dimensi Pada Design di Android

dp

Density independent pixel. Dengan menggunakan ukuran dimensi ini maka ukuran akan menjadi proporsional untuk berbagai layar berbeda.



1 dp berukuran 160 pixel pada layar MDPI. Pada layar dengan density yang lebih besar maka jumlah pixel akan lebih banyak. Jadi bisa kita bilang ukuran dp adalah proporsional menyesuaikan ukuran density layar.

sp

Scale independent pixel. Sp memiliki karakteristik yang sama dengan dp di mana ukuran ini proporsional sesuai ukuran layar, akan tetapi ukuran ini diatur oleh ukuran font sesuai preferensi user sehingga ukuran ini cocok digunakan untuk ukuran font. Gunakan sp hanya untuk font dan dp untuk komponen lainnya selain font !

pt

Points. Ukuran 1/72 dari 1 inch.

px

Pixels. Ukuran aktual berupa pixel dari ukuran layar.

mm

Milimeter. Ukuran aktual berupa milimeter dari ukuran layar

in

Inches. Ukuran aktual berupa inch dari ukuran layar.

D. Contoh Praktek Mendesign Layout

Setelah di atas kita memahami berbagai teori tentang view di android, pada kesempatan ini kita akan langsung praktik untuk menguji berbagai konsep yang telah kita pelajari sebelumnya.

Beberapa trik yang akan dipelajari :

D.1. Membuat Background Transparan

Sebelumnya buat proyek baru dengan nama Design. Proyek ini akan digunakan untuk menguji coba beberapa trik. Trik pertama adalah membuat warna background transparant terhadap background di belakangnya.

Pada contoh kali ini kita akan menggunakan background wallpaper dengan nama file wallpaper.jpg



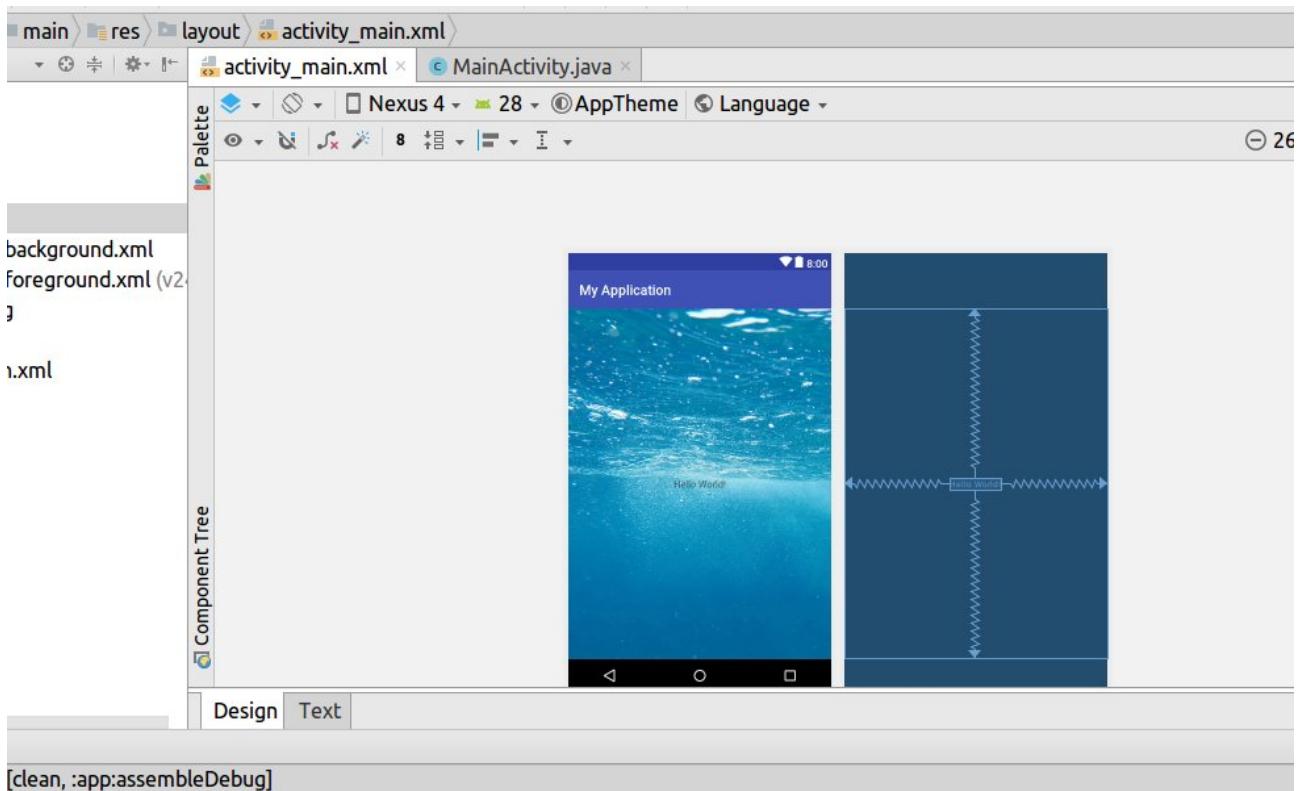
Simpan file wallpaper.jpg di folder drawable, lalu tambahkan baris ini pada layout :

android:background="@drawable/wallpaper"

The screenshot shows the Android Studio interface. The top menu bar includes options like Design, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The toolbar below has various icons for file operations. The left sidebar shows the project structure under the app module: manifests, java, res (with drawable, layout, mipmap, and values folders). The layout section contains activity_main.xml. The main editor window displays the XML code for activity_main.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.ringlayer.myapplication.MainActivity"
8     android:background="@drawable/wallpaper">
9
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13    </TextView>
```

Jika berhasil, maka di tampilan layout akan berubah :



Jika diperhatikan penggantian background menjadi image di atas, menyebabkan textView menjadi tidak terlihat.

Selanjutnya kita akan coba membuat background berwarna transparant untuk textView

Selanjutnya tambahkan beberapa baris pada komponen textView:

```
android:background="#70ffffffff"
android:padding="10dp"
android:textColor="@color/colorPrimary"
android:textStyle="bold"
```

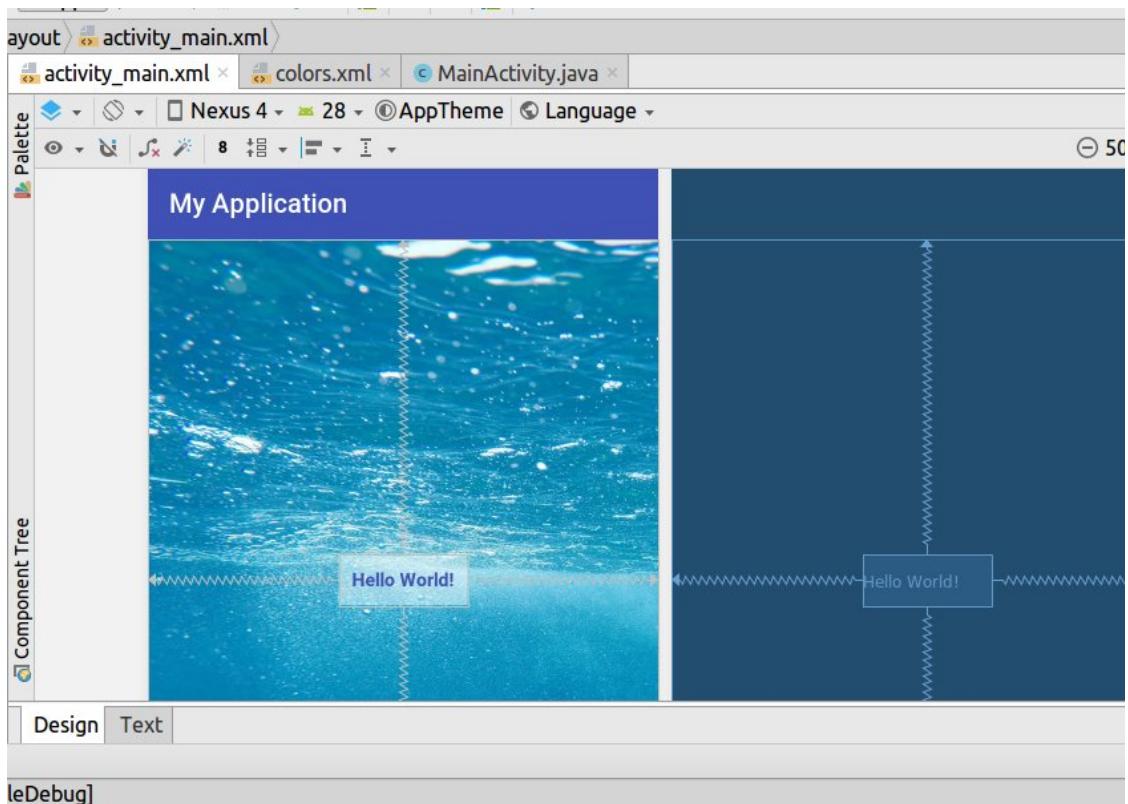
The screenshot shows the Android Studio interface with the XML editor open. The file is `activity_main.xml`. The code defines a `ConstraintLayout` containing a single `TextView` with the text "Hello World!". The `TextView` has a background color of `#70ffffff`, padding of `10dp`, primary text color, and bold style. It is constrained to the parent with wrap content dimensions and positioned centrally.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.myapplication.MainActivity"
    android:background="@drawable/wallpaper">

    <TextView
        android:background="#70ffffff"
        android:padding="10dp"
        android:textColor="@color/colorPrimary"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

Jika berhasil, maka tampilan akan menjadi :



Pada tampilan di atas terlihat `textView` hello world menjadi memiliki background transparan dan bisa terlihat lebih jelas.

Penjelasan

android:background="#70ffffff"

Baris ini digunakan untuk mengganti warna background menjadi transparant, di mana kita buat transparan hingga 70% dan warna dasar putih (#ffffff)

android:padding="10dp"

Baris ini digunakan untuk menambahkan padding sejumlah 10dp

android:textColor="@color/colorPrimary"

Baris ini digunakan untuk mengganti warna teks

android:textStyle="bold"

Baris ini digunakan untuk menebalkan teks

D.2. match_parent, wrap_content

match_parent

Dengan match_parent, misal jika dispesifikasi pada width akan menyebabkan ukuran komponen menjadi sebesar ukuran parent.

Untuk lebih jelasnya buka kembali proyek “Design” sebelumnya.

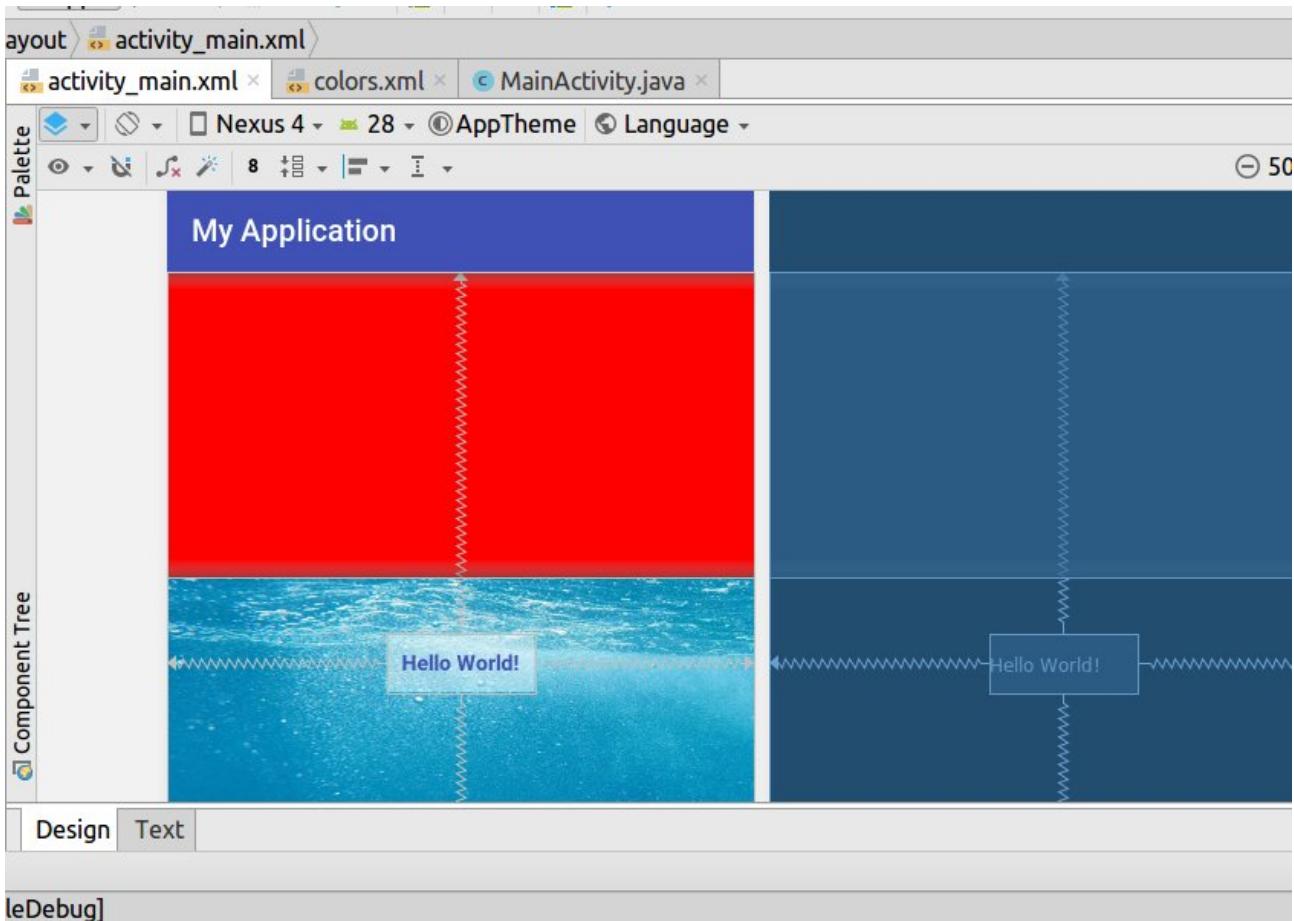
Di atas textview, tambahkan baris ini:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:orientation="vertical">
    <View
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#FF0000" />
</LinearLayout>
```

Pada contoh di atas, kita akan fokus pada atribut

android:layout_height="match_parent"

Pemberian atribut layout_height match_parent pada komponen view akan menyebabkan tinggi dari view sama dengan tinggi dari parentnya, di mana parentnya adalah LinearLayout dengan tinggi 200dp.



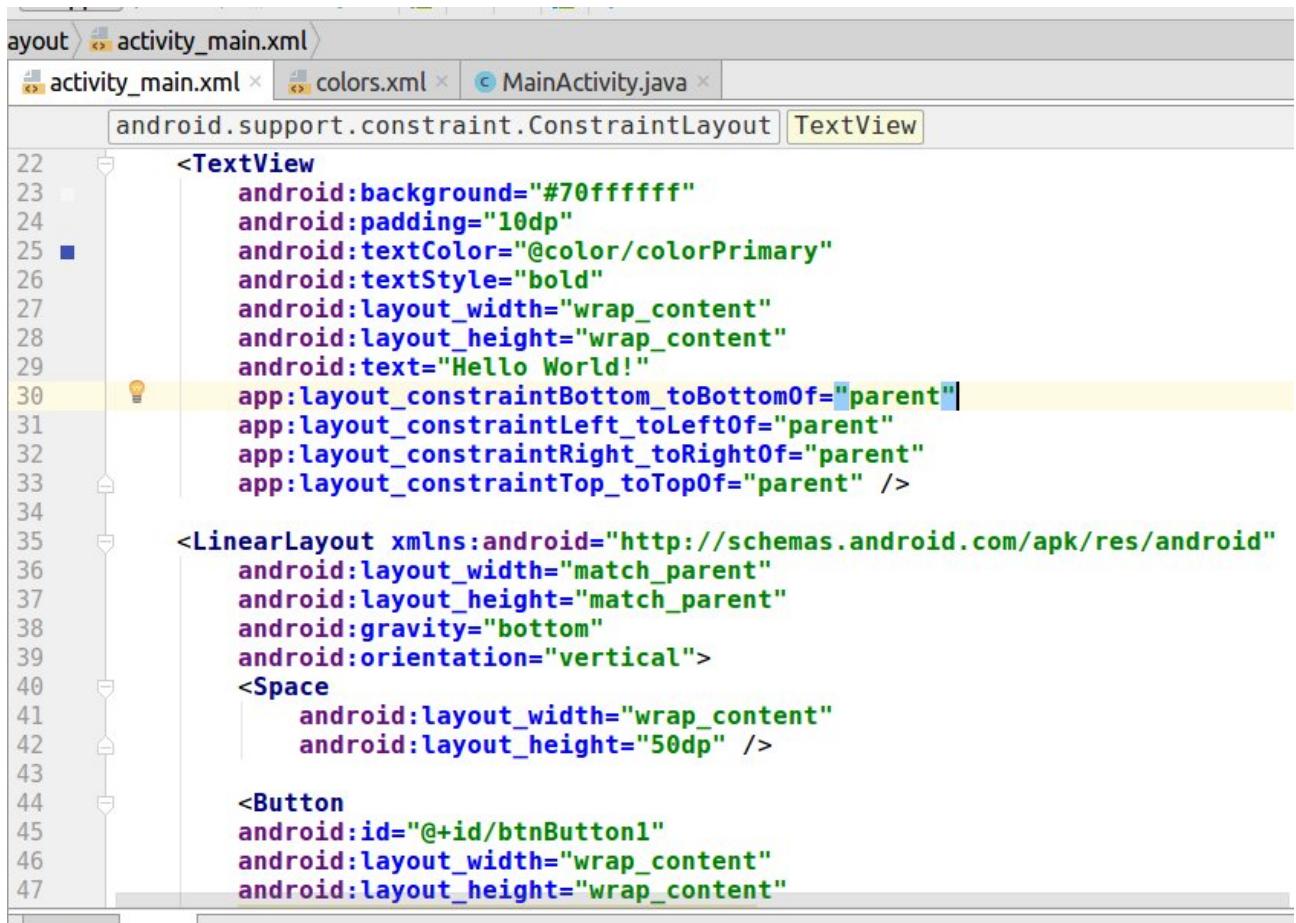
wrap_content

Wrap content adalah nilai atribut yang digunakan untuk menyesuaikan ukuran lebar dan tinggi view berdasarkan konten atau objek yang ada di dalamnya.

Untuk lebih jelasnya, buka kembali proyek Design, kita tambahkan beberapa baris pada xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="bottom"
    android:orientation="vertical">
    <Space
        android:layout_width="wrap_content"
        android:layout_height="50dp" />
    <Button
        android:id="@+id/btnButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="dengan wrap content"/>
    <Button
        android:id="@+id/btnButton2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="match parent"/>
</LinearLayout>
```

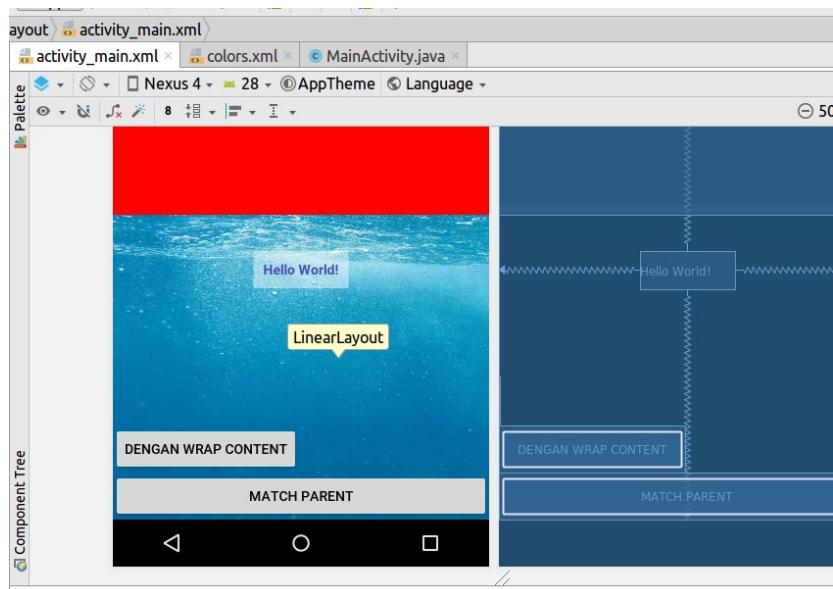
Baris tersebut kita tambahkan setelah textView yang sebelumnya.



The screenshot shows the Android Studio interface with the XML code for `activity_main.xml`. The code defines a `ConstraintLayout` containing a `TextView` and a `LinearLayout`. The `TextView` has a background color of #70ffffff and contains the text "Hello World!". The `LinearLayout` contains a `Space` and a `Button` with ID `@+id/btnButton1`.

```
22 <TextView
23     android:background="#70fffffff"
24     android:padding="10dp"
25     android:textColor="@color/colorPrimary"
26     android:textStyle="bold"
27     android:layout_width="wrap_content"
28     android:layout_height="wrap_content"
29     android:text="Hello World!"
30     app:layout_constraintBottom_toBottomOf="parent"
31     app:layout_constraintLeft_toLeftOf="parent"
32     app:layout_constraintRight_toRightOf="parent"
33     app:layout_constraintTop_toTopOf="parent" />
34
35 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
36     android:layout_width="match_parent"
37     android:layout_height="match_parent"
38     android:gravity="bottom"
39     android:orientation="vertical">
40     <Space
41         android:layout_width="wrap_content"
42         android:layout_height="50dp" />
43
44     <Button
45         android:id="@+id/btnButton1"
46         android:layout_width="wrap_content"
47         android:layout_height="wrap_content"
```

Hasilnya adalah :

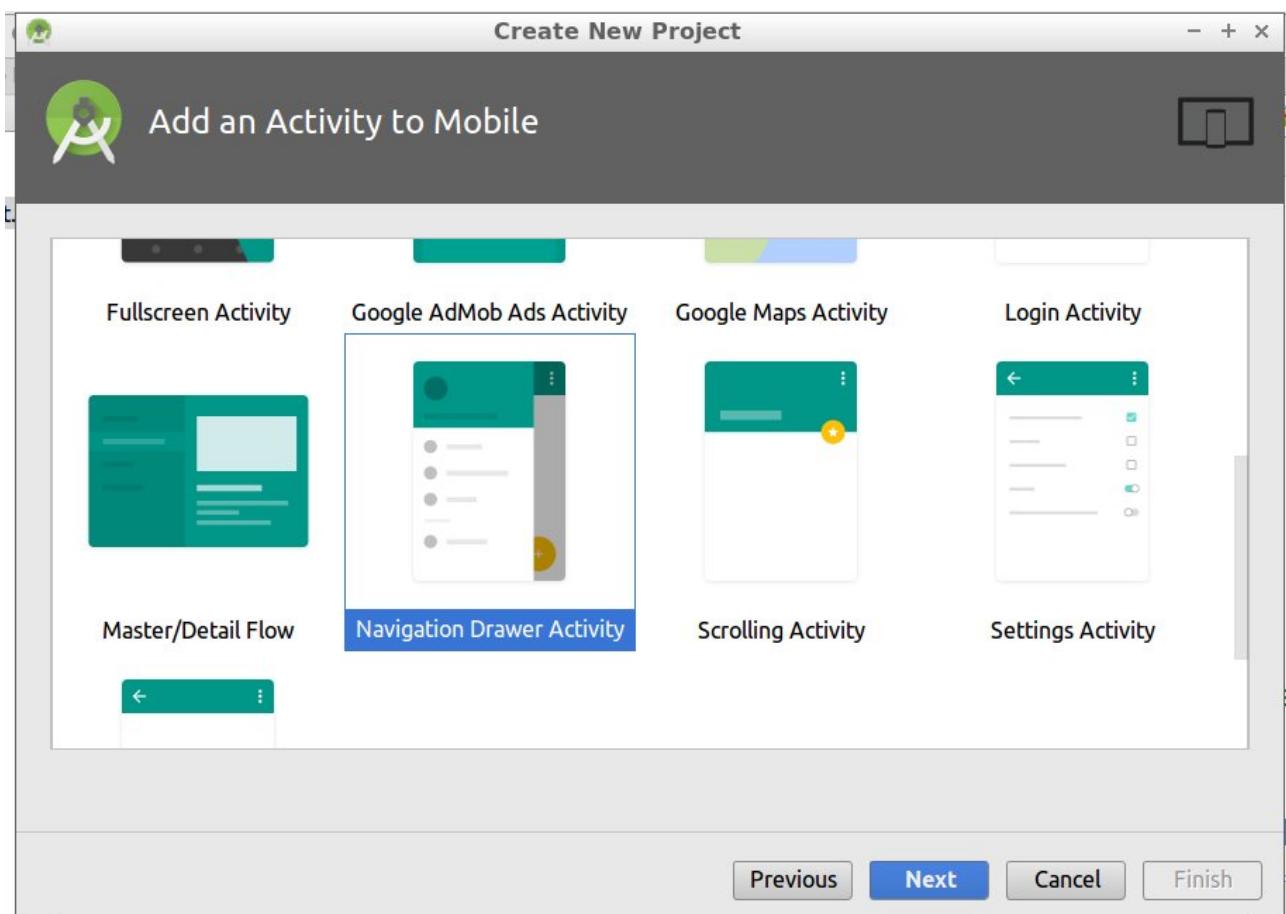


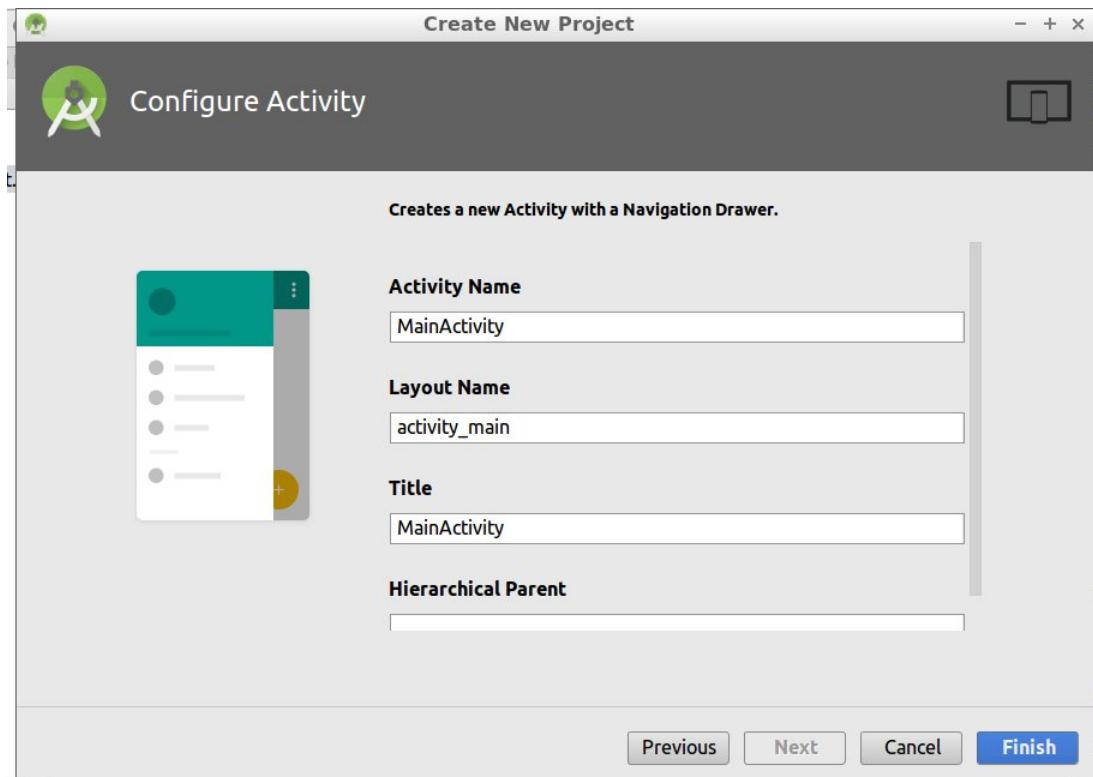
terlihat dengan `wrap_content` pada Button maka lebar elemen menyesuaikan dengan isi dari komponen.

D.3. Membuat Drawer Layout

Selanjutnya kita akan mencoba membuat aplikasi dengan drawer layout. Buat proyek baru dengan nama DrawerLt.

Pada saat pemilihan activity, pilih “Navigation Drawer Activity”





Setelah proyek siap, buka MainActivity.java, lalu buang baris baris ini :

```
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});
```

Setelah dibuang :

The screenshot shows the Android Studio interface with the file `MainActivity.java` open. The code implements a `NavigationView.OnNavigationItemSelectedListener`. It initializes a toolbar and drawer layout, sets up a drawer toggle, and configures the navigation view. The `onBackPressed()` method is overridden to close the drawer if it is open.

```
com>ringlayer>drawerlt>MainActivity>
MainActivity.java x activity_main.xml x app_bar_main.xml x content_main.xml x nav_header_main.xml x
MainActivity onCreate()
1 package com.ringlayer.drawerlt;
2
3 import ...
15
16 public class MainActivity extends AppCompatActivity
17     implements NavigationView.OnNavigationItemSelectedListener {
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
24         setSupportActionBar(toolbar);
25
26         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
27         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
28             activity, this, drawer, toolbar, "Open navigation drawer", "Close navigation d
29         drawer.addDrawerListener(toggle);
30         toggle.syncState();
31
32         NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
33         navigationView.setNavigationItemSelectedListener(this);
34     }
35
36     @Override
37     public void onBackPressed() {
38         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
39         if (drawer.isDrawerOpen(GravityCompat.START)) {
```

Selanjutnya buang baris baris ini :

```
else if (id == R.id.nav_manage) {
} else if (id == R.id.nav_share) {
} else if (id == R.id.nav_send) {
```

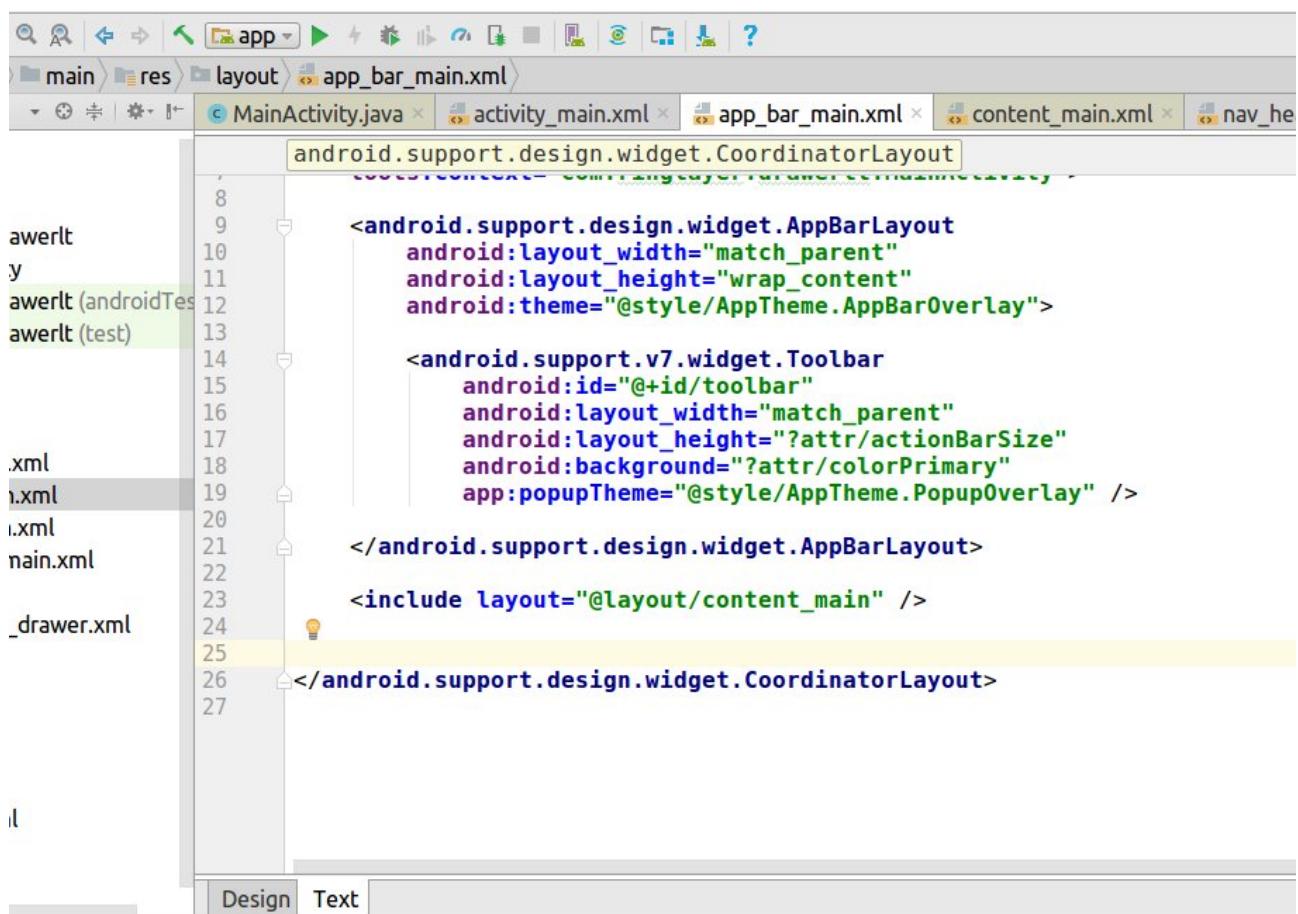
The screenshot shows the `MainActivity.java` code with a large portion of the `onNavigationItemSelected()` method body redacted in blue. The redacted area covers lines 73 through 85, which correspond to the `else if` statements listed above. The code continues below the redacted area with the drawer closing logic.

```
ANDROID/DrawerLt/app/src/main/java/com/ringlayer/drawerlt/MainActivity.java
MainActivity x activity_main_drawer.xml x fragment_camera.xml x CameraFragment.java x
MainActivity onNavigationItemSelected()
73
74     if (id == R.id.nav_camera) {
75     } else if (id == R.id.nav_gallery) {
76
77     } else if (id == R.id.nav_slideshow) {
78
79     } else if (id == R.id.nav_manage) {
80
81     } else if (id == R.id.nav_share) {
82
83     } else if (id == R.id.nav_send) {
84
85 }
86
87     DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout)
88     drawer.closeDrawer(GravityCompat.START);
89     return true;
90 }
```

Selanjutnya buka app_bar_main.xml pada layout, dan buang komponen android.support.design.widget.FloatingActionButton :

```
<android.support.design.widget.FloatingActionButton  
    android:id="@+id/fab"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="bottom|end"  
    android:layout_margin="@dimen/fab_margin"  
    app:srcCompat="@android:drawable/ic_dialog_email" />
```

Setelah dibuang :



Selanjutnya buka activity_main_drawer.xml, kita akan membuang baris baris ini :

```
NDROID/DrawerLt/app/src/main/java/com/ringlayer/drawerlt/MainActivity.java
>MainActivity.java activity_main_drawer.xml
```

```
menu
15 <item
16     android:id="@+id/nav_slideshow"
17     android:icon="@drawable/ic_menu_slideshow"
18     android:title="Slideshow" />
19 <item
20     android:id="@+id/nav_manage"
21     android:icon="@drawable/ic_menu_manage"
22     android:title="Tools" />
23 </group>
24
25 <item android:title="Communicate">
26     <menu>
27         <item
28             android:id="@+id/nav_share"
29             android:icon="@drawable/ic_menu_share"
30             android:title="Share" />
31         <item
32             android:id="@+id/nav_send"
33             android:icon="@drawable/ic_menu_send"
34             android:title="Send" />
35     </menu>
36 </item>
37 </menu>
38
39
```

Design Text TODO Event Log Gradle Console

Baris yang dibuang :

```
<item android:title="Communicate">
    <menu>
        <item
            android:id="@+id/nav_share"
            android:icon="@drawable/ic_menu_share"
            android:title="Share" />
        <item
            android:id="@+id/nav_send"
            android:icon="@drawable/ic_menu_send"
            android:title="Send" />
    </menu>
</item>
```

Selanjutnya, kita akan membuang baris baris ini :

```
menu > activity_main_drawer.xml
MainActivity.java x activity_main_drawer.xml x

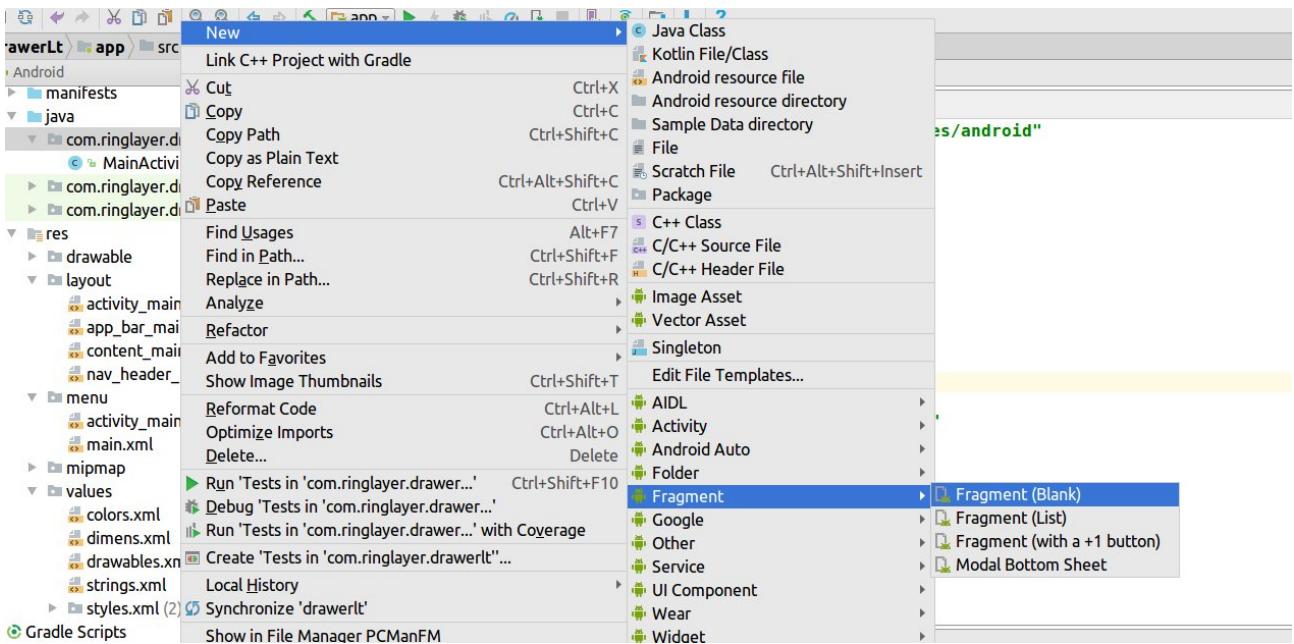
menu group
6   <group android:checkableBehavior="single">
7     <item
8       android:id="@+id/nav_camera"
9       android:icon="@drawable/ic_menu_camera"
10      android:title="Import" />
11
12     <item
13       android:id="@+id/nav_gallery"
14       android:icon="@drawable/ic_menu_gallery"
15      android:title="Gallery" />
16
17     <item
18       android:id="@+id/nav_slideshow"
19       android:icon="@drawable/ic_menu_slideshow"
20      android:title="Slideshow" />
21
22     <item
23       android:id="@+id/nav_manage"
24       android:icon="@drawable/ic_menu_manage"
25      android:title="Tools" />
26   </group>
27
28 </menu>
```

Buang baris ini :

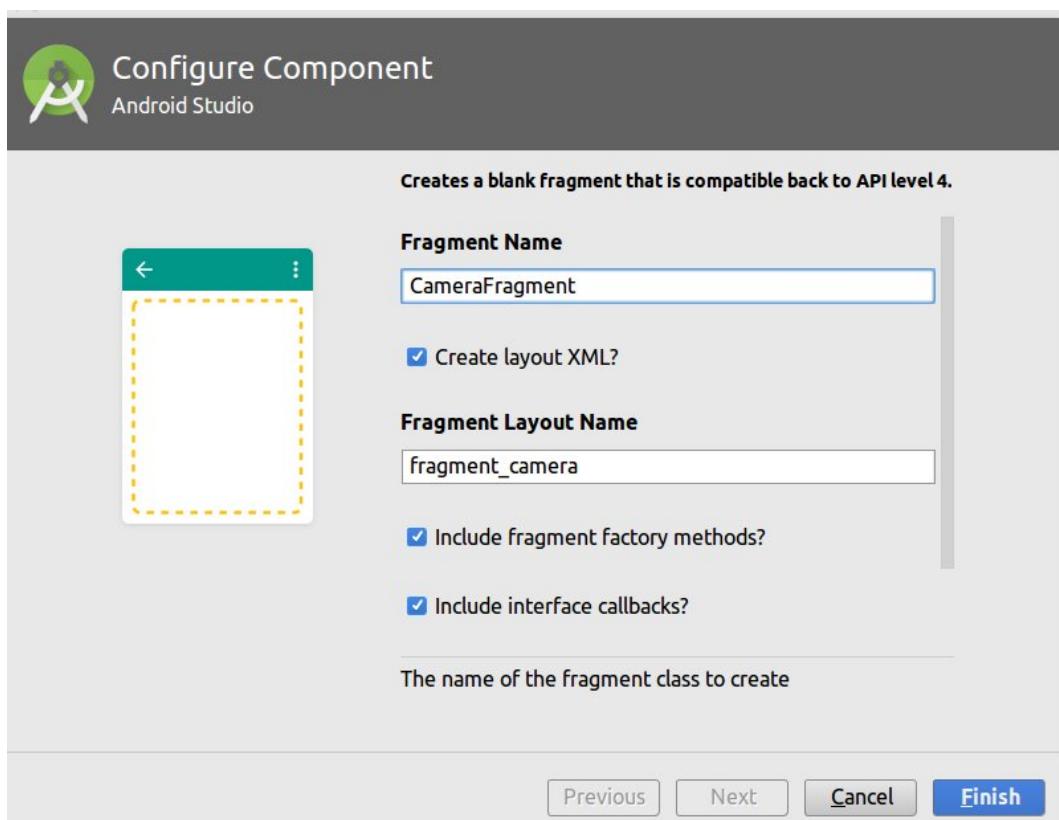
```
<item
  android:id="@+id/nav_manage"
  android:icon="@drawable/ic_menu_manage"
  android:title="Tools" />
```

Selanjutnya kita akan menambahkan 3 fragment kosong.

Buat blank fragment baru :



Beri nama fragment CameraFragment

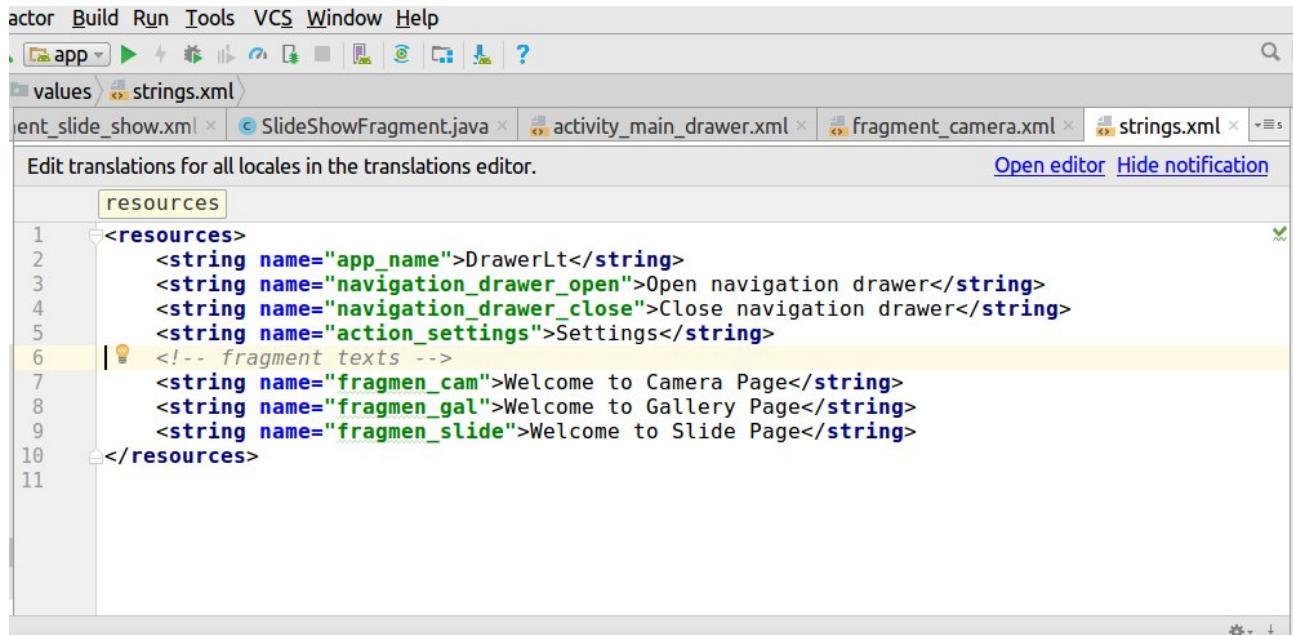


Selanjutnya tambahkan 2 fragment lagi, masing masing dengan nama fragment : SlideShowFragment dan GalleryFragment.

Selanjutnya buka strings.xml pada folder values.

Tambahkan beberapa baris untuk string yang akan ditampilkan pada halaman fragment :

```
<!-- fragment texts -->
<string name="fragmen_cam">Welcome to Camera Page</string>
<string name="fragmen_gal">Welcome to Gallery Page</string>
<string name="fragmen_slide">Welcome to Slide Page</string>
```



Selanjutnya buka ketiga xml untuk fragment :

fragment_slide_show.xml :

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.drawerlt.SlideShowFragment">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/fragmen_slide" />
</FrameLayout>
```

fragment_gallery.xml :

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.drawerlt.GalleryFragment">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/fragmen_gal" />
```

```
</FrameLayout>
```

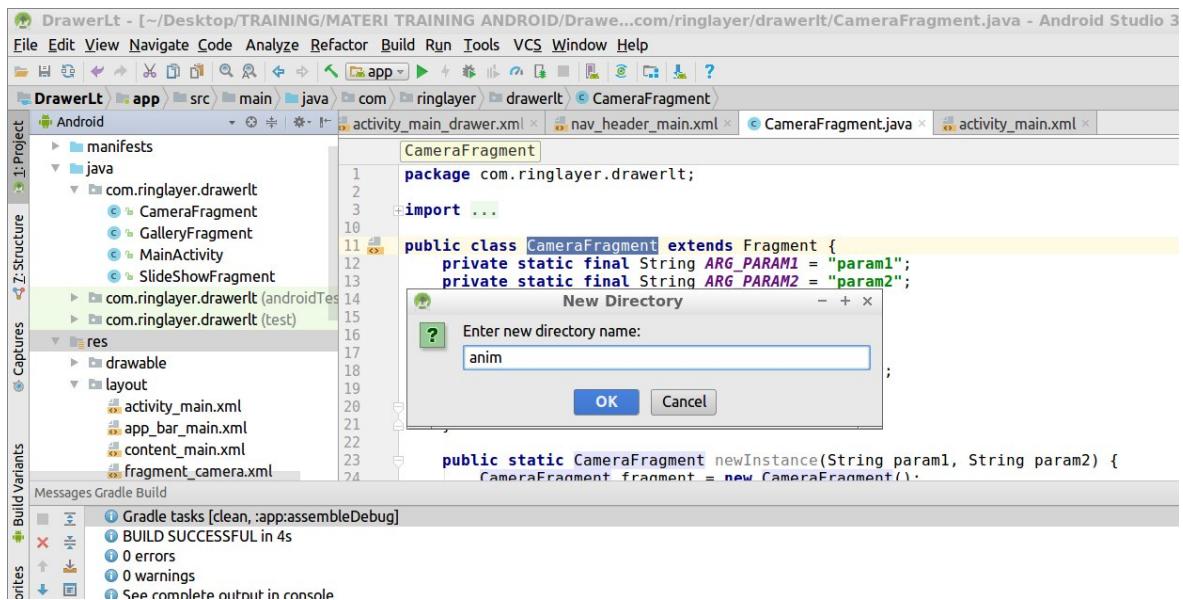
fragment_camera.xml :

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.drawerlt.CameraFragment">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/fragmen_cam" />
</FrameLayout>
```

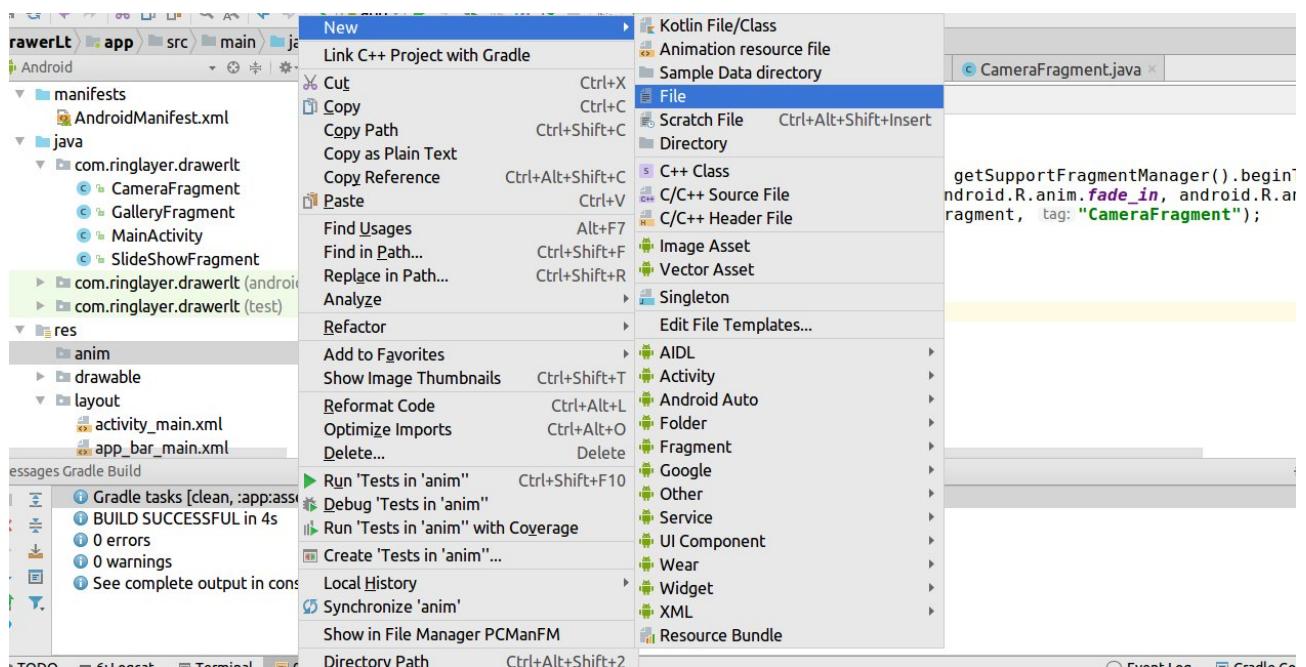
Selanjutnya buka content_main.xml, ganti TextView menjadi FrameLayout dengan id : “frame”. Berikut ini isi content_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.ringlayer.drawerlt.MainActivity"
    tools:showIn="@layout/app_bar_main">
    <FrameLayout
        android:id="@+id/frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
    </FrameLayout>
</android.support.constraint.ConstraintLayout>
```

Selanjutnya kita akan menambahkan efek fade in dan fade out saat pergantian fragment di drawer layout ini, buat direktori baru dengan nama anim pada res :



Selanjutnya pada folder anim, klik kanan, buat file baru dengan nama fade_in.xml dan fade_out.xml



Berikut ini isi fade_in.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha
        android:duration="2000"
        android:fromAlpha="0.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="1.0" />
</set>
```

Berikut ini isi fade_out.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha
        android:duration="2000"
        android:fromAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="0.0" />
</set>
```

Penjelasan

fade_in.xml dan fade_out.xml berguna untuk memberikan efek transisi selama 2 detik saat pergantian fragment:

android:duration="2000"

Selanjutnya buka MainActivity.java, tambahkan rutin untuk pergantian fragment dengan cara mereplace fragment dengan id "frame" yang tadi telah kita siapkan, kita akan menambahkan rutin ini di bagian method "onNavigationItemSelected"

```
if (id == R.id.nav_camera) {
    Fragment fragment = new CameraFragment();
    FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
} else if (id == R.id.nav_gallery) {
    Fragment fragment = new GalleryFragment();
    FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
} else if (id == R.id.nav_slideshow) {
    Fragment fragment = new SlideShowFragment();
    FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
}
```

The screenshot shows the Android Studio interface with the code editor open to MainActivity.java. The code implements logic to replace fragments based on selected navigation items. It uses custom animations for fading in and out.

```
if (id == R.id.nav_camera) {
    Fragment fragment = new CameraFragment();
    FragmentTransaction fragmentTransaction = getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in, android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
} else if (id == R.id.nav_gallery) {
    Fragment fragment = new GalleryFragment();
    FragmentTransaction fragmentTransaction = getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in, android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
} else if (id == R.id.nav_slideshow) {
    Fragment fragment = new SlideShowFragment();
    FragmentTransaction fragmentTransaction = getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in, android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
}
```

Penjelasan

```
if (id == R.id.nav_camera) {
    Fragment fragment = new CameraFragment();
    FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
android.R.anim.fade_out);
    fragmentTransaction.replace(R.id.frame, fragment, "CameraFragment");
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
}
```

Sebelumnya pada activity_main_drawer.xml, kita telah menyiapkan 3 item menu masing masing dengan id : nav_camera, nav_gallery dan nav_slideshow

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_camera"
            android:icon="@drawable/ic_menu_camera"
            android:title="Import" />
        <item
            android:id="@+id/nav_gallery"
            android:icon="@drawable/ic_menu_gallery"
            android:title="Gallery" />
        <item
            android:id="@+id/nav_slideshow"
            android:icon="@drawable/ic_menu_slideshow"
            android:title="Slideshow" />
    </group>
</menu>
```

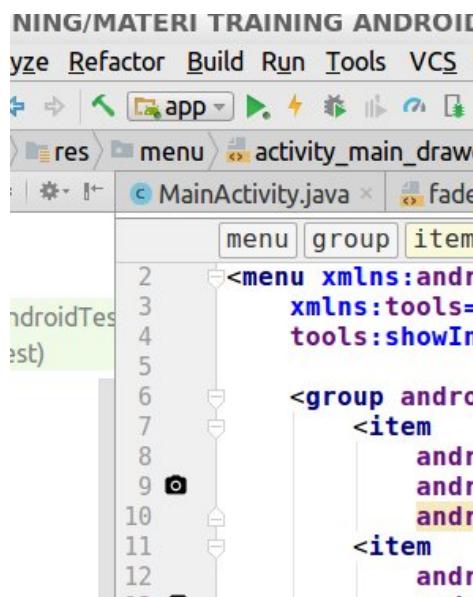
Jika didapati user mengklik pada menu drawer dengan id : “nav_camera” maka akan dilakukan transaksi pergantian fragment untuk mereplace layout framelayout dengan id : “frame” dengan layout untuk CameraFragment yang telah kita siapkan pada fragment_camera.xml.

Rutin ini :

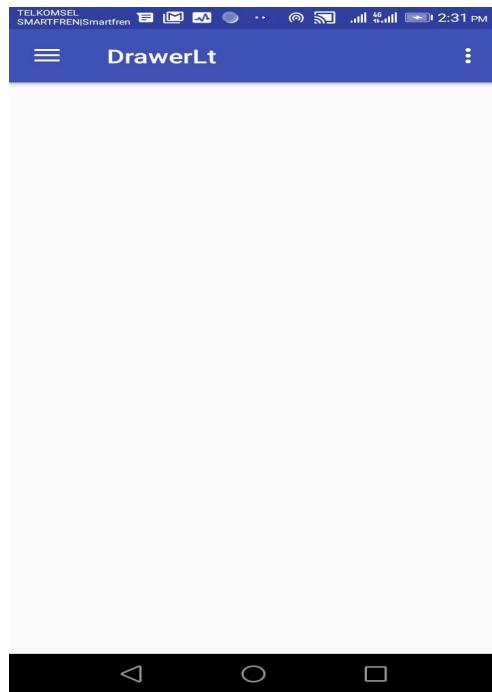
```
fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
    android.R.anim.fade_out);
```

Berguna untuk memberikan efek transisi pergantian fragment yang telah kita siapkan sebelumnya.

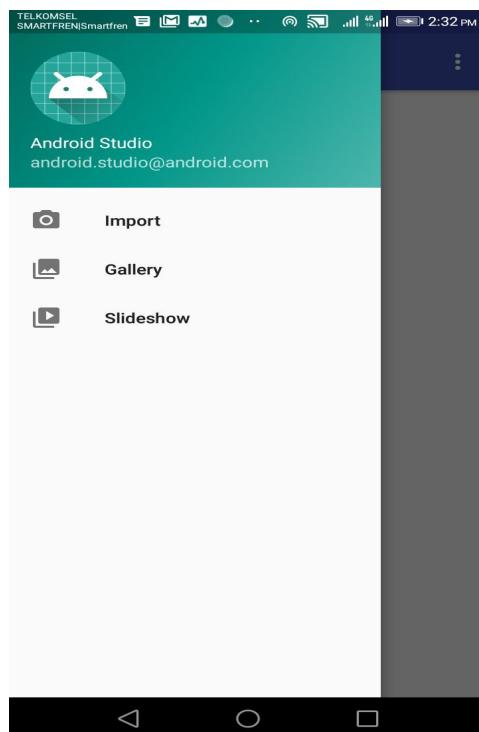
Untuk menguji jalanya aplikasi, hubungkan device android dengan laptop / pc yang menjalankan android studio via kabel data, lalu klik tombol run berwarna hijau di android studio :



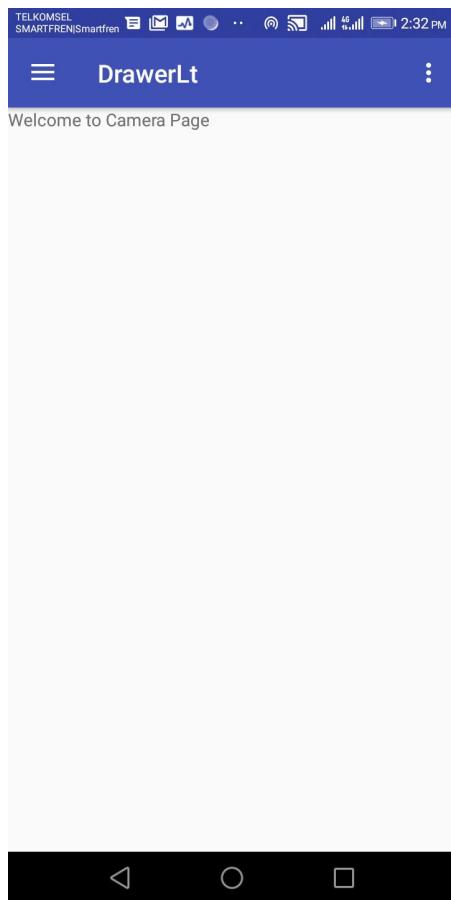
Berikut ini tampilan aplikasi dengan drawer :



Klik bar di kiri maka akan muncul menu drawer :



Saat klik menu import, maka akan terjadi transaksi pergantian framelayout dengan id “frame”



Frame terlihat tergantikan dengan frame camera yang telah kita siapkan sebelumnya.

D.4. Orientasi Tetap

Pada saat pembuatan aplikasi, kita bisa menonaktifkan fitur pergantian orientasi pada saat posisi layar android berubah ubah. Misal kita ingin membuat layout agar tetap menjadi portrait.

Untuk membuat suatu activity tidak terpengaruh orientasi layar, tambakan 2 atribut ini pada tag xml untuk activity di AndroidManifest.xml :

```
android:screenOrientation="portrait"
android:windowSoftInputMode="stateUnchanged"
```

Contoh jika ingin membuat activity MainActivity tidak terpengaruh orientasi layar :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity"
            android:screenOrientation="portrait"
            android:windowSoftInputMode="stateUnchanged">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

D.5. Aneka Layout

Pada bagian ini, kita akan mencoba beberapa design layout yang sering digunakan.

Buat proyek baru dengan nama “AnekaLayout”

Bagian XML

Pada aplikasi kali ini, kita membuat 8 activity, buat 8 activity dengan nama berikut ini :

1. MainActivity

Activity utama yang merupakan activity launcher (akan dijalankan pada saat pertama kali aplikasi berjalan). Activity ini memiliki layout dengan nama : activity_main.xml

2. ConstraintActivity

Contoh activity dengan layout constraint di mana activity ini memiliki layout dengan nama : activity_constraint.xml

3. ConstraintComplexActivity

Contoh activity dengan layout constraint kedua, di mana activity ini memiliki layout dengan nama : activity_constraint_complex.xml

4. Linear1Activity

Contoh activity dengan layout linear, di mana activity ini memiliki layout dengan nama : activity_linear1.xml

5. Linear2Activity

Contoh activity dengan layout linear kedua, di mana activity ini memiliki layout dengan nama :

activity_linear2.xml

6. Relative1Activity

Contoh activity dengan layout relative, di mana activity ini memiliki layout dengan nama activity_relative1.xml

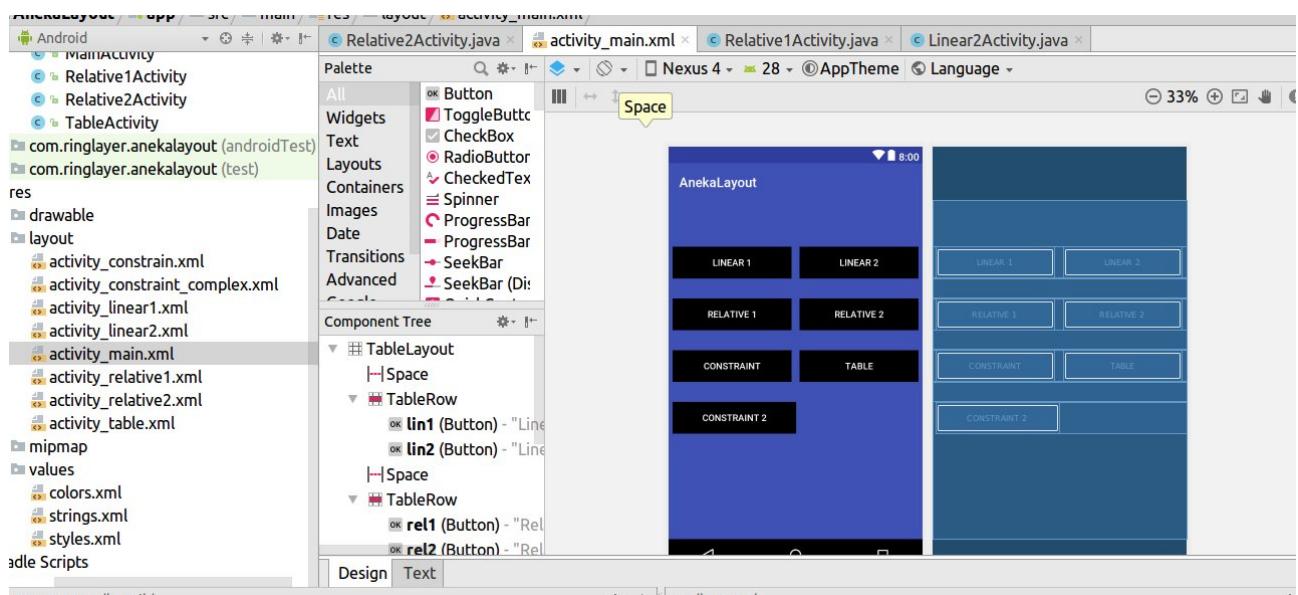
7. Relative2Activity

Contoh activity dengan layout relative kedua, di mana activity ini memiliki layout dengan nama activity_relative2.xml

8. TableActivity

Contoh activity dengan layout table, di mana activity ini memiliki layout dengan nama activity_table.xml

1. Merancang activity_main.xml



Pada dasarnya activity_main.xml merupakan table layout di mana masing masing baris berisi 2 tombol untuk menuju contoh contoh tampilan layout yang lainnya.

Berikut ini ini activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
```

```
tools:context="com.ringlayer.anekalayout.MainActivity">
<Space
    android:layout_width="1dp"
    android:layout_height="70dp"/>
<TableRow android:weightSum="2"
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <Button
        android:id="@+id/lin1"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:background="@color/black"
        android:text="Linear 1"
        android:layout_weight="1"
        android:layout_column="1"
    />
    <Button
        android:id="@+id/lin2"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:background="@color/black"
        android:text="Linear 2"
        android:layout_weight="1"
        android:layout_column="2"
    />
</TableRow>
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
<TableRow android:weightSum="2"
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <Button
        android:id="@+id/rel1"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:background="@color/black"
        android:text="Relative 1"
        android:layout_weight="1"
        android:layout_column="1"
    />
    <Button
        android:id="@+id/rel2"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:background="@color/black"
        android:text="Relative 2"
        android:layout_weight="1"
        android:layout_column="2"
    />
</TableRow>
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
<TableRow android:weightSum="2"
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <Button
        android:id="@+id/tb_constraint"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
```

```

        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:background="@color/black"
        android:text="Constraint"
        android:layout_weight="1"
        android:layout_column="1"
    />
<Button
    android:id="@+id/tb_table"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:textColor="@color/colorAccent"
    android:background="@color/black"
    android:text="Table"
    android:layout_weight="1"
    android:layout_column="2"
/>
</TableRow>
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
<TableRow android:weightSum="2"
    android:marginLeft="5dp"
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <Button
        android:id="@+id/tb_constraint2"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:background="@color/black"
        android:text="Constraint 2"
        android:layout_weight="1"
        android:layout_column="1"
    />
    <Space
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_column="2" android:layout_weight="1" />
</TableRow>
</TableLayout>

```

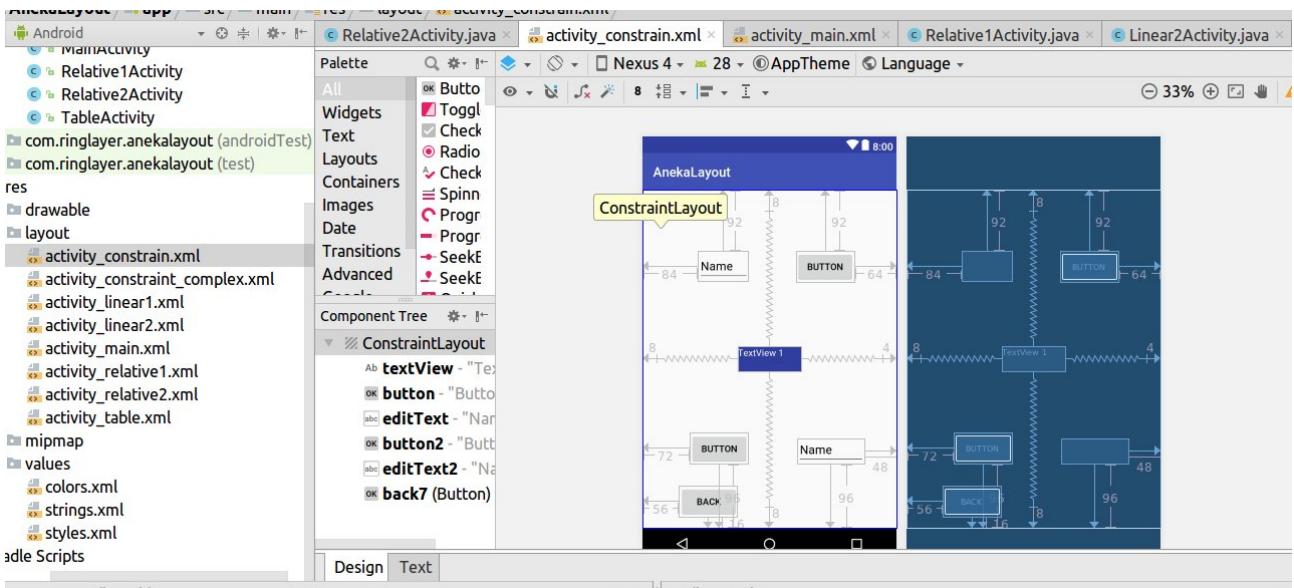
Keterangan :

Untuk membuat baris baru pada suatu table layout, kita menggunakan tag TableRow.

Pada contoh di atas masing masing baris memiliki layout weight 2, di mana tiap kolom memiliki layout weight 1 sehingga otomatis masing kolom akan dibagi setengah layar. Untuk pengaturan nomor kolom apakah kesatu atau kedua menggunakan atribut

android:layout_column=

2. Merancang activity_constraint.xml



Pada masing masing komponen, kita buat 2 tali constraint dengan patokan pojokan layar. Khusus pada Textbox berwarna biru di tengah layar, komponen itu menggunakan 4 tali constraint.

Untuk menggeser geser posisi komponen pada layout constraint sangat mudah, tinggal klik pada tengah komponen yang akan digeser, lalu seret ke posisi yang kita inginkan. Perhatian ! **Minimal masing masing komponen harus diberikan 2 tali constraint agar dapat tampil pada layout constraint.**

Berikut ini isi activity_constraint.xml :

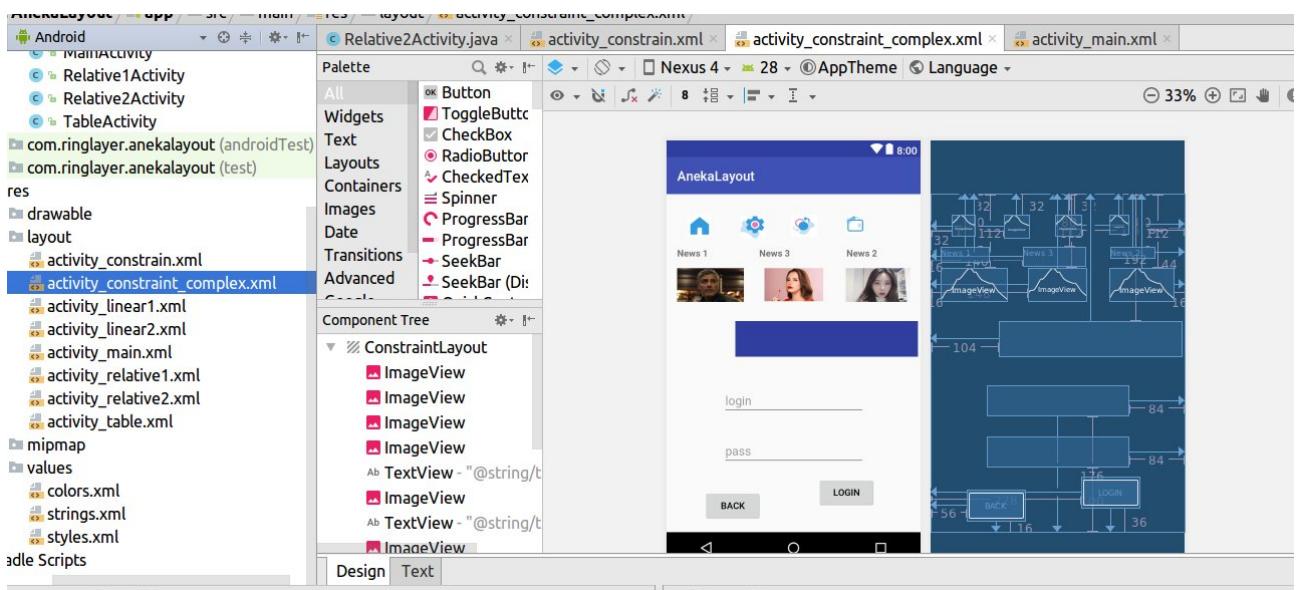
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.anekalayout.ConstrainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="96dp"
        android:layout_height="38dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="4dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@color/colorPrimaryDark"
        android:text="TextView 1"
        android:textColor="@color/colorAccent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.492"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="64dp"
        android:layout_marginTop="92dp"
        android:text="Button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```

<EditText
    android:id="@+id/editText"
    android:layout_width="77dp"
    android:layout_height="46dp"
    android:layout_marginStart="84dp"
    android:layout_marginTop="92dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="96dp"
    android:layout_marginStart="72dp"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
<EditText
    android:id="@+id/editText2"
    android:layout_width="102dp"
    android:layout_height="39dp"
    android:layout_marginBottom="96dp"
    android:layout_marginEnd="48dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
<Button
    android:id="@+id/back7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:layout_marginStart="56dp"
    android:text="Back"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>

```

3. Merancang activity_constraint_complex.xml



Pada activity ini, masing masing komponen kita berikan 2 tali constraint yang terhubung dengan pojok layar.

Berikut ini isi activity_constraint_complex.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
    tools:context="com.ringlayer.anekalayout.ConstraintComplexActivity">
    <ImageView
        android:layout_width="35dp"
        android:layout_height="33dp"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/icon1"
        android:padding="20dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <ImageView
        android:layout_width="38dp"
        android:layout_height="35dp"
        android:layout_marginStart="112dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/icon2"
        android:padding="20dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <ImageView
        android:layout_width="40dp"
        android:layout_height="33dp"
        android:layout_marginEnd="156dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/icon3"
        android:padding="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <ImageView
        android:layout_width="30dp"
        android:layout_height="29dp"
        android:layout_marginEnd="84dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/icon4"
        android:padding="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:layout_width="74dp"
        android:layout_height="23dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="80dp"
        android:text="@string/title1"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <ImageView
        android:layout_width="101dp"
        android:layout_height="51dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="112dp"
        android:background="@drawable/a1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
```

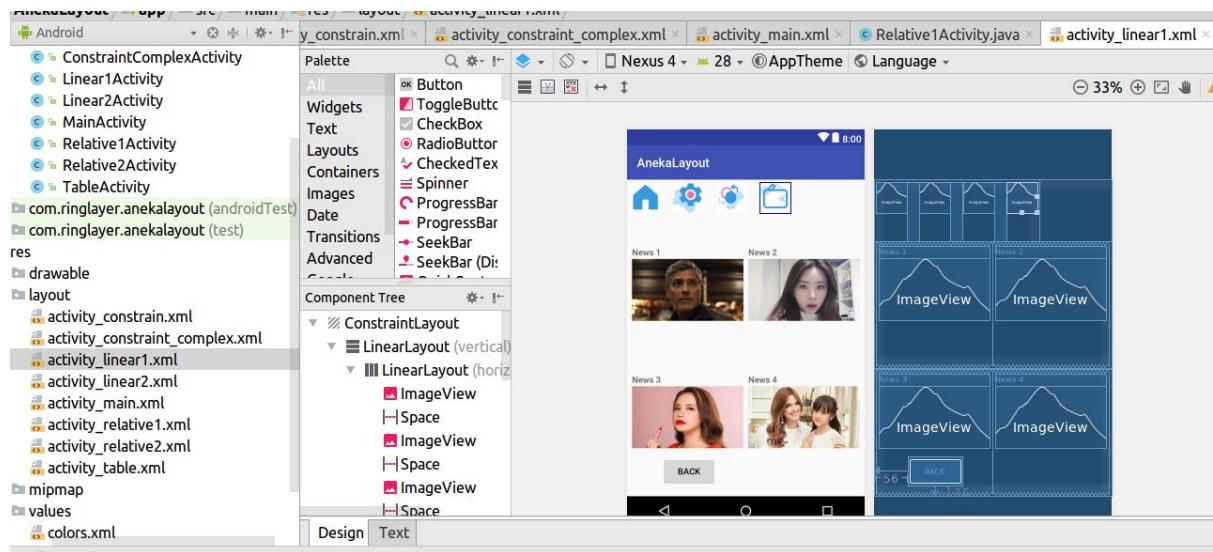
```
    android:layout_width="68dp"
    android:layout_height="17dp"
    android:layout_marginEnd="44dp"
    android:layout_marginTop="80dp"
    android:text="@string/title2"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<ImageView
    android:layout_width="97dp"
    android:layout_height="51dp"
    android:layout_marginEnd="16dp"
    android:layout_marginTop="112dp"
    android:background="@drawable/a2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<TextView
    android:layout_width="95dp"
    android:layout_height="29dp"
    android:layout_marginStart="140dp"
    android:layout_marginTop="80dp"
    android:text="@string/title3"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<ImageView
    android:layout_width="89dp"
    android:layout_height="50dp"
    android:layout_marginStart="148dp"
    android:layout_marginTop="112dp"
    android:background="@drawable/a3"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<view
    class="android.support.constraint.Group"
    id="@+id/view"
    android:layout_width="276dp"
    android:layout_height="54dp"
    android:layout_marginStart="104dp"
    android:layout_marginTop="192dp"
    android:background="@color/colorPrimaryDark"
    android:text="Your Ads Here"
    android:textColor="@color/colorAccent"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<EditText
    android:hint="login"
    android:id="@+id/editText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="176dp"
    android:layout_marginEnd="84dp"
    android:ems="10"
    android:inputType="textEmailAddress"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
<EditText
    android:hint="pass"
    android:id="@+id/editText4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="100dp"
    android:layout_marginEnd="84dp"
    android:ems="10"
    android:inputType="textPassword"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
<Button
```

```

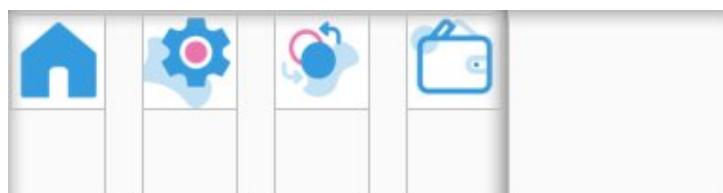
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="36dp"
        android:layout_marginStart="228dp"
        android:text="Login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
<Button
        android:id="@+id/back6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:layout_marginStart="56dp"
        android:text="Back"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>

```

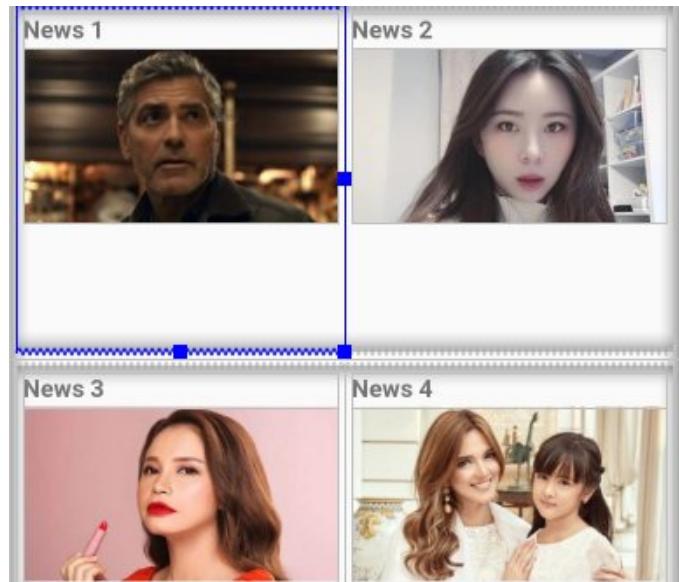
4. Merancang activity_linear1.xml



Activity ini menggunakan layout linear, Bagian atas terdiri dari linear layout dengan orientasi horizontal :



Dengan orientasi horizontal maka di dalam linear layout bisa ditempatkan komponen secara horizontal.



Bagian bawah menggunakan 2 baris , di mana kita buat dengan membuat 2 linear layout dengan orientasi horizontal sebagai parent layout , pada masing masing kolom menggunakan linear layout dengan orientasi vertical yang berisi komponen textView dan imageView.

Perhatian ! Penggunaan terlalu banyak nested layout (layout di dalam layout) sebaiknya dihindari, agar render lebih cepat dan design lebih mudah dilakukan, sebaiknya gunakan constraint layout untuk mendesign layout yang berpotensi menyebabkan adanya nested layout.

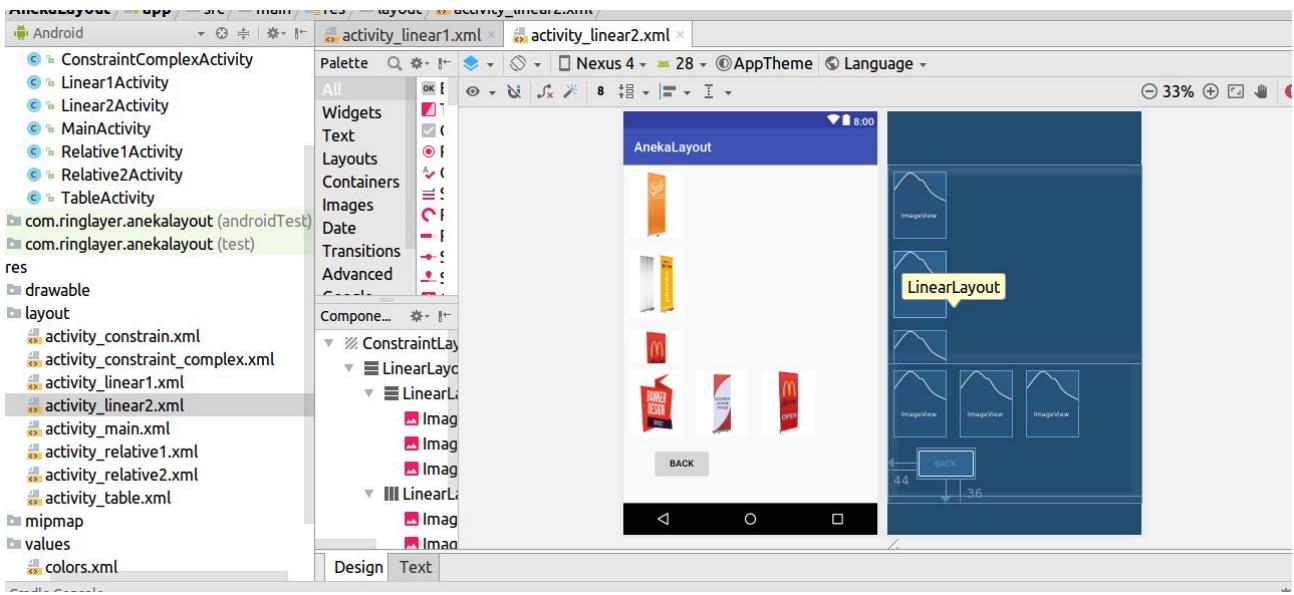
Source code activity_linear1.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
    tools:context="com.ringlayer.anekalayout.Linear1Activity">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        >
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            xmlns:tools="http://schemas.android.com/tools"
            android:orientation="horizontal"
            android:padding="4dp">
            <ImageView
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:background="@drawable/ikon1"
                android:padding="20dp" />
            <Space
                android:layout_width="20dp"
                android:layout_height="100dp"/>
            <ImageView
                android:layout_width="50dp"
```

```
        android:layout_height="50dp"
        android:background="@drawable/ikon2"
        android:padding="20dp" />
    <Space
        android:layout_width="20dp"
        android:layout_height="100dp"/>
    <ImageView
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/ikon3"
        android:padding="20dp" />
    <Space
        android:layout_width="20dp"
        android:layout_height="100dp"/>
    <ImageView
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/ikon4"
        android:padding="20dp" />
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="120dp"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="horizontal"
    android:padding="4dp"
    android:layout_weight="1">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="0.5"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:padding="4dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="20dp"
            android:text="@string/title1"
            android:textStyle="bold" />
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:background="@drawable/a1" />
    </LinearLayout>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="0.5"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:padding="4dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="20dp"
            android:text="@string/title2"
            android:textStyle="bold" />
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:background="@drawable/a2" />
    </LinearLayout>
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="120dp"
```

```
xmlns:tools="http://schemas.android.com/tools"
    android:orientation="horizontal"
    android:padding="4dp"
    android:layout_weight="1">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="0.5"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:padding="4dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:text="@string/title3"
        android:textStyle="bold" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/a3" />
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="0.5"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:padding="4dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:text="@string/title4"
        android:textStyle="bold" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/a4" />
</LinearLayout>
</LinearLayout>
</LinearLayout>
<Button
    android:id="@+id/back5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:layout_marginStart="56dp"
    android:text="Back"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

5. Merancang activity_linear2.xml



Activity ini menggunakan 2 layout linear, di mana yang pertama memiliki orientasi vertikal, dan yang kedua memiliki orientasi horizontal. Penggunaan constraint layout di sini tidak difokuskan hanya digunakan untuk memudahkan penempatan tombol di pojok kiri bawah.

Source code activity_linear2.xml :

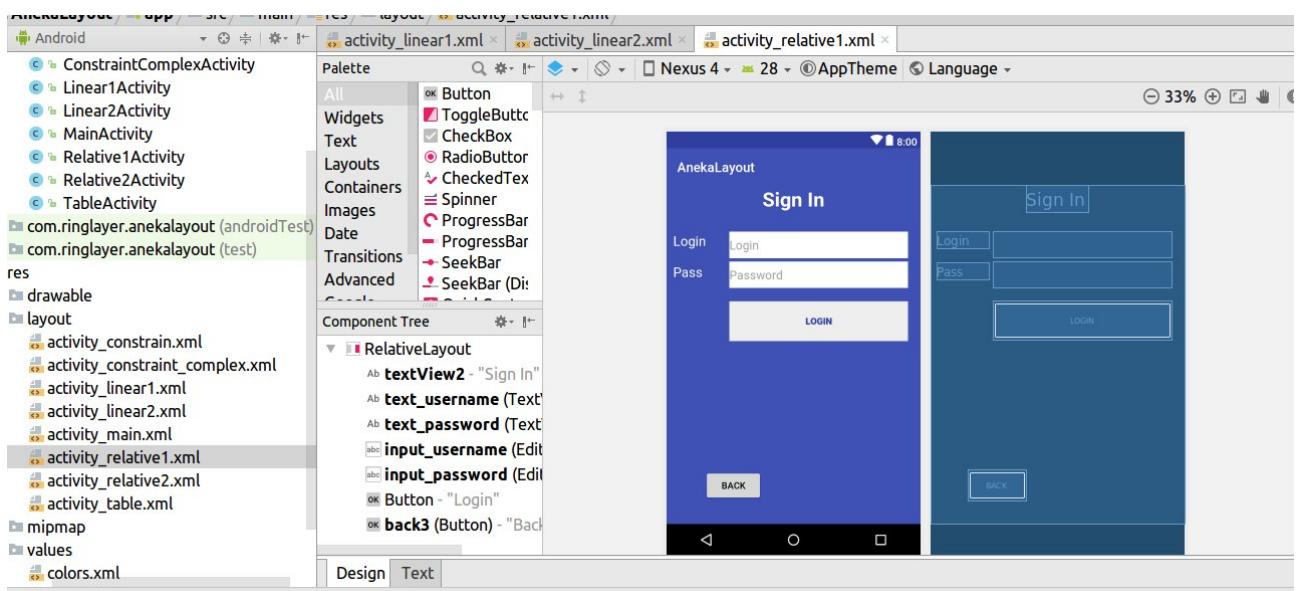
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.anekalayout.Linear2Activity">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context="com.ringlayer.anekalayout.Linear2Activity"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp">
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="300dp"
            android:orientation="vertical">
            <ImageView
                android:layout_width="80dp"
                android:layout_height="100dp"
                android:layout_margin="10dp"
                android:background="@drawable/b1" />
            <ImageView
                android:layout_width="80dp"
                android:layout_height="100dp"
                android:layout_margin="10dp"
                android:background="@drawable/b2" />
            <ImageView
                android:layout_width="80dp"
                android:layout_height="100dp"
                android:layout_margin="10dp"
                android:background="@drawable/b3" />
        </LinearLayout>
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```


        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="80dp"
            android:layout_height="100dp"
            android:layout_margin="10dp"
            android:background="@drawable/b4" />
        <ImageView
            android:layout_width="80dp"
            android:layout_height="100dp"
            android:layout_margin="10dp"
            android:background="@drawable/b5" />
        <ImageView
            android:layout_width="80dp"
            android:layout_height="100dp"
            android:layout_margin="10dp"
            android:background="@drawable/b3" />
    </LinearLayout>
</LinearLayout>
<Button
    android:id="@+id/back4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="36dp"
    android:layout_marginStart="44dp"
    android:text="Back"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>

```

6. Merancang activity_relative1.xml



Activity ini menggunakan 1 layout relative. Pada layout relative penempatan posisi tergantung pada komponen lain, oleh karena itu komponen yang akan dijadikan patokan penempatan komponen lain akan memerlukan id . Pada layout ini kita menggunakan beberapa atribut :

android:layout_below

Posisi komponen ditempatkan di bawah komponen dengan id yang dispesifikasi di tag

android:layout_toLeftOf

Posisi komponen ditempatkan di kiri komponen dengan id yang dispesifikasi di tag

android:layout_toRightOf

Posisi komponen ditempatkan di kanan komponen dengan id yang dispesifikasi di tag

android:layout_toStartOf

Penempatan pinggir dari suatu komponen ditempatkan pada awal / pinggir dari komponen dengan id yang dispesifikasi di tag, pada dasarnya penempatan ini serupa dengan **android:layout_toLeftOf**

Source code activity_relative1.xml :

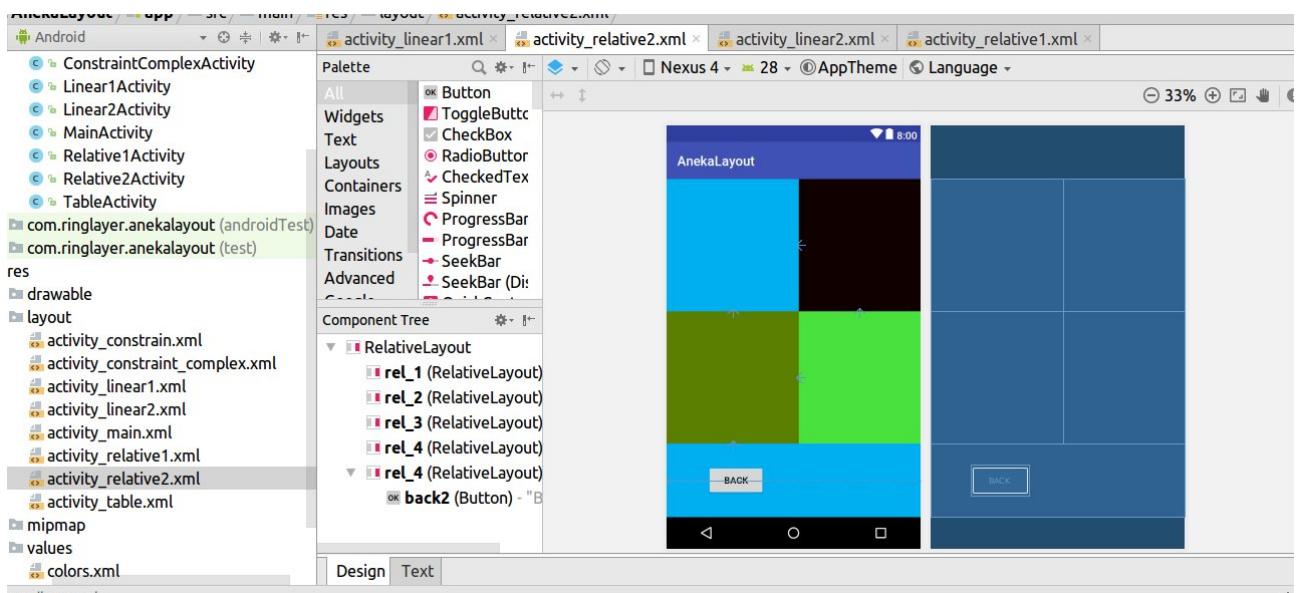
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative1Activity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign In"
        android:textStyle="bold"
        android:textSize="30sp"
        android:textColor="@color/colorAccent"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/textView2" />
    <TextView
        android:id="@+id/text_username"
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="70dp"
        android:text="Login"
        android:textColor="@color/colorAccent"
        android:textSize="20sp" />
    <TextView
        android:layout_marginTop="20dp"
        android:id="@+id/text_password"
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_username"
        android:layout_marginLeft="10dp"
        android:text="Pass"
        android:textColor="@color/colorAccent"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/input_username"
        android:layout_width="270dp"
        android:layout_height="40dp"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="70dp"
        android:layout_toRightOf="@+id/text_username"
        android:textColor="@color/colorPrimaryDark"
        android:background="@color/colorAccent"
        android:hint="Login" />
    <EditText
        android:id="@+id/input_password"
        android:layout_marginTop="5dp"
        android:background="@color/colorAccent"
        android:layout_width="270dp"
        android:layout_height="40dp"
```

```

        android:layout_below="@+id/input_username"
        android:layout_marginLeft="5dp"
        android:textColor="@color/colorPrimaryDark"
        android:layout_toRightOf="@+id/text_password"
        android:hint="Password" />
<Button
        android:layout_marginLeft="5dp"
        android:layout_width="270dp"
        android:layout_height="60dp"
        android:layout_below="@+id/input_password"
        android:layout_toRightOf="@+id/text_password"
        android:layout_marginTop="20dp"
        android:background="#eeeeee"
        android:text="Login"
        android:textColor="@color/colorPrimaryDark"
        android:textStyle="bold" />
<Button
        android:id="@+id/back3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="34dp"
        android:layout_toLeftOf="@+id/textView2"
        android:layout_toStartOf="@+id/textView2"
        android:text="Back" />
</RelativeLayout>

```

7. Merancang activity_relative2.xml



Activity ini menggunakan layout relative

Source code activity_relative2.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="@color/colorAccent"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative2Activity">

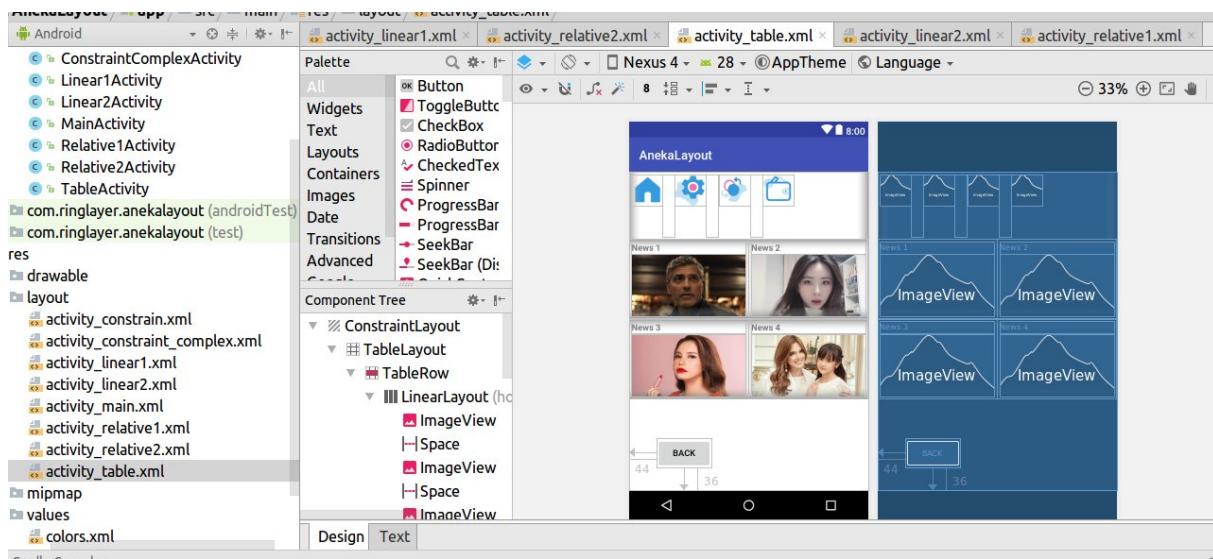
```

```

<!-- START FIRST ROW -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rel_1"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:background="@color/biru_skype"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative2Activity">
</RelativeLayout>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_toRightOf="@+id/rel_1"
    android:id="@+id/rel_2"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:background="@color/hitam1"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative2Activity">
</RelativeLayout>
<!-- EOF FIRST ROW -->
<!-- START SECOND ROW -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_below="@+id/rel_1"
    android:id="@+id/rel_3"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:background="@color/ijo_gelap"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative2Activity">
</RelativeLayout>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rel_4"
    android:layout_below="@+id/rel_2"
    android:layout_toRightOf="@+id/rel_3"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:background="@color/ijo1"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative2Activity">
</RelativeLayout>
<!-- EOF SECOND ROW -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rel_4"
    android:layout_below="@+id/rel_3"
    android:layout_width="400dp"
    android:layout_height="200dp"
    android:background="@color/biru_skype"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.ringlayer.anekalayout.Relative2Activity">
<Button
    android:id="@+id/back2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:layout_marginLeft="61dp"
    android:layout_marginStart="61dp"
    android:text="Back" />
</RelativeLayout>
</RelativeLayout>

```

8. Merancang activity_table.xml



Activity ini menggunakan 1 layout table, penggunaan pembungkus berupa constraint layout hanya digunakan untuk memudahkan penempatan tombol back di pojok kiri.

Pada bagian ini :



kita menggunakan satu tablerow dengan weight 1 , di mana di dalamnya kita tempatkan 1 linear layout dengan width 1 (lebar 100%), linear layout ini memiliki orientasi horizontal, di mana kita tempatkan 4 buah imageview di dalamnya.



Pada contoh ini kita menggunakan baris dengan weight 1, di mana untuk membagi masing masing kolom menjadi setengah lebar dari parent berupa tablerow dengan weight 1, kita gunakan atribut weight 0.5 (50%) pada masing masing komponen di dalamnya, di mana masing masing kolom merupakan 1 linear layout dengan orientasi vertical sehingga penempatan textView dan imageView di dalamnya menjadi vertikal.

Source code activity_table.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
        tools:context="com.ringlayer.anekalayout.TableActivity">
    <TableLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/colorAccent">
        <TableRow android:weightSum="1"
            android:layout_width="match_parent"
            android:layout_height="100dp">
            <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
                xmlns:app="http://schemas.android.com/apk/res-auto"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:layout_weight="1"
                xmlns:tools="http://schemas.android.com/tools"
                android:orientation="horizontal"
                android:layout_column="1"
                android:padding="4dp">
                <ImageView
                    android:layout_width="50dp"
                    android:layout_height="50dp"
                    android:background="@drawable/icon1"
                    android:padding="20dp" />
                <Space
                    android:layout_width="20dp"
                    android:layout_height="100dp"/>
                <ImageView
                    android:layout_width="50dp"
                    android:layout_height="50dp"
                    android:background="@drawable/icon2"
                    android:padding="20dp" />
                <Space
                    android:layout_width="20dp"
                    android:layout_height="100dp"/>
                <ImageView
                    android:layout_width="50dp"
                    android:layout_height="50dp"
                    android:background="@drawable/icon3"
                    android:padding="20dp" />
                <Space
                    android:layout_width="20dp"
                    android:layout_height="100dp"/>
                <ImageView
                    android:layout_width="50dp"
                    android:layout_height="50dp"
                    android:background="@drawable/icon4"
                    android:padding="20dp" />
            </LinearLayout>
        </TableRow>
        <!-- content white -->
        <TableRow android:weightSum="2"
            android:layout_width="match_parent"
            android:layout_height="50dp">
            <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
                xmlns:app="http://schemas.android.com/apk/res-auto"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
```

```
    android:layout_weight="1"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_column="1"
    android:padding="4dp"
    android:background="@drawable/line">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:text="@string/title1"
        android:textStyle="bold" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/a1" />
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:background="@drawable/line"
    android:layout_column="2"
    android:padding="4dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:text="@string/title2"
        android:textStyle="bold" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/a2" />
</LinearLayout>
</TableRow>
<TableRow android:weightSum="2"
    android:layout_width="match_parent"
    android:layout_height="50dp">
</TableRow>
<TableRow android:weightSum="2"
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:layout_column="1"
        android:padding="4dp"
        android:background="@drawable/line">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="20dp"
            android:text="@string/title3"
            android:textStyle="bold" />
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:background="@drawable/a3" />
    </LinearLayout>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
```

```

    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:background="@drawable/line"
    android:layout_column="2"
    android:padding="4dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:text="@string/title4"
        android:textStyle="bold" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/a4" />
    </LinearLayout>
</TableRow>
<!-- eof content white -->
</TableLayout>
<Button
    android:id="@+id/back1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="36dp"
    android:layout_marginStart="44dp"
    android:text="Back"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>

```

Bagian Source Code

1. Source Code MainActivity.java :

```

package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    Button lin1;
    Button lin2;
    Button rel1;
    Button rel2;
    Button constrain;
    Button table;
    Button constrain2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            lin1 = (Button) findViewById(R.id.lin1);
            lin2 = (Button) findViewById(R.id.lin2);
            rel1 = (Button) findViewById(R.id.rel1);
            rel2 = (Button) findViewById(R.id.rel2);
            constrain = (Button) findViewById(R.id.tb_constrain);
            table = (Button) findViewById(R.id.tb_table);
            constrain2 = (Button) findViewById(R.id.tb_constrain2);
        }
    }
}

```

```
lin1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Linear1();
    }
});
lin2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Linear2();
    }
});
rel1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Relative1();
    }
});
rel2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Relative2();
    }
});
constrain.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Constrain();
    }
});
table.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Table();
    }
});
constrain2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Constrain2();
    }
});
}
catch (Exception e) {
    Log.e("error", e.toString());
}
}
void Linear1() {
try {
    Intent intent = new Intent(MainActivity.this, Linear1Activity.class);
    startActivity(intent);
    finish();
}
catch (Exception e) {
    Log.e("err", e.toString());
}
}
void Linear2() {
try {
    Intent intent = new Intent(MainActivity.this, Linear2Activity.class);
    startActivity(intent);
    finish();
}
catch (Exception e) {
    Log.e("err", e.toString());
}
}
void Relative1() {
try {
    Intent intent = new Intent(MainActivity.this, Relative1Activity.class);
    startActivity(intent);
    finish();
}
catch (Exception e) {
    Log.e("err", e.toString());
}
}
```

```

    }
    void Relative2() {
        try {
            Intent intent = new Intent(MainActivity.this, Relative2Activity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
    void Constrain() {
        try {
            Intent intent = new Intent(MainActivity.this, ConstrainActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
    void Constrain2() {
        try {
            Intent intent = new Intent(MainActivity.this,
ConstraintComplexActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
    void Table() {
        try {
            Intent intent = new Intent(MainActivity.this, TableActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}

```

PENJELASAN

```
public class MainActivity extends AppCompatActivity {
```

Baris ini digunakan untuk membuat kelas baru bernama MainActivity yang menurunkan atribut dan method dari parent class kelas AppCompatActivity

```

Button lin1;
Button lin2;
Button rel1;
Button rel2;
Button constrain;
Button table;
Button constrain2;
```

Selanjutnya kita inisialisasi setiap objek button yang akan kita gunakan.

```

@Override
protected void onStart() {
    super.onStart();
```

Di sini kita melakukan override pada metode onStart dari kelas parent, di mana dengan keyword super maka rutin di dalam blok method onStart yang asli akan dieksekusi terlebih dahulu sebelum rutin yang kita tambahkan saat override.

```
lin1 = (Button) findViewById(R.id.lin1);
lin2 = (Button) findViewById(R.id.lin2);
rel1 = (Button) findViewById(R.id.rel1);
rel2 = (Button) findViewById(R.id.rel2);
constrain = (Button) findViewById(R.id.tb_constraint);
table = (Button) findViewById(R.id.tb_table);
constrain2 = (Button) findViewById(R.id.tb_constraint2);
```

Pada rutin rutin ini kita menyimpan referensi untuk masing masing tombol dengan id yang telah kita tentukan di layout ke masing masing objek button yang telah kita siapkan sebelumnya.

```
lin1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Linear1();
    }
});
lin2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Linear2();
    }
});
<CUTTED>
```

Untuk masing masing objek tombol tadi kita pasang callback method untuk menangkap klik mouse dari user, di mana saat diklik maka akan memanggil method yang telah ditentukan.

```
void Linear1() {
    try {
        Intent intent = new Intent(MainActivity.this, Linear1Activity.class);
        startActivity(intent);
        finish();
    }
    catch (Exception e) {
        Log.e("err", e.toString());
    }
}
```

Contoh di atas adalah contoh salah satu method di mana method ini akan menjalankan activity untuk merender layout xml yang telah kita buat tadi. Kegunaan baris finish() adalah untuk mengakhiri MainActivity. Untuk method lain memiliki fungsi yang sama sehingga tidak perlu dijelaskan satu persatu.

2. Source Code ConstrainActivity.java

```
package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class ConstrainActivity extends AppCompatActivity {
    Button back7;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_constraint);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back7 = (Button) findViewById(R.id.back7);
            back7.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Back();
                }
            });
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private void Back() {
        try {
            Intent intent = new Intent(ConstrainActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}

```

PENJELASAN

Method Back merupakan method yang dipanggil saat terjadi klik tombol Back, di mana metode ini akan kembali menjalankan activity MainActivity dan mengakhiri activity yang sedang berjalan dengan finish().

3. Source Code ConstraintComplexActivity.java

```

package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class ConstraintComplexActivity extends AppCompatActivity {
    Button back6;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_constraint_complex);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back6 = (Button) findViewById(R.id.back6);
            back6.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Back();
                }
            });
        }
    }
}

```

```

        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private void Back() {
        try {
            Intent intent = new Intent(ConstraintComplexActivity.this,
MainActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}

```

PENJELASAN

Pada dasarnya cara kerja activity ini dan activity activity di bawahnya adalah sama seperti contoh activity sebelumnya sehingga tidak perlu dijelaskan ulang.

4. Source Code Linear1Activity.java

```

package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class Linear1Activity extends AppCompatActivity {
    Button back5;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_linear1);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back5 = (Button) findViewById(R.id.back5);
            back5.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Back();
                }
            });
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private void Back() {
        try {
            Intent intent = new Intent(Linear1Activity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}

```

5. Source Code Linear2Activity.java

```
package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class Linear2Activity extends AppCompatActivity {
    Button back4;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_linear2);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back4 = (Button) findViewById(R.id.back4);
            back4.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Back();
                }
            });
        } catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private void Back() {
        try {
            Intent intent = new Intent(Linear2Activity.this, MainActivity.class);
            startActivity(intent);
            finish();
        } catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}
```

6. Source Code Relative1Activity.java

```
package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class Relative1Activity extends AppCompatActivity {
    Button back3;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_relative1);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back3 = (Button) findViewById(R.id.back3);
            back3.setOnClickListener(new View.OnClickListener() {
```

```

        public void onClick(View v) {
            Back();
        }
    });
}
catch (Exception e) {
    Log.e("error", e.toString());
}
}
private void Back() {
try {
    Intent intent = new Intent(RelativeActivity.this, MainActivity.class);
    startActivity(intent);
    finish();
}
catch (Exception e) {
    Log.e("err", e.toString());
}
}
}
}

```

7. Source Code Relative2Activity.java

```

package com.ringlayer.anekalayout;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class Relative2Activity extends AppCompatActivity {
    Button back2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_relative2);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back2 = (Button) findViewById(R.id.back2);
            back2.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Back();
                }
            });
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private void Back() {
        try {
            Intent intent = new Intent(Relative2Activity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}

```

8. Source Code TableActivity.java

```
package com.ringlayer.anekalayout;
```

```

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
public class TableActivity extends AppCompatActivity {
    Button back1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_table);
    }
    @Override
    protected void onStart() {
        super.onStart();
        try {
            back1 = (Button) findViewById(R.id.back1);
            back1.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Back();
                }
            });
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private void Back() {
        try {
            Intent intent = new Intent(TableActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
        catch (Exception e) {
            Log.e("err", e.toString());
        }
    }
}

```

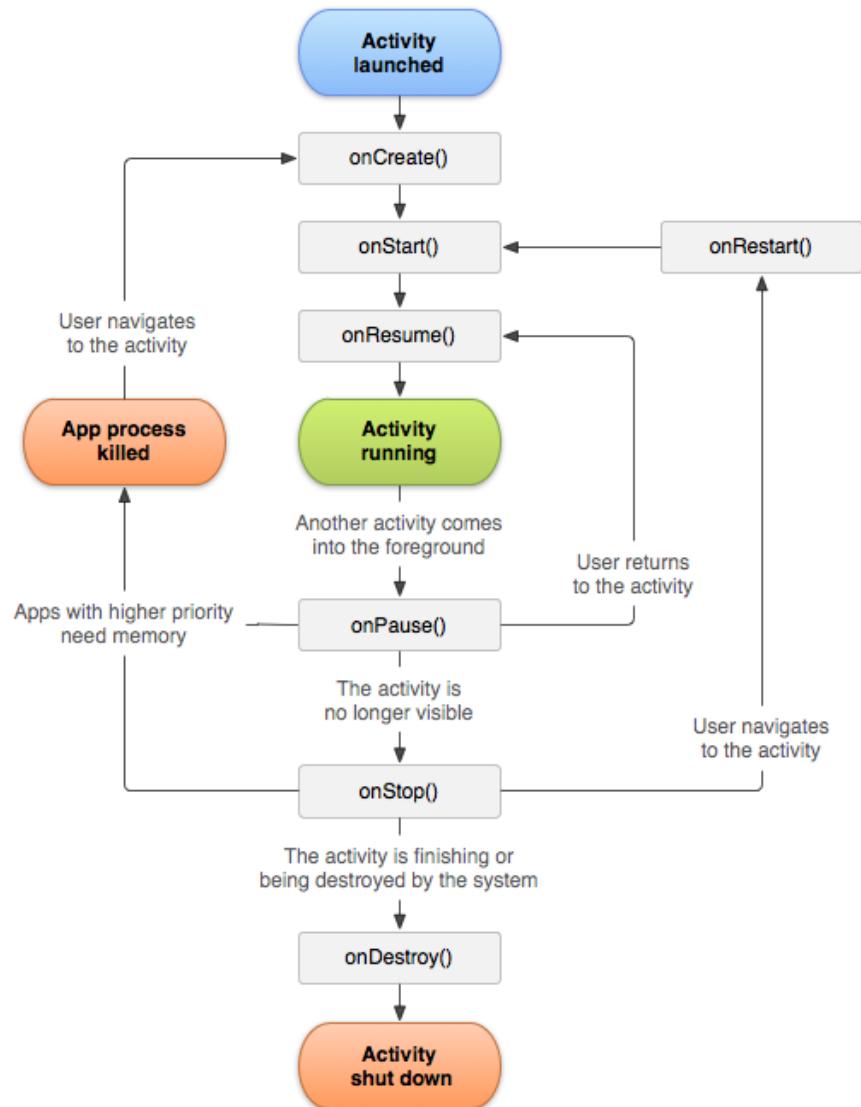
E. Activity dan Intent

Activity

Activity merupakan 1 layar dengan user interface pada aplikasi android. Activity di Android menggunakan layout sebagai container untuk view lain di dalamnya, bahkan juga bisa berisi layout lain.

Daur Hidup Activity

Berikut ini adalah diagram yang menunjukkan daur hidup dari suatu activity



Seluruh activity pada aplikasi android direpresentasikan dengan kelas activity. Activity class ini merupakan subclass dari android.app.activity. Kelas activity ini berisi metode-metode yang merepresentasikan daur hidup dari suatu activity yaitu :

`onCreate()`

Callback ini dipanggil pada saat activity dibentuk, pada onCreate() layout belum terlihat pengguna. Pada body onCreate terdapat rutin untuk melakukan render dari layout. Berikut ini contoh onCreate pada activity dengan nama MainActivity yang akan merender layout activity_main :

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

onStart()

Callback ini dipanggil saat activity sudah terlihat oleh user tapi interaksi belum dimulai. Bagian ini bisa disisipkan beberapa rutin seperti melakukan registrasi broadcast receiver.

onResume()

Callback ini dipanggil saat user sudah mulai berinteraksi dengan aplikasi.

onPause()

Callback ini dipanggil saat user akan meninggalkan activity, saat callback ini dipanggil activity ini tidak ada di foreground, tapi masih bisa terlihat jika user berada pada modus multi window.

onStop()

Callback ini dipanggil saat activity sudah tidak terlihat lagi.

onDestroy()

Callback ini dipanggil sebelum activity dihancurkan. Sistem memanggil callback ini karena :

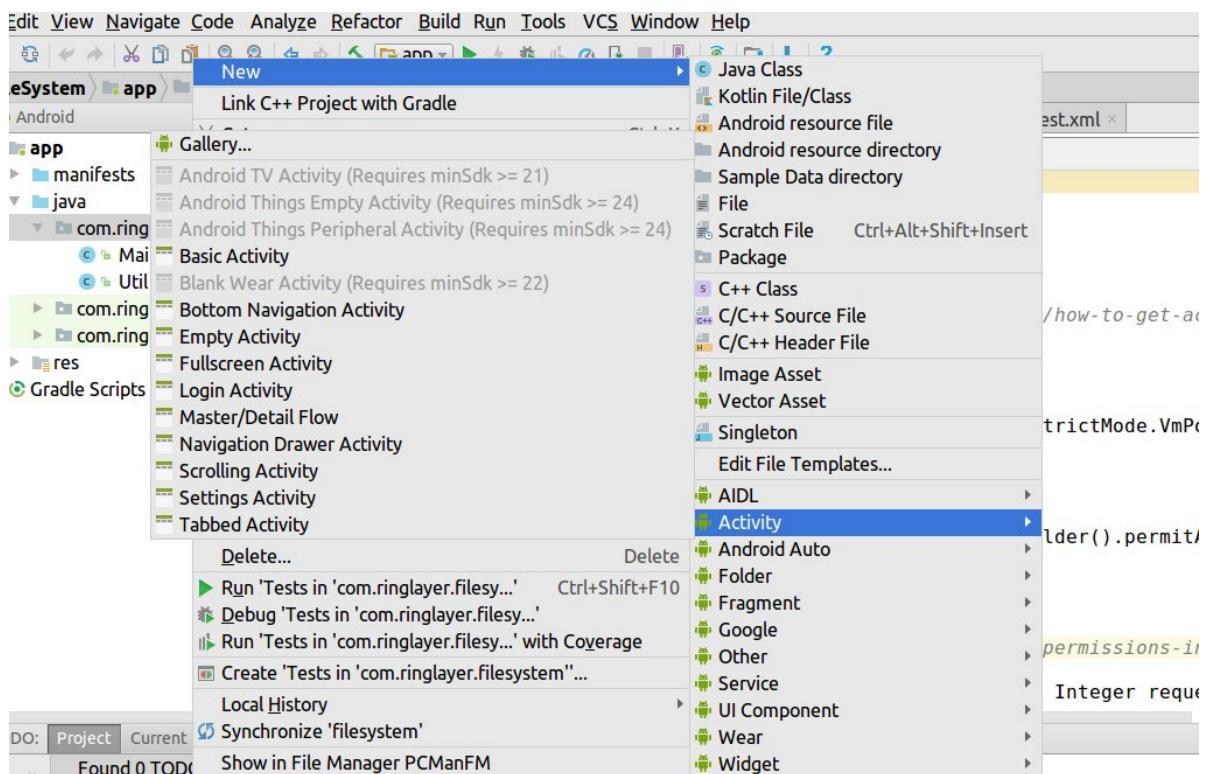
1. Activity selesai dijalankan (bisa karena user menutup activity atau bisa juga karena pemanggilan metode finish())
2. Sistem menghancurkan activity karena adanya perubahan konfigurasi, misal karena adanya perubahan rotasi layar dari landscape menjadi portrait.

onRestart()

Callback ini dipanggil setelah pemanggilan callback onStop(). Callback ini dipanggil saat activity ditampilkan ulang di layar.

Membuat Activity

Untuk membuat activity bisa dilakukan langsung dari menu android studio. Caranya:



Pada subfolder java, klik kanan pada folder source code aplikasi, lalu klik kanan, pilih New lalu pilih Activity.

F. Trik Dasar Pemrograman di Android Studio

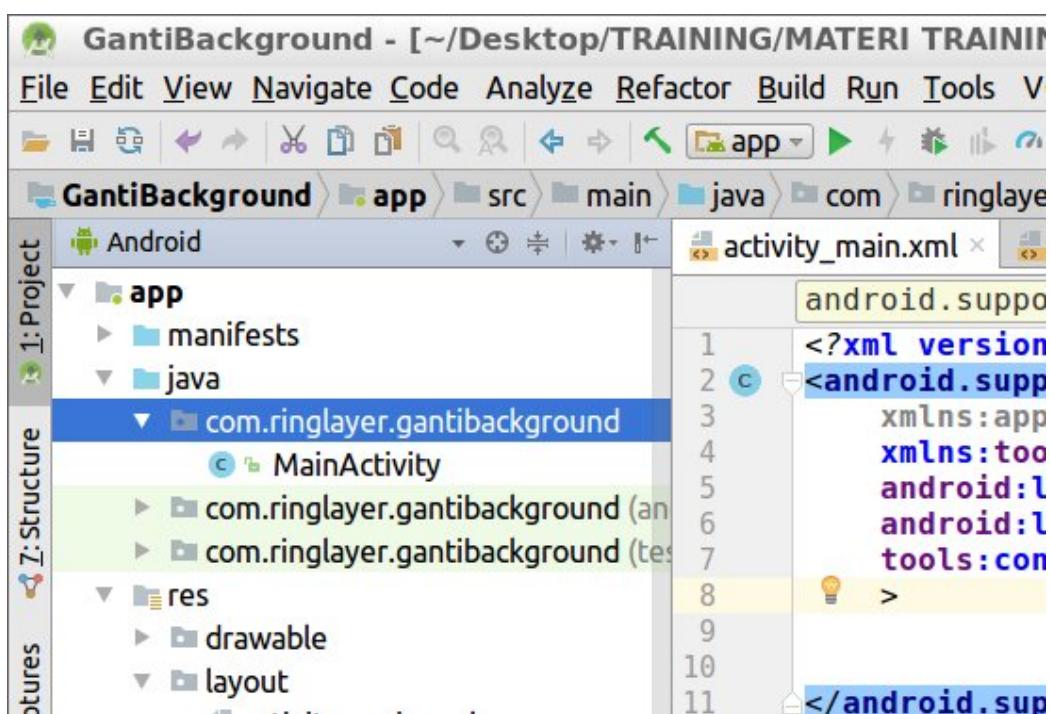
F.1. Mengganti Background pada Layout

Pada contoh ini kita akan mencoba mengganti background color dan image untuk aplikasi android yang kita buat.

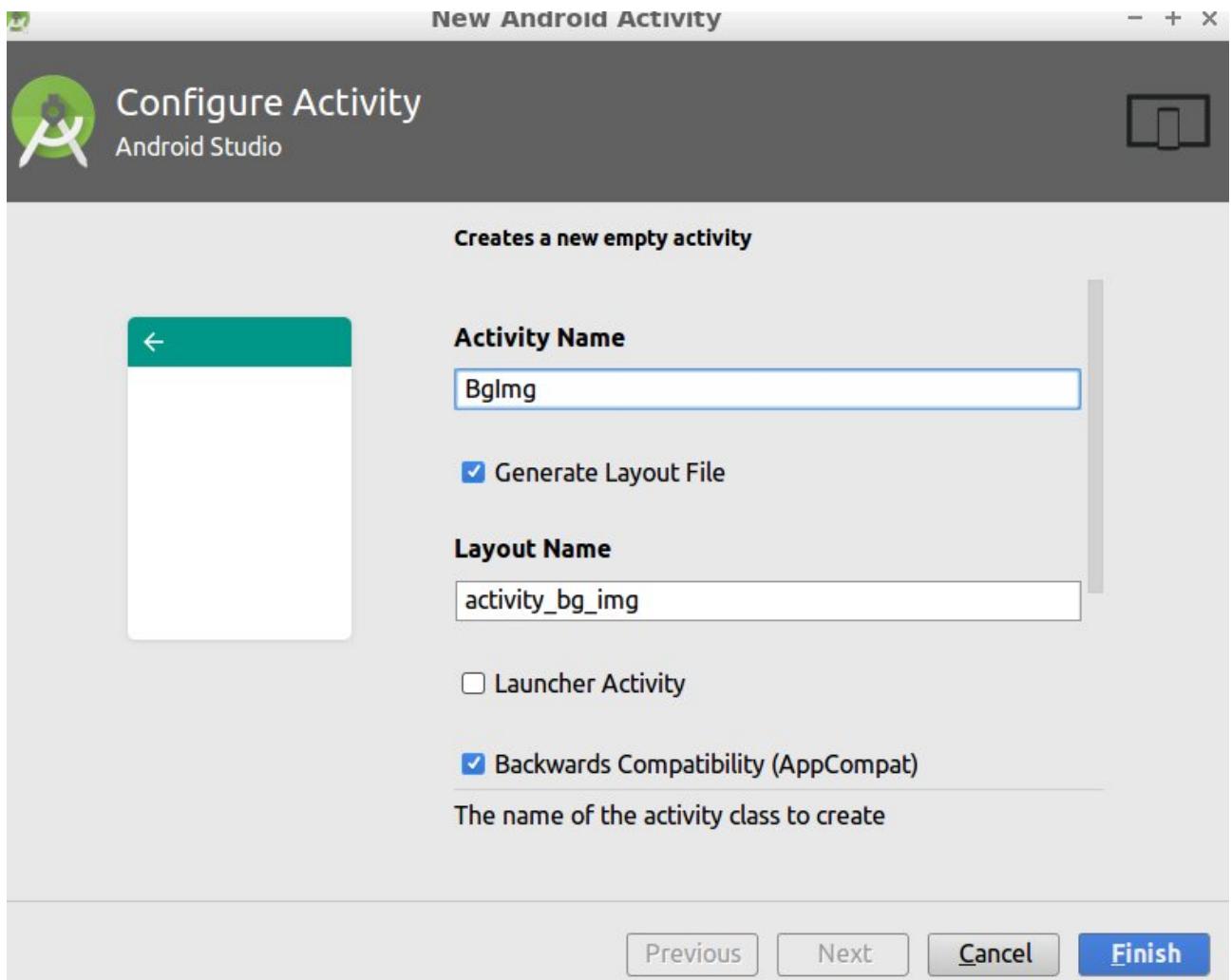
Siapkan proyek baru dengan nama GantiBackground

Secara default, kita sudah memiliki 1 layout bernama activity_main.xml dengan activity MainActivity.

Pada contoh kali ini, kita akan menambah 1 activity lagi dengan nama BgImg. Klik kanan pada folder java untuk proyek ini



Klik kanan lalu pilih new → activity, lalu pilih Empty Activity.



Beri nama activity dengan nama BgImg, layout yang akan digunakan untuk menguji background image nanti adalah adalah activity_bg_img (activity_bg_img.xml).

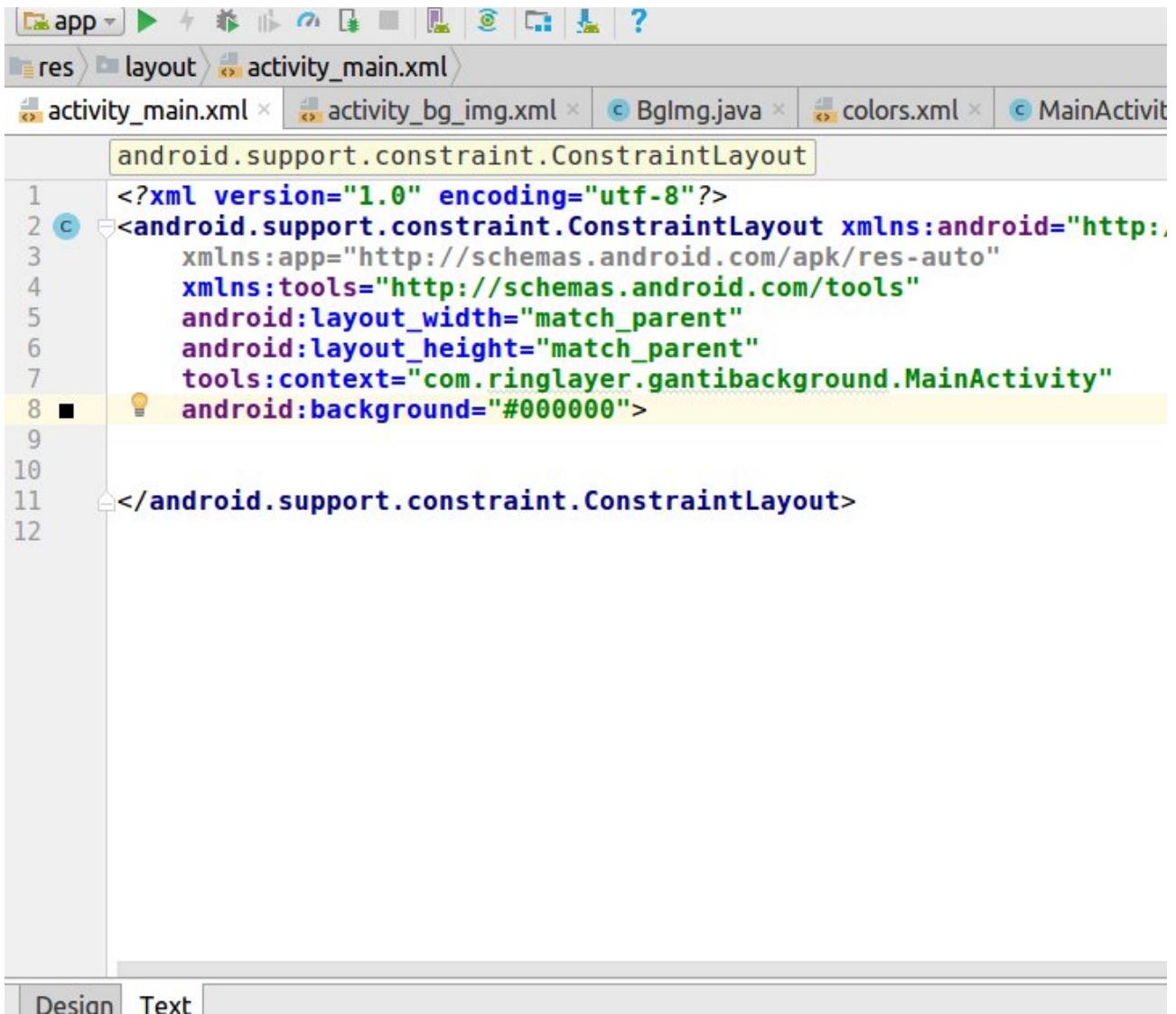
Untuk uji coba pertama kita akan mencoba mengganti warna background pada activity_main. Buka activity_main.xml

Untuk mengganti warna background layout, kita bisa mendefinisikan warna background di colors.xml atau langsung menulis kode warna di xml activity_main.xml.

Untuk mengganti warna background dengan cara langsung menulis kode warna, caranya adalah dengan menggunakan atribut :

android:background="KODE WARNA ATAU RESOURCE COLOR DARI COLORS.XML ATAU IMAGE RESOURCE"

contoh:



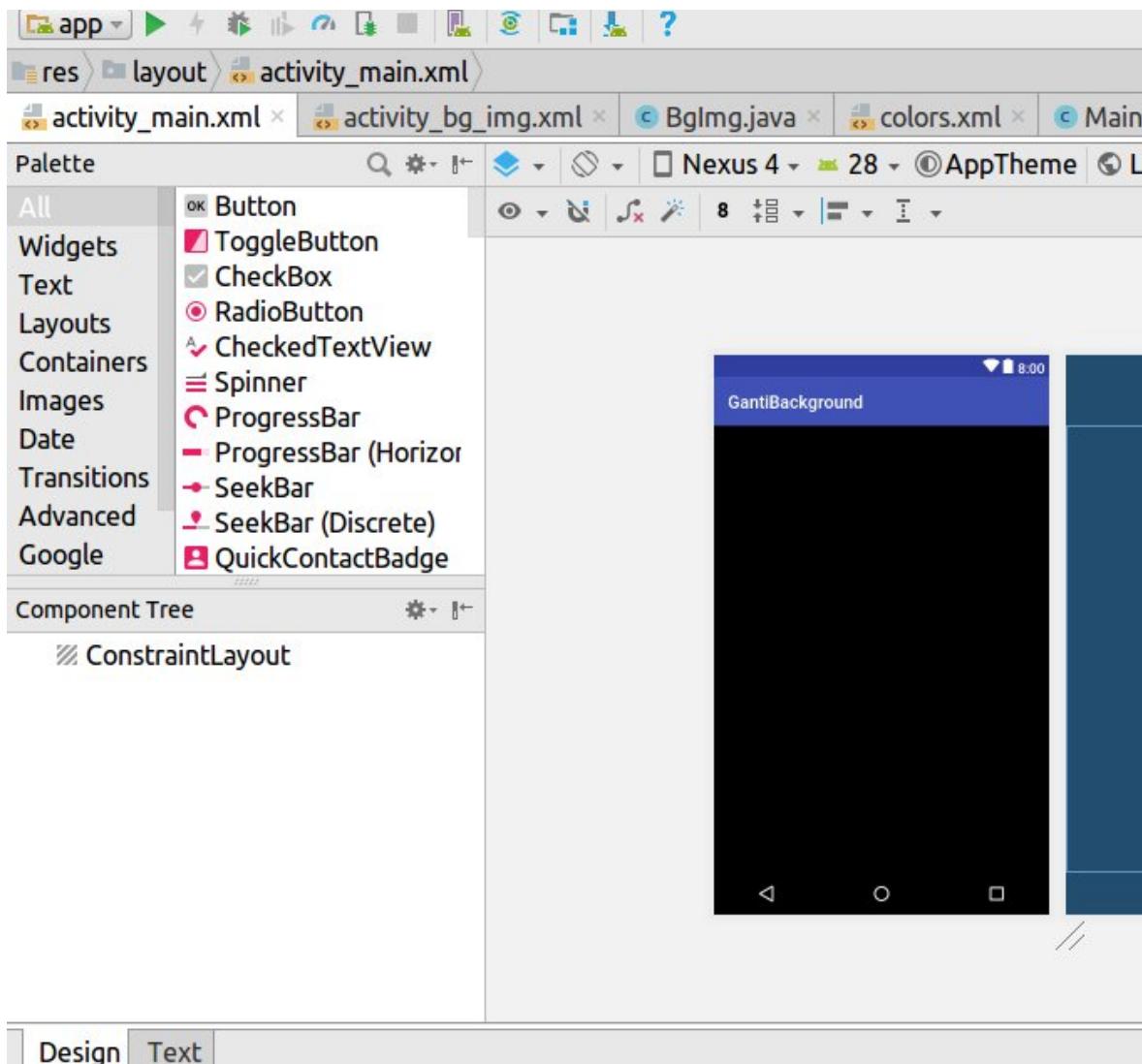
The screenshot shows the Android Studio interface with the XML editor open. The file path is `res/layout/activity_main.xml`. The code editor displays the following XML code:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.ringlayer.gantibackground.MainActivity"
8     android:background="#000000">
9
10
11 </android.support.constraint.ConstraintLayout>
12
```

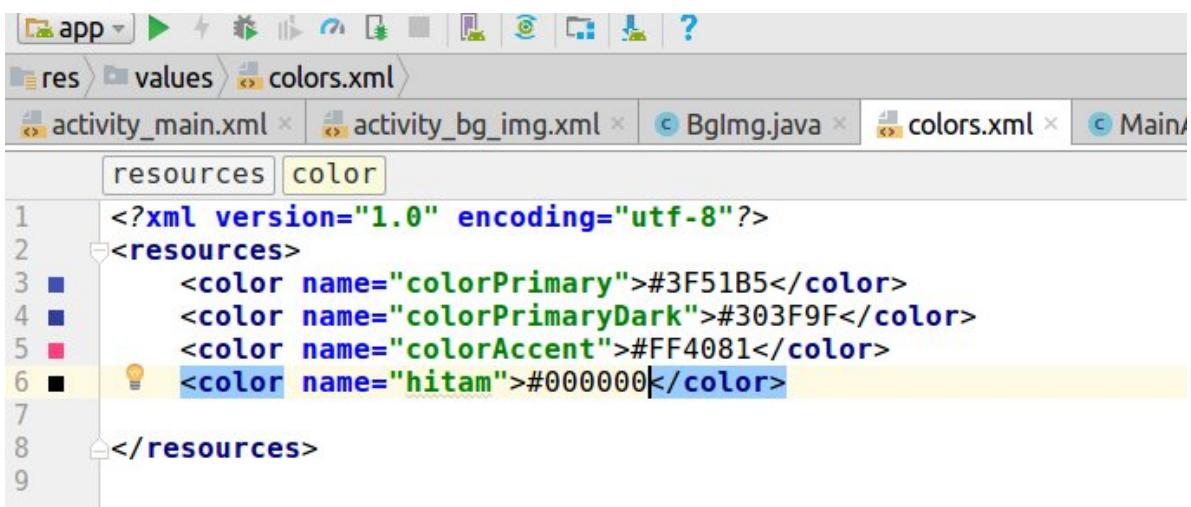
The line `android:background="#000000"` is highlighted in yellow, indicating it is selected or being edited. The bottom navigation bar shows the "Design" tab is active, while the "Text" tab is visible.

Pada contoh di atas kita akan mencoba mengganti background menjadi berwarna hitam kode warna adalah `#000000`. Untuk mendapatkan kode kode warna rgb bisa diambil dari <https://rgbcolorcode.com> atau bisa juga dengan cara diambil dari photoshop atau image editor lainnya.

Klik tab design, kita bisa melihat warna layout activity_main berubah menjadi hitam:

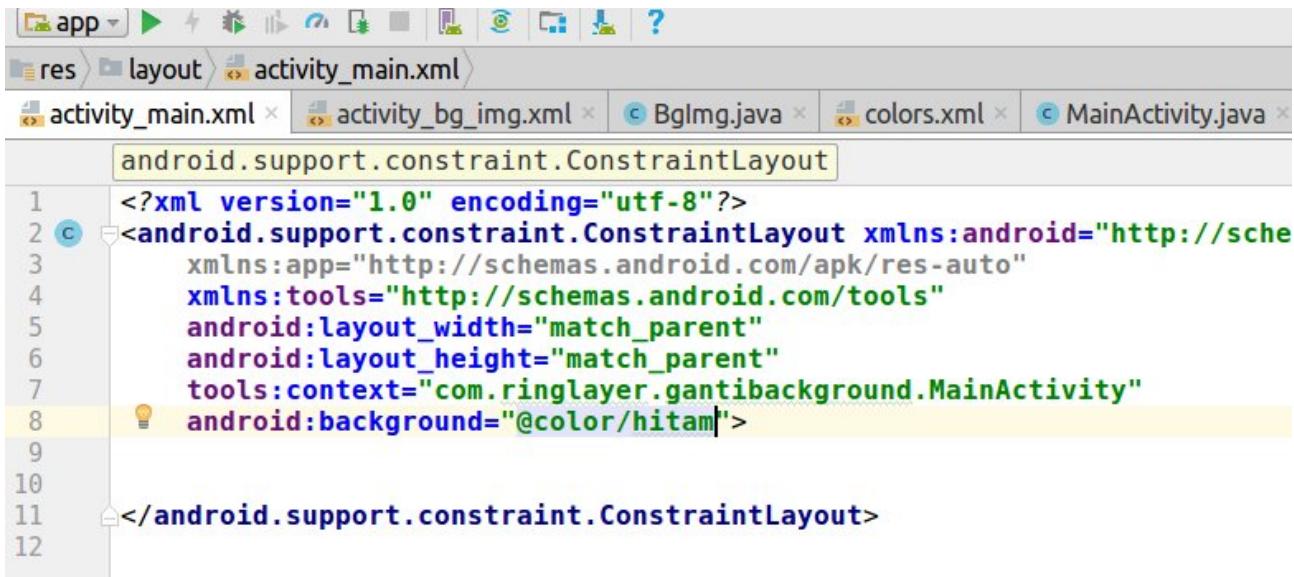


Selain mengetik langsung kode warna, bisa juga mendefinisikan terlebih dahulu melalui colors.xml, lalu selanjutnya kita pakai di activity_main.xml.
Pertama tama kita definisikan terlebih dahulu kode warna di colors.xml



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="hitam">#000000</color>
</resources>
```

Selanjutnya, gunakan kode warna yang sudah kita definisikan di colors.xml ke dalam activity_main.xml

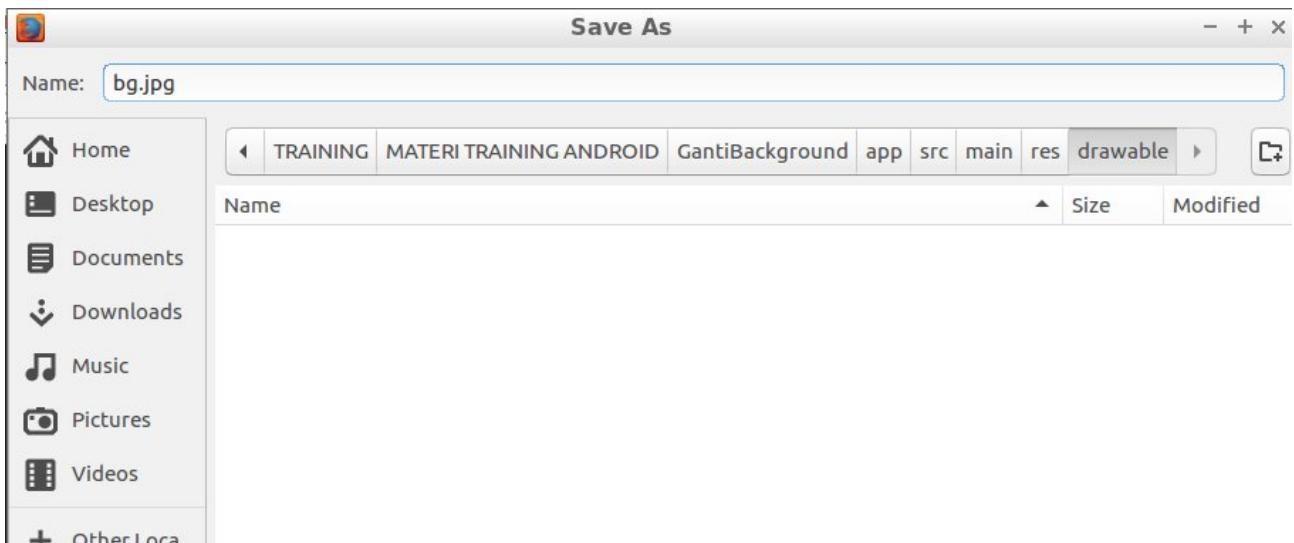


The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it, the file tree shows 'res' and 'layout'. The main editor window displays the XML code for 'activity_main.xml'. The code defines a ConstraintLayout with various attributes. Line 8 highlights the 'android:background' attribute set to '@color/hitam'. The code is as follows:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.ringlayer.gantibackground.MainActivity"
8     android:background="@color/hitam">
9
10
11 </android.support.constraint.ConstraintLayout>
12
```

Selanjutnya kita akan mencoba mengganti background image pada layout BgImg.

Pada contoh kali ini kita akan mengganti background dari layout activity_bg_img.xml dengan file jpg dengan nama bg.jpg



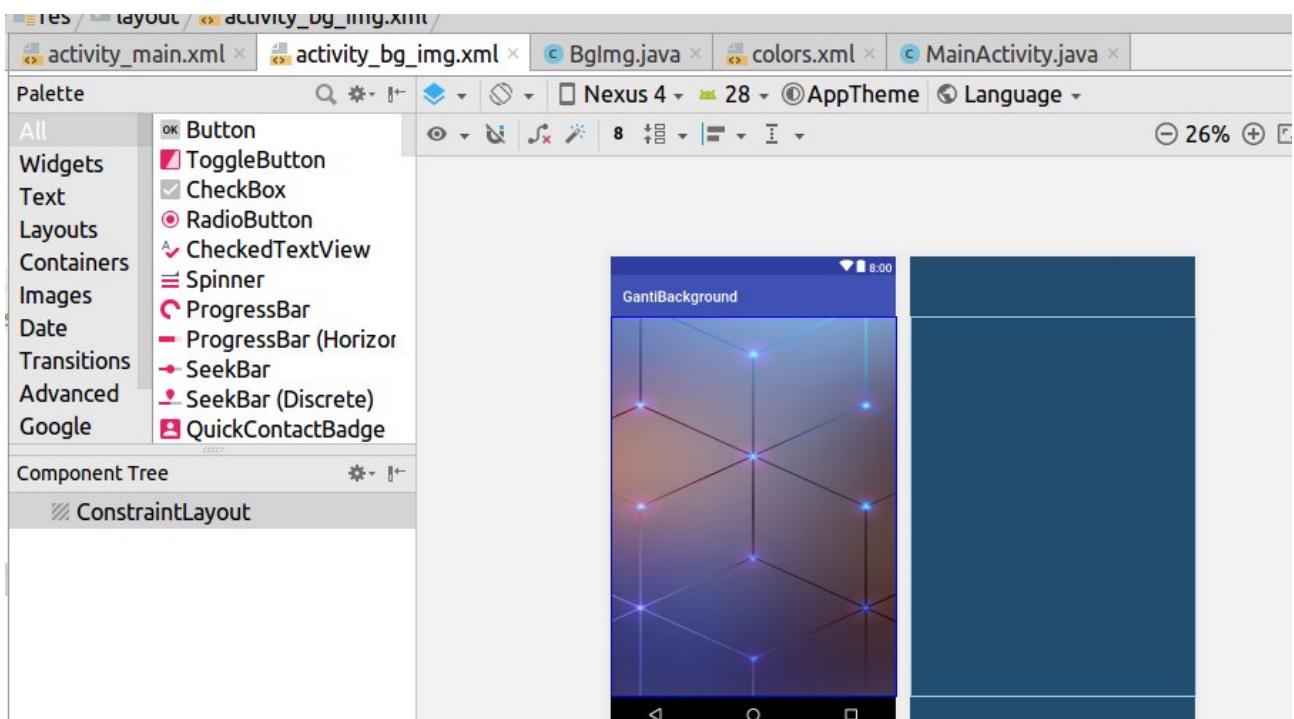
Simpan file jpg dengan nama bg.jpg di folder drawable.

Selanjutnya buka xml activity_bg_img.xml. Ganti backgound menjadi image dengan cara menambahkan atribut ini:

android:background="@drawable/bg"

```
res/layout/activity_bg_img.xml
activity_main.xml activity_bg_img.xml Bglmg.java colors.xml MainActivity.java
    android.support.constraint.ConstraintLayout
1     <?xml version="1.0" encoding="utf-8"?>
2     <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.
3         xmlns:app="http://schemas.android.com/apk/res-auto"
4         xmlns:tools="http://schemas.android.com/tools"
5         android:layout_width="match_parent"
6         android:layout_height="match_parent"
7         tools:context="com.ringlayer.gantibackground.BgImg"
8         android:background="@drawable/bg"
9     >
10
11     </android.support.constraint.ConstraintLayout>
12
```

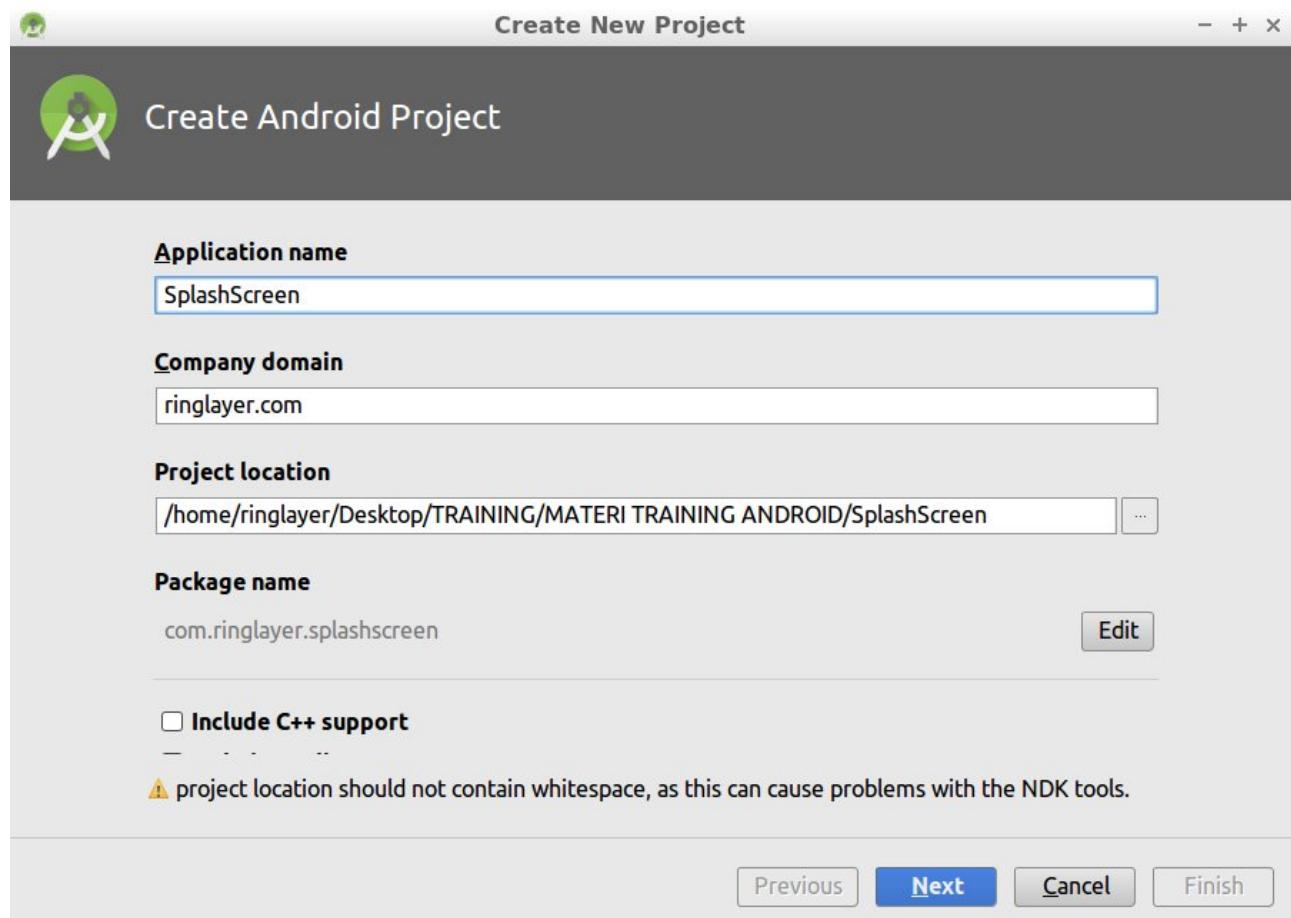
Untuk melihat hasil perubahan di tab design, pertama tama klik dahulu Build lalu pilih Rebuild Project. Selanjutnya klik tab “Design”, kita bisa melihat background image berhasil kita ganti

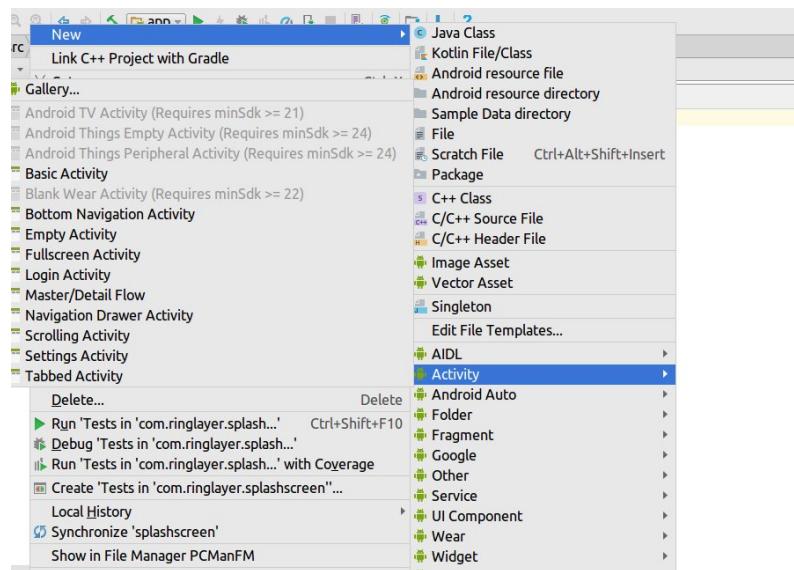


F.2. Membuat Splash Screen

Pada contoh ini kita akan mencoba membuat splash screen untuk aplikasi android yang kita buat.

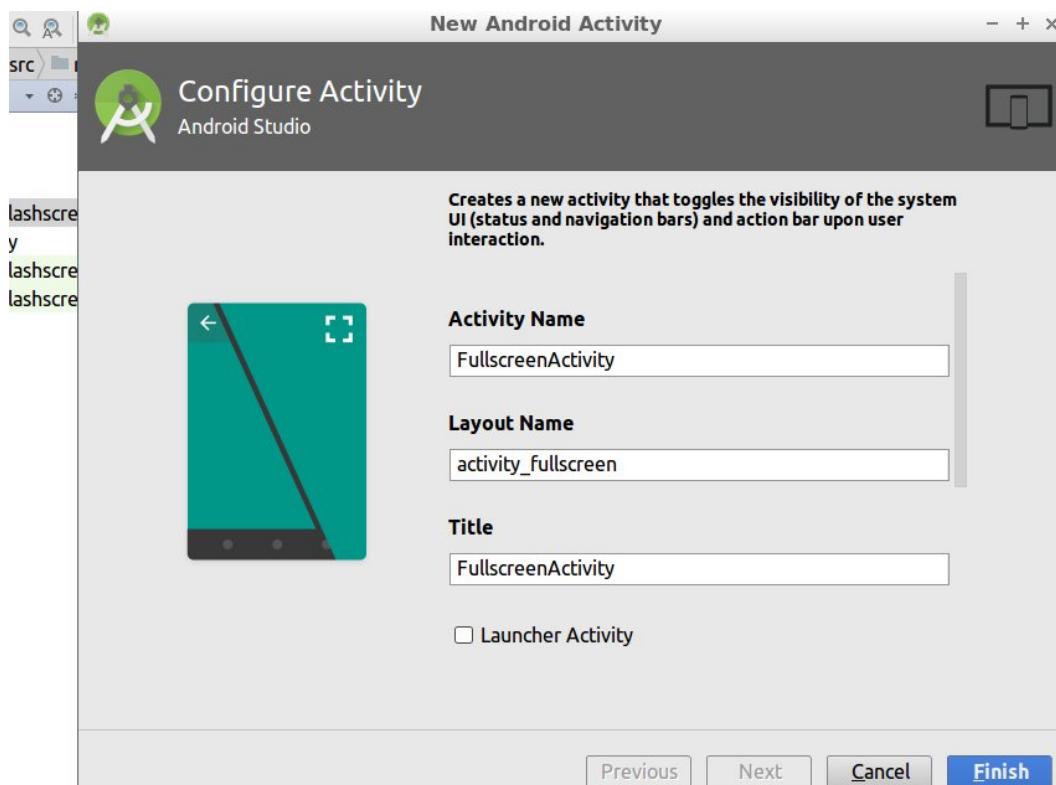
Siapkan proyek baru dengan nama SplashScreen





Selanjutnya tambahkan activity baru

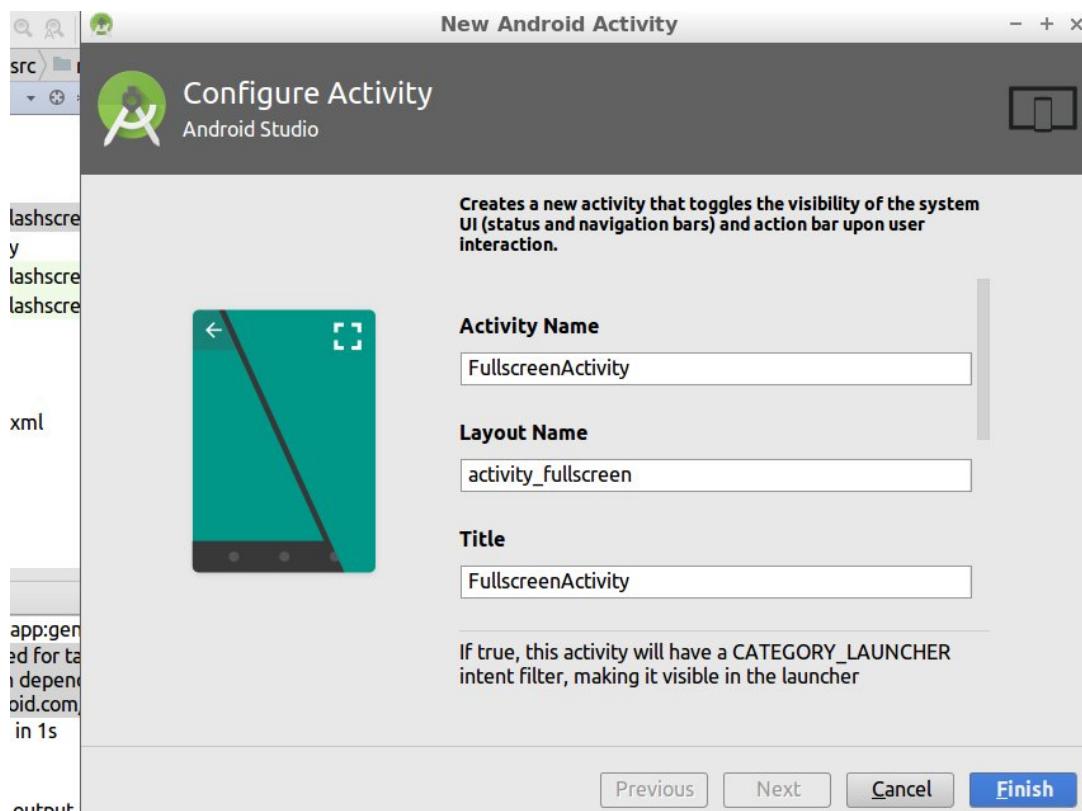
Pilih FullScreenActivity



Centang pada kotak checkbox “Launcher Activity”

Lalu klik Finish.

Langkah selanjutnya adalah kita perlu melakukan sedikit pengeditan agar activity FullScreenActivity ditampilkan secara layar penuh.



Selanjutnya agar layout activity_fullscreen bisa tampil menjadi layar penuh (full screen), siapkan tema untuk tampilan layar penuh. Pada app → res → values → styles.xml, buat style untuk layar penuh, tambahkan style berikut ini :

```
<style name="AppTheme.Fullscreen">
    <item name="android:windowFullscreen">true</item>
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
</style>
```

Berikut ini adalah isi AndroidManifest.xml yang kita gunakan pada proyek ini :

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.splashscreen">
    <application>
```

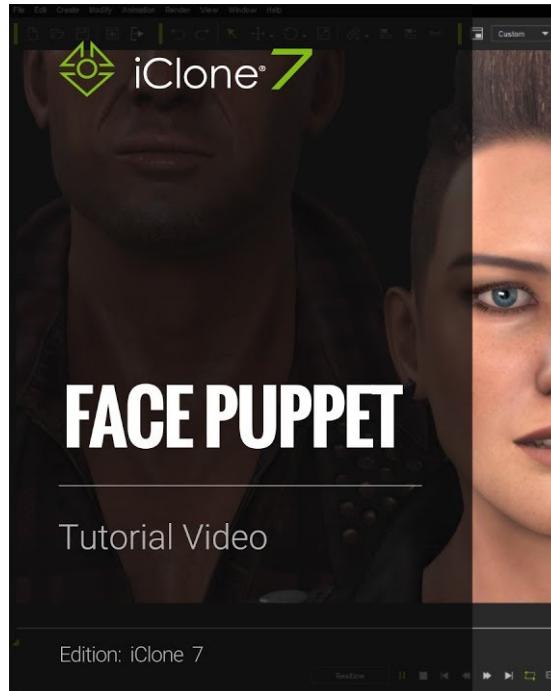
```

    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
</activity>
<activity
    android:name=".FullscreenActivity"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:label="@string/title_activity_fullscreen"
    android:theme="@style/AppTheme.Fullscreen">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>

```

Pada activity_fullscreen,kita akan menggunakan custom background image. Background yang akan digunakan adalah background portrait.

Ini adalah contoh background yang akan digunakan :



Simpan dengan nama file icloud7.jpg di folder app/src/main/res/drawable

Berikut ini isi activity_fullscreen.xml

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/icloud7"
    tools:context="com.ringlayer.splashscreen.FullscreenActivity">

```

```

<!-- The primary full-screen view. This can be replaced with whatever view
    is needed to present your content, e.g. VideoView, SurfaceView,
    TextureView, etc. -->
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">
    <LinearLayout
        android:id="@+id/fullscreen_content_controls"
        style="?metaButtonBarStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|center_horizontal"
        android:background="@color/black_overlay"
        android:orientation="horizontal"
        tools:ignore="UselessParent">
    </LinearLayout>
</FrameLayout>
</FrameLayout>

```

Berikut ini isi activity_main.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
    tools:context="com.ringlayer.splashscreen.MainActivity">
</android.support.constraint.ConstraintLayout>

```

Bagian Source Code

Berikut ini isi FullscreenActivity.java :

```

package com.ringlayer.splashscreen;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
public class FullscreenActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fullscreen);
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(FullscreenActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, 3000);
    }
}

```

PENJELASAN

FullscreenActivity merupakan activity yang berfungsi untuk splash di mana activity ini akan menjalankan MainActivity dengan intent dengan delay 3 detik setelah activity_fullscreen dirender

```
@Override
public void run() {
    Intent intent = new Intent(FullscreenActivity.this, MainActivity.class);
    startActivity(intent);
    finish();
}
,3000);
```

Berikut ini isi MainActivity.java

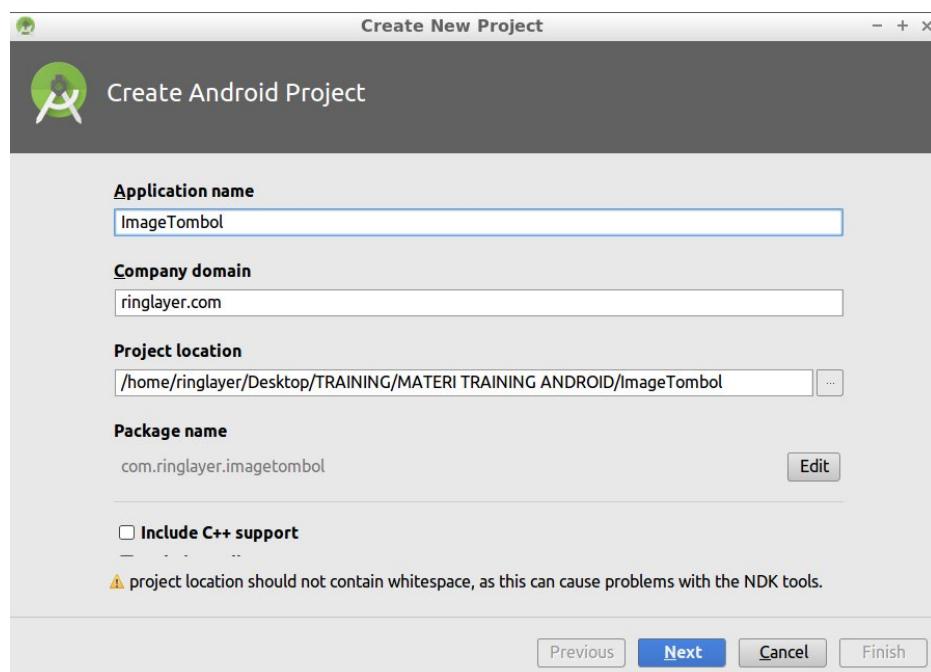
```
package com.ringlayer.splashscreen;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

PENJELASAN

Activity di atas hanyalah berisi source code default yang diciptakan android studio.

F.3. Menggunakan Image pada Tombol

Pada contoh ini kita akan mencoba menggunakan image pada tombol untuk aplikasi android yang kita buat. Buat proyek baru dengan nama “ImageTombol”



Bagian XML

Sebelum mulai, kita persiapkan image yang akan digunakan sebagai tombol di app/src/main/res/drawable , simpan dengan nama imgtombol.png



Pada contoh kali ini kita hanya menggunakan 1 layout : activity_main.xml dengan activitynya MainActivity. Berikut ini isi file activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
        tools:context="com.ringlayer.imagetombol.MainActivity">
    <Button
        android:layout_width="174dp"
        android:layout_height="78dp"
        android:background="@drawable/imgtombol" />
</android.support.constraint.ConstraintLayout>
```

Bagian Source Code

Untuk source code MainActivity tidak dilakukan perubahan, merupakan source code yang digenerate android studio secara default.

MainActivity.java :

```
package com.ringlayer.imagetombol;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

F.4. Debugging Aplikasi

Debugging pada pembuatan aplikasi android bermanfaat untuk melihat log warning, error atau pesan debug yang kita tulis dengan fasilitas log atau pesan pesan aplikasi yang muncul dari sistem android.

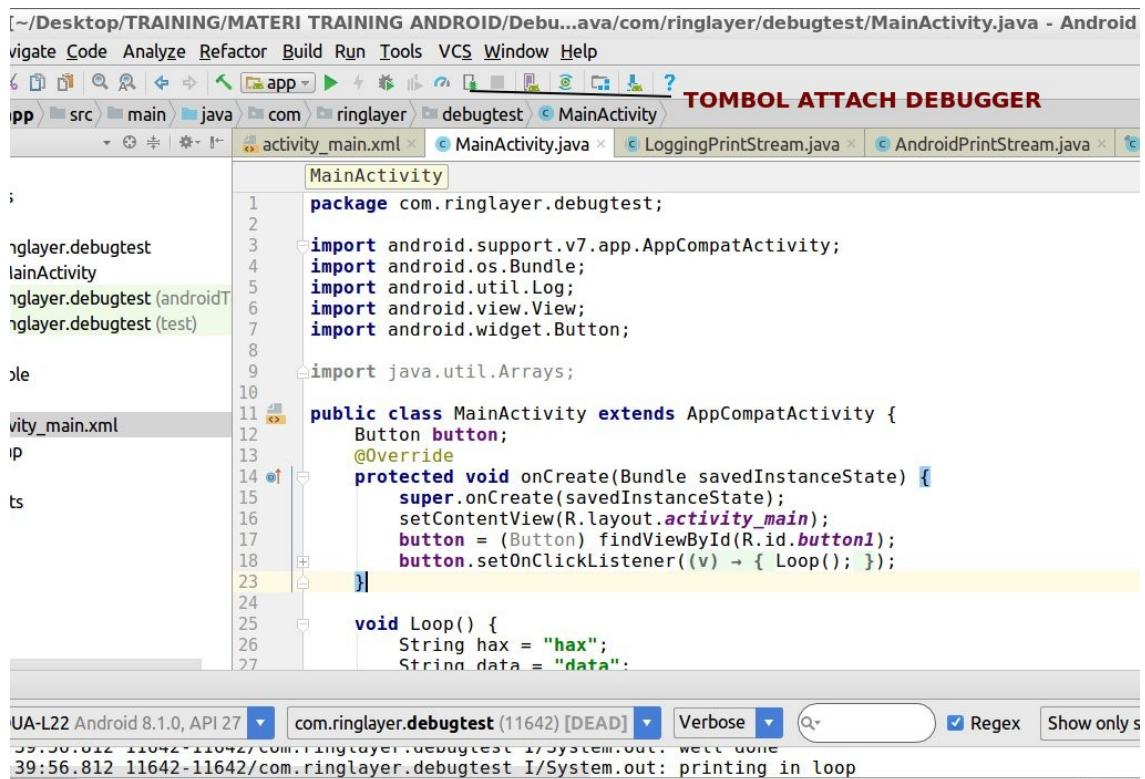
Ada beberapa metode untuk debugging aplikasi android yang sedang kita buat. Untuk mencoba debugging, kita akan membuat proyek baru dengan nama DebugTest dengan 1 activity : MainActivity.

Berikut ini source code MainActivity.java :

```
package com.ringlayer.debugtest;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import java.util.Arrays;
public class MainActivity extends AppCompatActivity {
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Loop();
            }
        });
    }
    void Loop() {
        String hax = "hax";
        String data = "data";
        try {
            int i = 1;
            int x = 0;
            char[] charArray = new char[3];
            while (i < 10) {
                System.out.println("printing in loop");
                i++;
                data += hax;
                charArray[x] = 'A';
                x++;
                System.out.println("well done");
            }
            System.out.println(data);
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
}
```

1. Menggunakan tombol “Attach Debugger”

Metode pertama adalah dengan menggunakan tombol “Attach Debugger”



The screenshot shows the Android Studio interface. At the top, the title bar reads: [~/Desktop/TRAINING/MATERI TRAINING ANDROID/Debu...ava/com/ringlayer/debugtest/MainActivity.java - Android]. Below the title bar is the toolbar with various icons. A red box highlights the 'TOMBOL ATTACH DEBUGGER' button, which is located near the top right of the toolbar. The main area contains the Java code for MainActivity.java:

```
1 package com.ringlayer.debugtest;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.util.Log;
6 import android.view.View;
7 import android.widget.Button;
8
9 import java.util.Arrays;
10
11 public class MainActivity extends AppCompatActivity {
12     Button button;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         button = (Button) findViewById(R.id.button1);
18         button.setOnClickListener(v -> { Loop(); });
19     }
20
21     void Loop() {
22         String hax = "hax";
23         String data = "data";
24     }
25 }
```

Below the code editor is the Logcat window, which displays log messages from the application's process:

```
UA-L22 Android 8.1.0, API 27 com.ringlayer.debugtest (11642) [DEAD] Verbose Regex Show only s
39:56.812 11642-11642/com.ringlayer.debugtest I/System.out: well done
39:56.812 11642-11642/com.ringlayer.debugtest I/System.out: printing in loop
```

Tombol attach debugger ini bisa digunakan untuk mendebug aplikasi yang sedang kita jalankan di ponsel android dengan tombol run android studio.

Berikut ini contoh mendebug aplikasi dengan tombol “attach debugger”, misal kita ingin mendebug pada bagian looping pada source code di atas, tempatkan breakpoint dengan cara klik pada area abu abu di sebelah kanan nomor baris pada baris kode anda ingin memasang breakpoint.

KLIK DI AREA
ABU ABU DI KANAN NOMOR
BARIS

```
23     }
24
25     void Loop() {
26         String hax = "hax";
27         String data = "data";
28         try {
29             int i = 1;
30             int x = 0;
31             char[] charArray = new char[3];
32             while (i < 10) {
33                 System.out.println("printing in loop");
34                 i++;
35                 data += hax;
36                 charArray[x] = 'A';
37                 x++;
38                 System.out.println("well done");
39             }
40         } catch (Exception e) {
41             Log.e(tag: "error", e.toString());
42         }
43     }
44 }
```

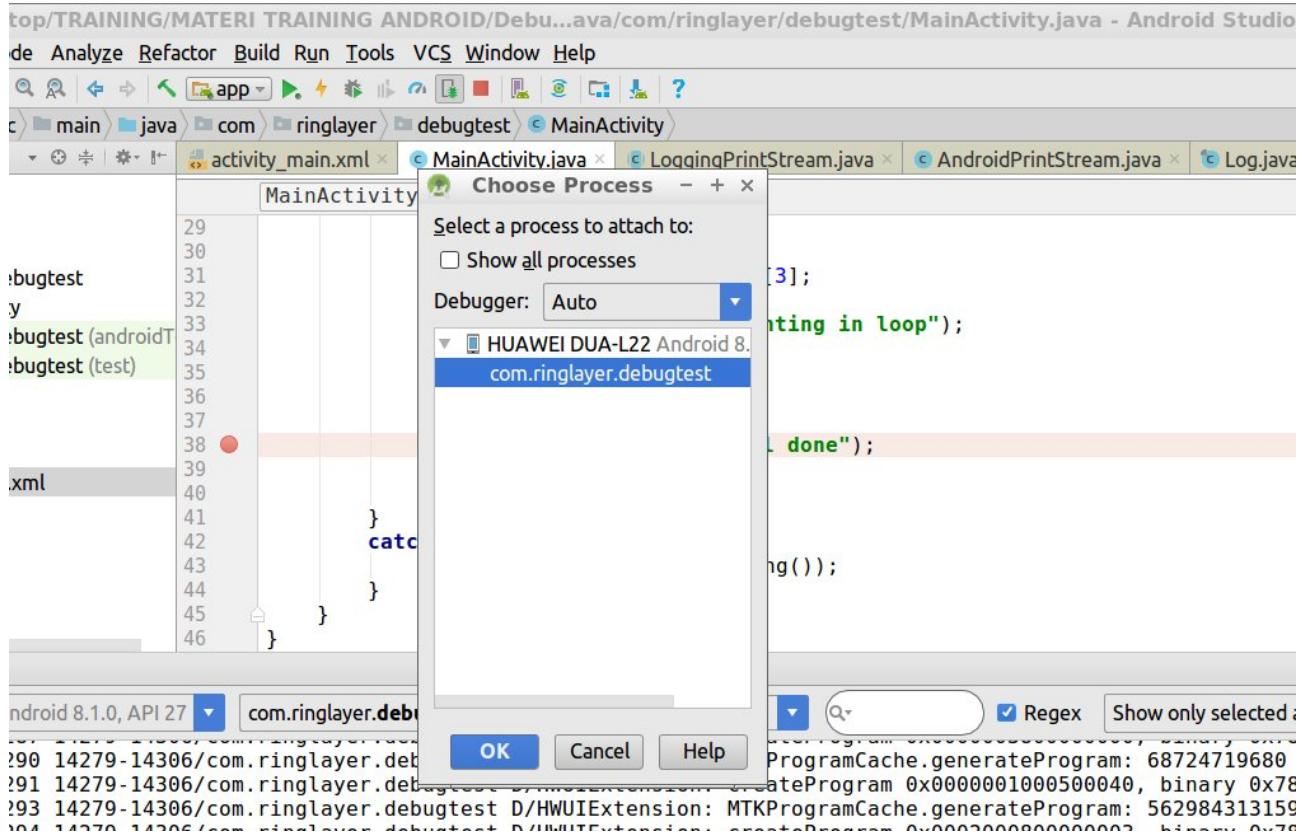
DUA-L22 Android 8.1.0, API 27 com.ringlayer.debugtest (11642) [DEAD] Verbose Regex Show only selected application
159.50.812 11642-11642/com.ringlayer.debugtest I/System.out: well done
159.50.812 11642-11642/com.ringlayer.debugtest I/System.out: printing in loop
Event Log Gradle Con

Setelah breakpoint terpasang pada baris kode yang anda pilih maka akan muncul titik merah. Selanjutnya klik tombol run untuk mulai menjalankan aplikasi ini pada device android yang terhubung dengan kabel data.

KLIK RUN UNTUK MENJALANKAN APLIKASI
DI DEVICE ANDROID

```
23     }
24
25     void Loop() {
26         String hax = "hax";
27         String data = "data";
28         try {
29             int i = 1;
30             int x = 0;
31             char[] charArray = new char[3];
32             while (i < 10) {
33                 System.out.println("printing in loop");
34                 i++;
35                 data += hax;
36                 charArray[x] = 'A';
37                 x++;
38             }
39             System.out.println("well done");
40         } System.out.println(data);
```

Setelah tombol run dijalankan, tunggu hingga aplikasi berjalan di perangkat android, setelah berjalan, klik tombol “attach debugger”



Selanjutnya klik “OK”

Pada contoh aplikasi di atas, baris tempat kita memasang breakpoint terdapat pada method loop, di mana method loop baru terpanggil dengan klik pada tombol, perhatikan bagian ini :

```
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Loop();
    }
});
```

Klik button pada aplikasi yang sedang berjalan (klik button pada perangkat android di button aplikasi yang sedang berjalan), Selanjutnya aplikasi akan berhenti pada line 38 :

The screenshot shows the Android Studio interface. The code editor displays Java code for `MainActivity` with a highlighted loop. The variable inspector shows local variables: `this`, `hax`, `data`, `i`, `x`, and `charArray`. The debugger frames panel shows the stack trace starting from `onClick:20` in `MainActivity`.

```

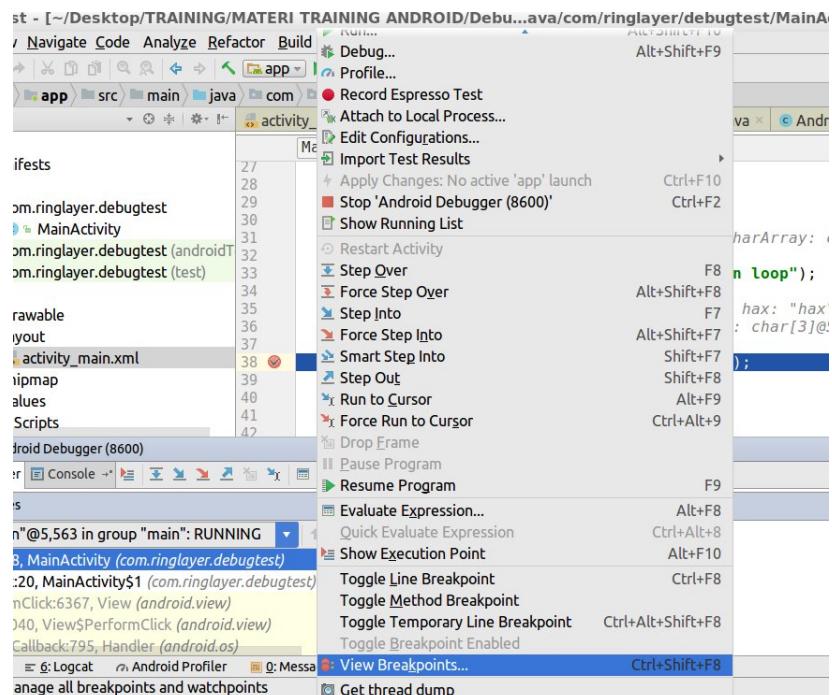
MainActivity | Loop()
27     String data = "data";  data: "datahax"
28     try {
29         int i = 1;  i: 2
30         int x = 0;  x: 1
31         char[] charArray = new char[3];  charArray: char[3]@5572
32         while (i < 10) {
33             System.out.println("printing in loop");
34             i++;  i: 2
35             data += hax;  data: "datahax"  hax: "hax"
36             charArray[x] = 'A';  charArray: char[3]@5572
37             x++;  x: 1
38         }
39         System.out.println("well done");
40     } catch (Exception e) {
41
42

```

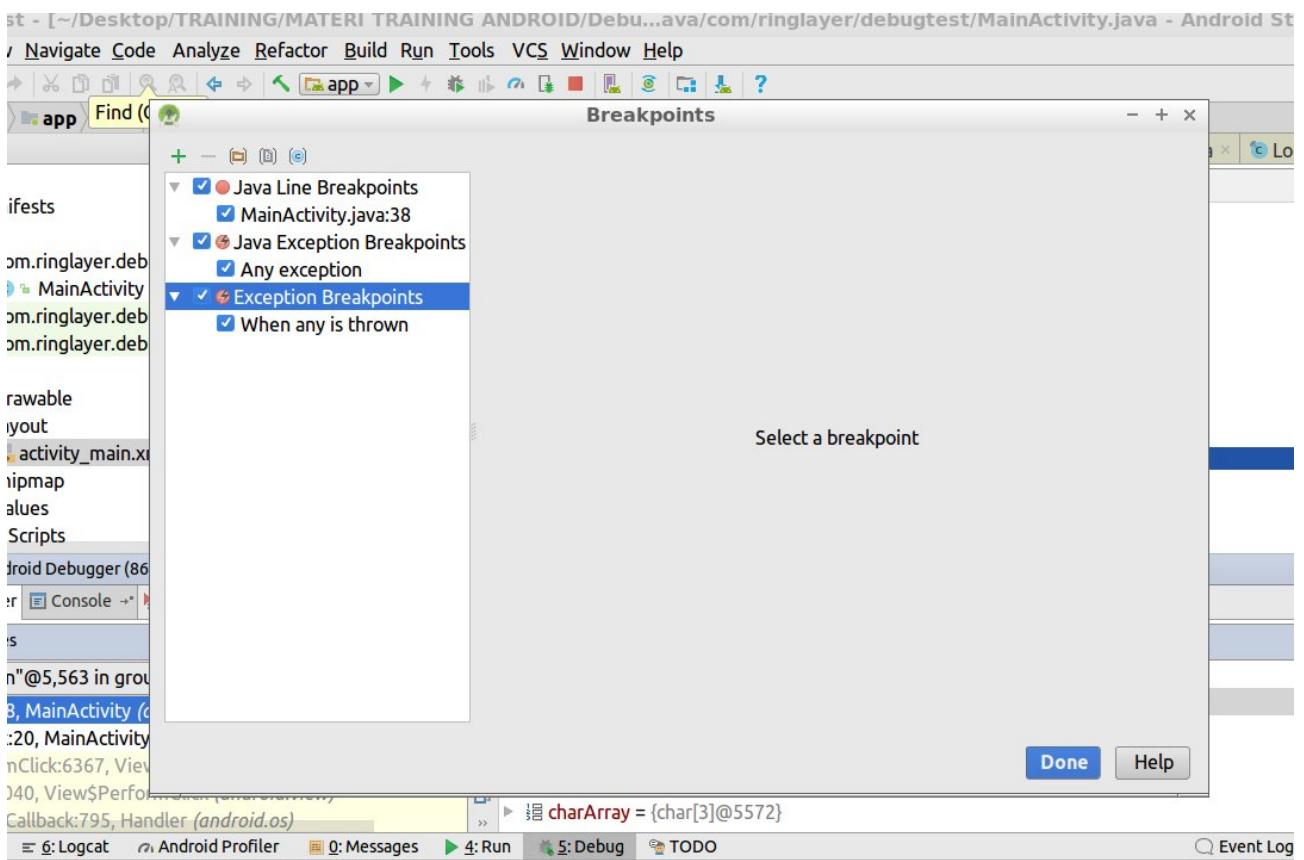
Pada bagian variables kita bisa mengecek pada pengulangan pertama ini, isi dari setiap variabel.

Pada jendela frame di sebelah kiri terlihat kita sedang berhenti di baris 38 pada method loop. Pada baris baris di bawahnya terlihat method yang dieksekusi sebelumnya.

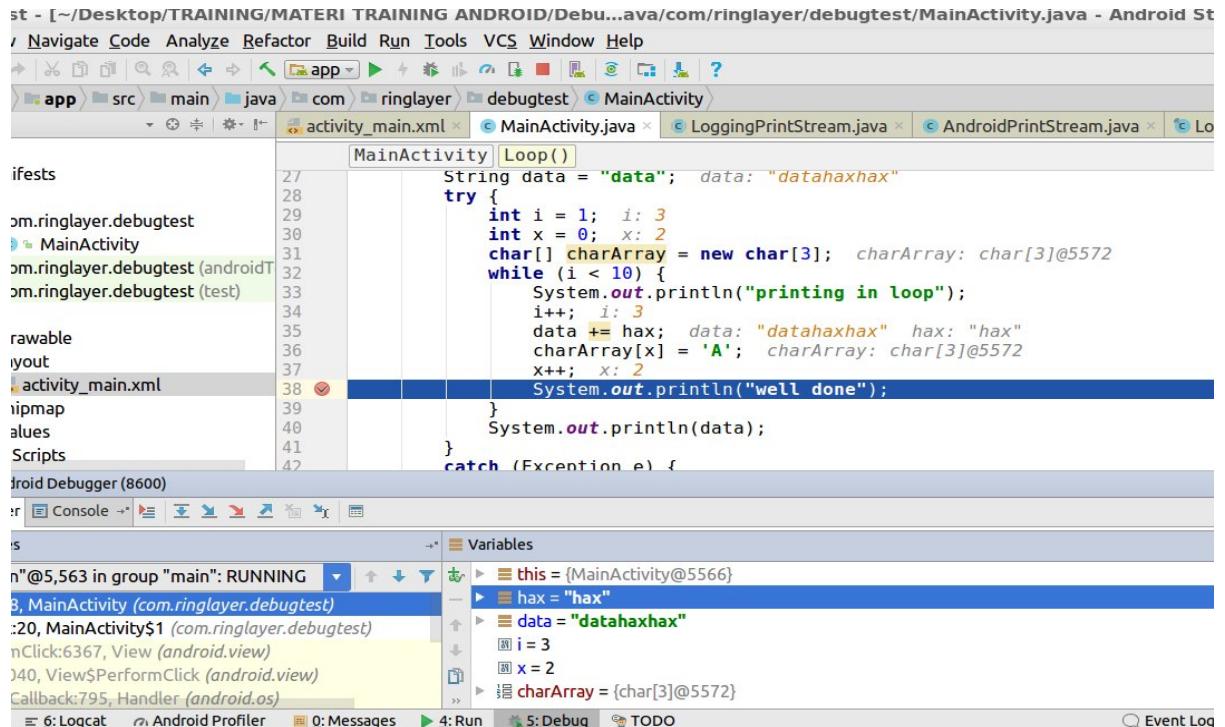
Selanjutnya klik “Run” , pilih “View BreakPoints”



Centang pada “Java Exception Breakpoints” dan “Exception Breakpoints”



Selanjutnya tekan “F9” agar aplikasi lanjut ke pengulangan selanjutnya dan berhenti pada breakpoint selanjutnya



Terlihat ada perubahan perubahan pada isi variabel variabel. Selanjutnya tekan lagi F9 hingga akhirnya terjadi bug buffer overflow di mana terjadi exception breakpoint baru :

The screenshot shows the Android Studio interface. In the top navigation bar, the tabs are: File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help. Below the navigation bar is a toolbar with various icons. The project structure is visible on the left, showing a tree with 'app' (manifests, java, com.ringlayer.debugtest, res, layout, activity_main.xml), 'gradle Scripts', and 'Build Variants'. The main area displays a Java file named 'MainActivity.java' with the following code:

```
String data = "data"; data: "datahaxhaxhaxhax"
try {
    int i = 1; i: 5
    int x = 0; x: 3
    char[] charArray = new char[3]; charArray: char[3]@5572
    while (i < 10) {
        System.out.println("printing in loop");
        i++; i: 5
        data += hax; data: "datahaxhaxhaxhax" hax: "hax"
        charArray[x] = 'A'; charArray: char[3]@5572 x: 3
        x++;
        System.out.println("well done");
    }
    System.out.println(data);
} catch (Exception e) {
```

The line 'charArray[x] = 'A'' is highlighted in blue, indicating it is the current line of execution. The bottom right corner of the code editor has a red circle with a '3', indicating three breakpoints have been set. In the bottom right corner of the screen, there is a status bar with the text 'Android Studio 3.5'.

The bottom half of the screen shows the 'Debug' tab of the Android Debugger. It displays the stack trace:

```
"main"@5,563 in group "main": RUNNING
Loop:36, MainActivity (com.ringlayer.debugtest)
onClick:20, MainActivity$1 (com.ringlayer.debugtest)
performClick:6367, View (android.view)
run:25040, View$PerformClick (android.view)
handleCallback:795, Handler (android.os)
```

On the right side of the debugger, the 'Variables' pane is open, showing the current state of variables:

Variable	Type	Value
this	MainActivity	{MainActivity@5566}
Exception	Exception	{ArrayIndexOutOfBoundsException@5578}
hax	String	"hax"
data	String	"datahaxhaxhaxhax"
i	int	5
x	int	3

Terlihat pada gambar di atas muncul break point baru pada baris ke 36, artinya exception terjadi pada line 36 di mana eksekusi baris tersebut mentrigger bug buffer overflow.

Terlihat terjadi exception error : ArrayIndexOutOfBoundsException.

Jadi kesimpulanya kesalahan pemrograman ditrigger pada baris ke 36 saat buffer yang seharusnya 3 tapi kita isi lebih dari 3 dengan looping.

2. Menggunakan Class Log

Metode kedua adalah dengan memanfaatkan kelas Log dan Logcat. Untuk menguji debugging dengan menggunakan class Log.

Metode-metode dalam class log:

- Log.e , digunakan untuk menampilkan pesan error
- Log.w , digunakan untuk menampilkan warning
- Log.i , digunakan untuk menampilkan pesan yang bersifat informasi
- Log.d , digunakan untuk menampilkan pesan debug
- Log.v , digunakan untuk menampilkan seluruh pesan log

Untuk menguji bug error, kita sudah menambahkan Log.e pada source code di atas :

```

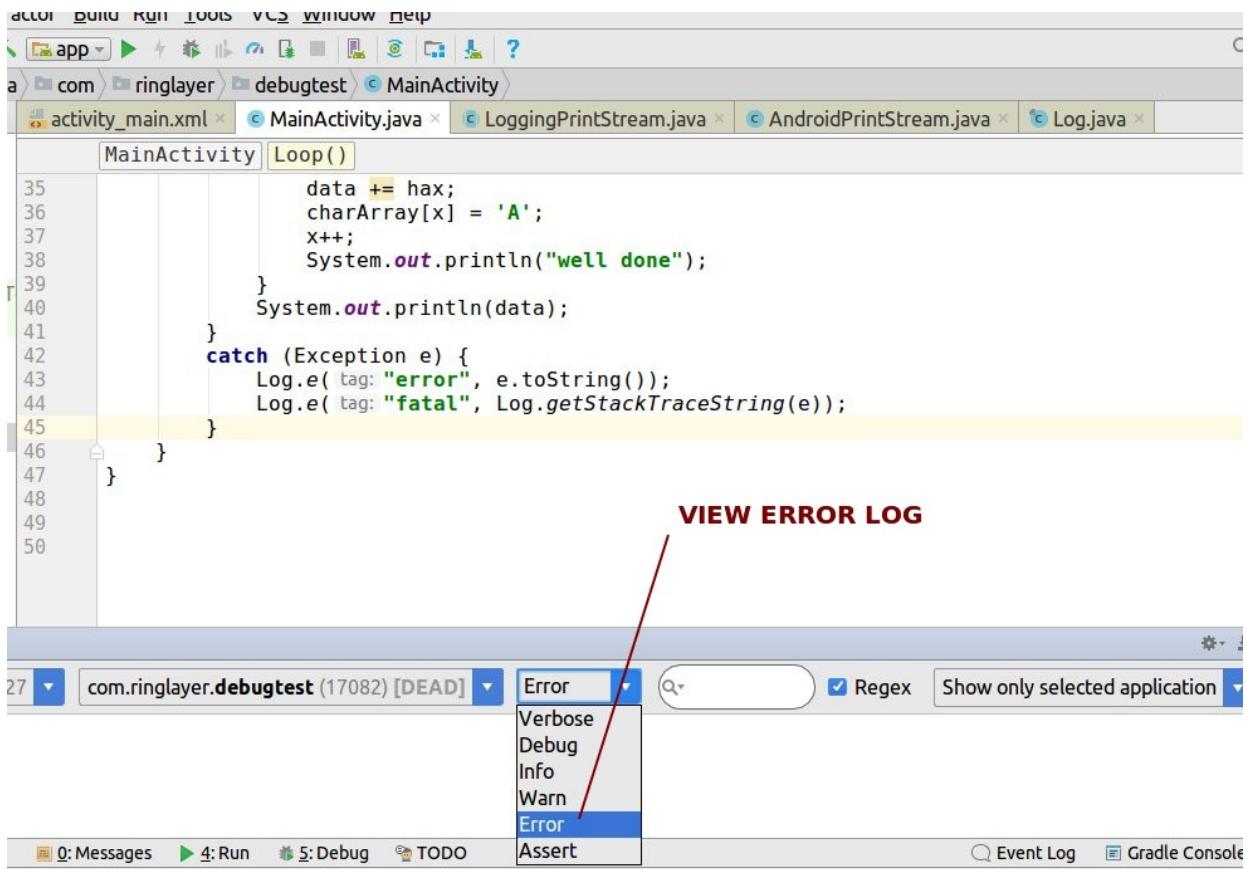
void Loop() {
    String hax = "hax";
    String data = "data";
    try {
        int i = 1;
        int x = 0;
        char[] charArray = new char[3];
        while (i < 10) {
            System.out.println("printing in loop");
            i++;
            data += hax;
            charArray[x] = 'A';
            x++;
            System.out.println("well done");
        }
        System.out.println(data);
    }
    catch (Exception e) {
        Log.e("error", e.toString());
    }
}
}

```

Di mana saat exception terjadi maka akan diprint berupa string, akan tetapi printing dengan Log.e di atas tidaklah lengkap, kita akan menambahkan data yang lebih verbose dengan menampilkan full stack trace. Setelah baris Log.e("error", e.toString()); tambahkan baris ini :

```
Log.e("fatal", Log.getStackTraceString(e));
```

Selanjutnya pada jendela logcat pilih tampilan untuk filter hanya error :



Saat aplikasi dijalankan, klik tombol, maka tidak lama kemudian muncul error dengan lebih verbose:

The Logcat window displays the following detailed error stack trace:

```

04-02 21:39:26.158 17538-17538/com.ringlayer.debugtest E/Minikin: Could not get cmap table size!
04-02 21:39:31.310 17538-17538/com.ringlayer.debugtest E/error: java.lang.ArrayIndexOutOfBoundsException: length=3; index=3
04-02 21:39:31.312 17538-17538/com.ringlayer.debugtest E/fatal: java.lang.ArrayIndexOutOfBoundsException: length=3; index=3
    at com.ringlayer.debugtest.MainActivity$1.onClick(MainActivity.java:20)
    at android.view.View.performClick(View.java:6367)
    at android.view.View$PerformClick.run(View.java:25040)
    at android.os.Handler.handleCallback(Handler.java:795)
    at android.os.Handler.dispatchMessage(Handler.java:99)
    at android.os.Looper.loop(Looper.java:166)
    at android.app.ActivityThread.main(ActivityThread.java:6857)
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:50)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:9)

```

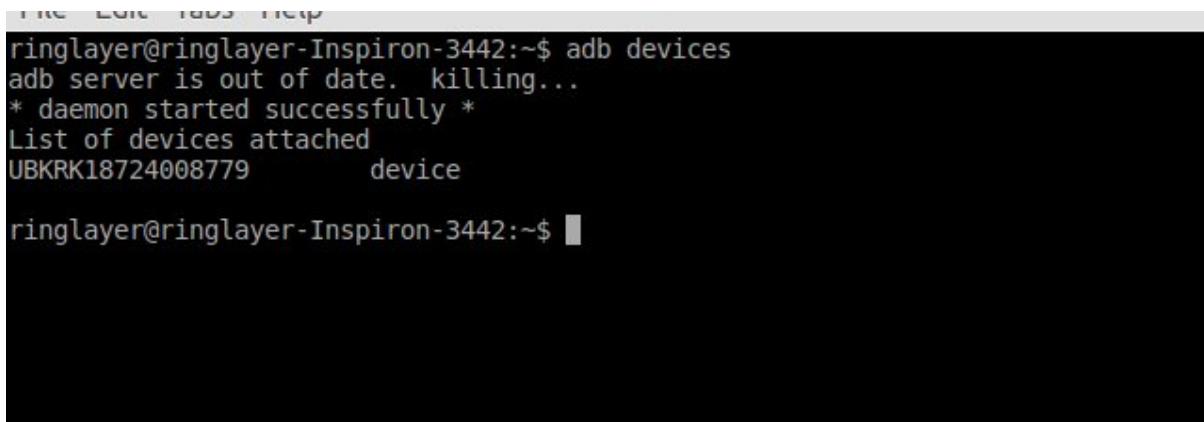
at com.ringlayer.debugtest.MainActivity\$1.onClick(MainActivity.java:20)
at android.view.View.performClick(View.java:6367)

Terlihat bug ditrigger oleh line ke 36 pada MainActivity di mana menyebabkan java.lang.ArrayIndexOutOfBoundsException karena kondisi buffer 3 tapi diisi lebih dari 3 saat pengulangan.

3. Menggunakan ADB

Metode ketiga untuk mendebug aplikasi adalah dengan menggunakan adb. Untuk menggunakan adb, adb harus sudah terinstall di komputer.

Untuk mulai menggunakan adb, hubungkan android dengan komputer melalui kabel data, buka terminal lalu ketik : adb devices (Jika menggunakan windows maka buka ms dos lalu ketik adb devices)



```
ringlayer@ringlayer-Inspiron-3442:~$ adb devices
adb server is out of date.  killing...
* daemon started successfully *
List of devices attached
UBKRK18724008779      device

ringlayer@ringlayer-Inspiron-3442:~$
```

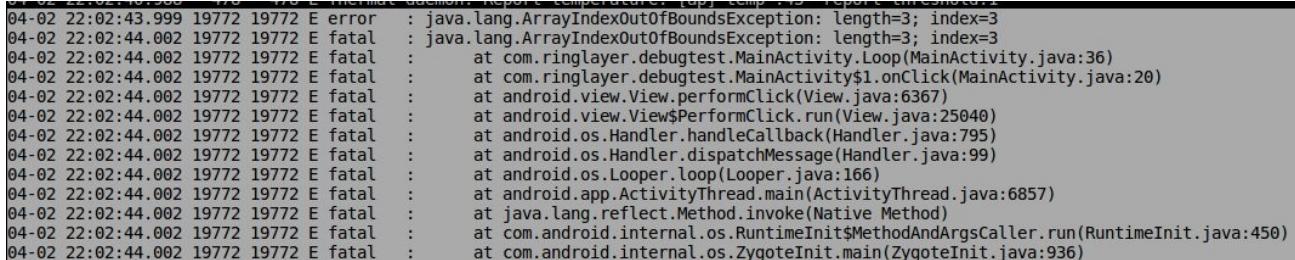
Jika sudah terhubung maka akan muncul tampilan device seperti di atas.

Untuk melihat daftar perintah lengkap adb bisa dicek di <http://adbshell.com/commands/adb-logcat>

Untuk menampilkan hanya error gunakan perintah

```
adb logcat *:E
```

Berikut ini tampilan error yang ditangkap oleh adb yang terjadi pada MainActivity.java di baris 36 :



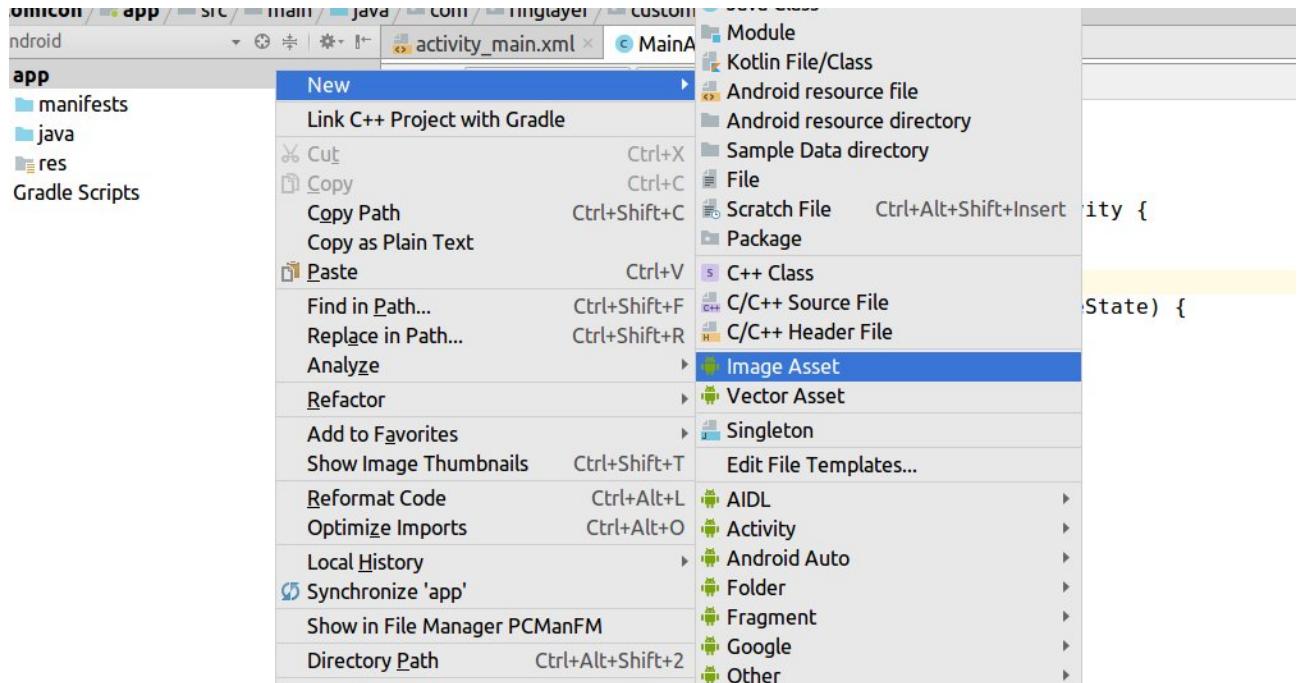
```
04-02 22:02:43.999 19772 19772 E error    : java.lang.ArrayIndexOutOfBoundsException: length=3; index=3
04-02 22:02:44.002 19772 19772 E fatal    : java.lang.ArrayIndexOutOfBoundsException: length=3; index=3
04-02 22:02:44.002 19772 19772 E fatal    : at com.ringlayer.debugtest.MainActivity$Loop(MainActivity.java:36)
04-02 22:02:44.002 19772 19772 E fatal    : at com.ringlayer.debugtest.MainActivity$1.onClick(MainActivity.java:20)
04-02 22:02:44.002 19772 19772 E fatal    : at android.view.View.performClick(View.java:6367)
04-02 22:02:44.002 19772 19772 E fatal    : at android.view.View$PerformClick.run(View.java:25040)
04-02 22:02:44.002 19772 19772 E fatal    : at android.os.Handler.handleCallback(Handler.java:795)
04-02 22:02:44.002 19772 19772 E fatal    : at android.os.Handler.dispatchMessage(Handler.java:99)
04-02 22:02:44.002 19772 19772 E fatal    : at android.os.Looper.loop(Looper.java:166)
04-02 22:02:44.002 19772 19772 E fatal    : at android.app.ActivityThread.main(ActivityThread.java:6857)
04-02 22:02:44.002 19772 19772 E fatal    : at java.lang.reflect.Method.invoke(Native Method)
04-02 22:02:44.002 19772 19772 E fatal    : at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:450)
04-02 22:02:44.002 19772 19772 E fatal    : at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:936)
```

F.5. Mengganti Ikon Default Launcher pada Aplikasi

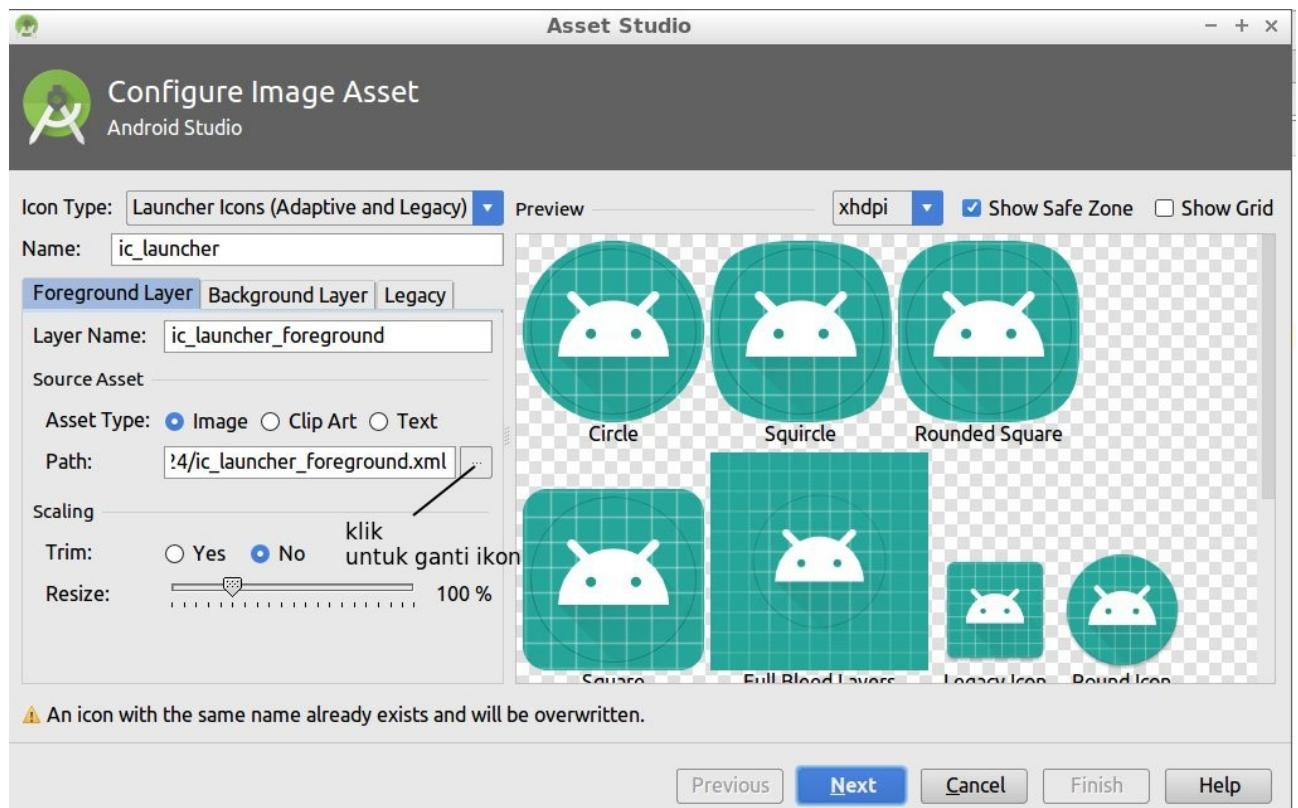
Kita bisa mengganti ikon default launcher aplikasi android kita di android studio. Pertama tama anda harus menyiapkan image yang akan anda gunakan sebagai pengganti launcher ikon default, bisa menggunakan image logo png atau jpg.

Untuk contoh projek aplikasi yang akan kita ganti ikon defaultnya, buat dahulu proyek baru dengan nama : “CustomIcon”

Untuk mengganti icon launcher, klik kanan pada app → new → Image Asset

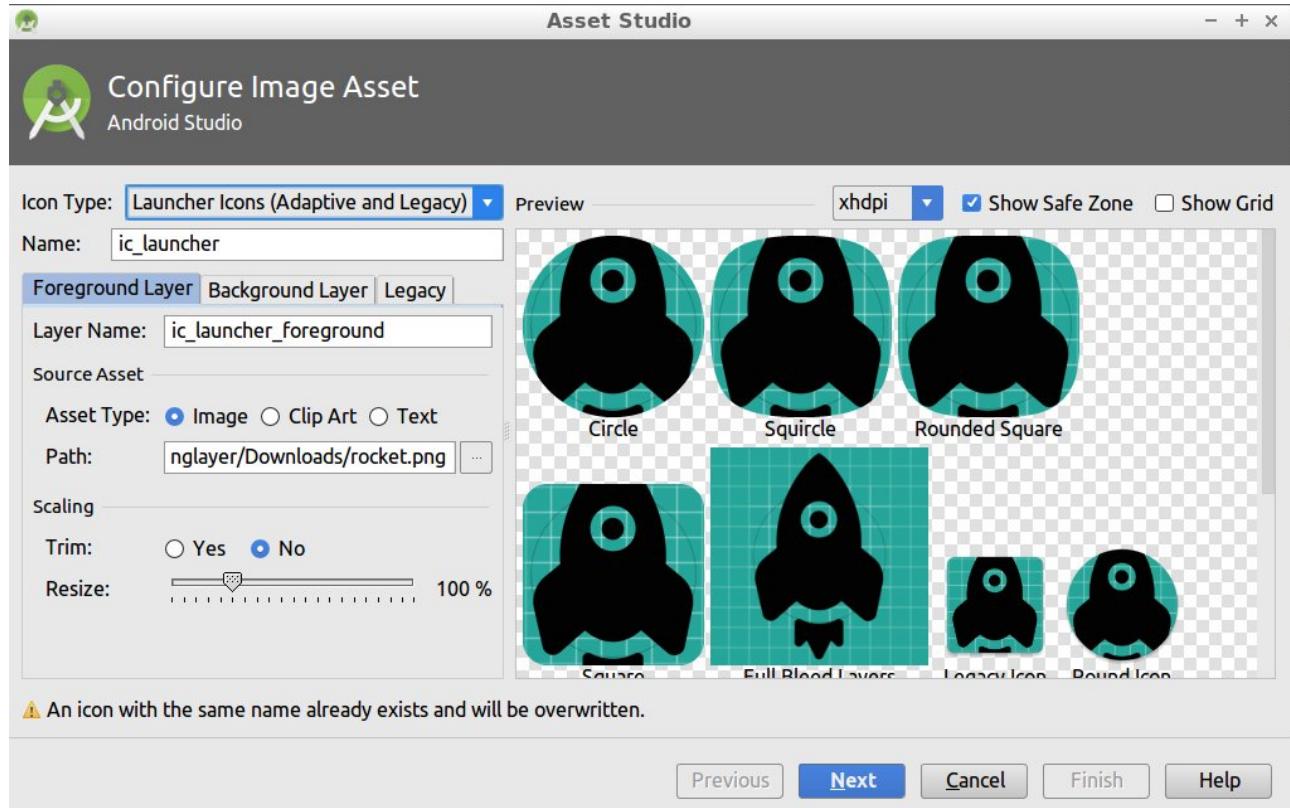


Selanjutnya akan keluar window asset studio:

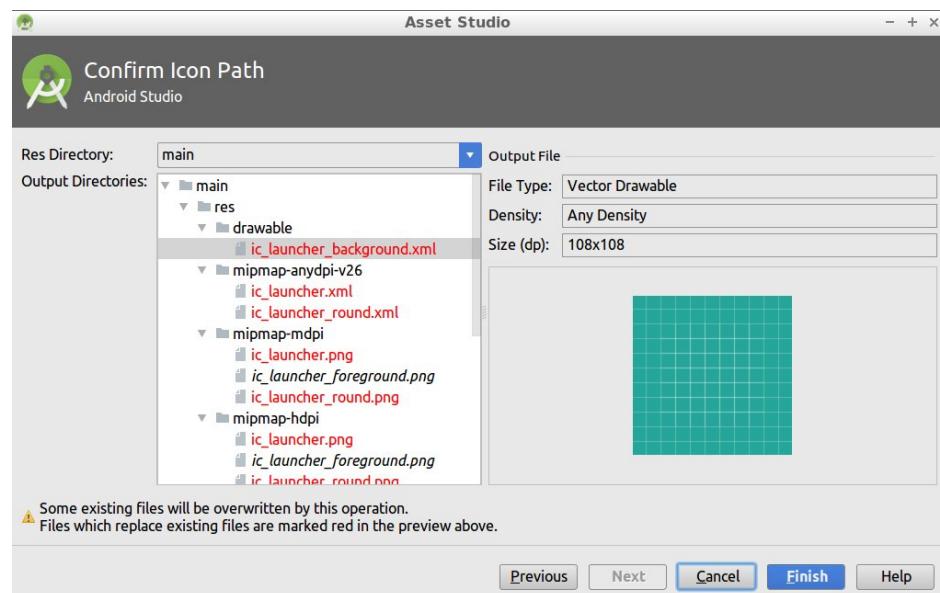


Klik pada tombol di samping path untuk mengganti ikon image. Pilih image dari komputer

Pada contoh ini kita ganti launcher menjadi image roket:



Selanjutnya klik next



Klik pada tombol finish.

G. Contoh Aplikasi dengan Toast

Toast merupakan pesan notifikasi yang bisa kita tampilkan pada aplikasi android. Untuk membuat contoh aplikasi dengan toast, buat proyek baru dengan nama proyek : “ToastSample”

Bagian XML

Berikut ini isi file MainActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
    tools:context="com.ringlayer.toastsample.MainActivity">
</android.support.constraint.ConstraintLayout>
```

Pada contoh ini kita akan menggunakan constraint layout.

Bagian Source Code

Isikan kode berikut ini pada MainActivity.java :

```
package com.ringlayer.toastsample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    private Toast mToastToShow;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        _popup("Contoh menampilkan toast di android");
    }
    public void _popup(String pesan) {
        try {
            mToastToShow = Toast.makeText(getApplicationContext(), pesan,
Toast.LENGTH_LONG);
            mToastToShow.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL,
0, 0);
            new Thread(new Runnable() {
                public void run() {
                    try {
                        Thread.sleep(4000);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    mToastToShow.show();
                }
            }).start();
        }
    }
}
```

```
        catch(Exception e) {
            Log.e("error", e.toString());
        }
    }
}
```

Perhatian ! Sesuakan nama package dengan nama package Anda, pada line ini :

```
package com.ringlayer.toastsample;
```

Penjelasan

Pada source di atas, pertama kali kita lakukan import berbagai kelas yang akan kita gunakan pada proyek ini, kelas kelas ini merupakan turunan dari `java.lang.Object` yang merupakan parent dari semua kelas lainnya.

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.widget.Toast;
```

Selanjutnya kita buat kelas baru dengan nama `MainActivity` yang merupakan kelas turunan dari parent class `AppCompatActivity`. Karena kelas `MainActivity` yang kita buat ini merupakan turunan dari `AppCompatActivity`, maka kelas ini akan memiliki karakteristik bawaan dari kelas `AppCompatActivity`.

```
public class MainActivity extends AppCompatActivity {
```

Selanjutnya, kita lakukan override pada callback method `OnCreate`

```
@Override
protected void onCreate(Bundle savedInstanceState) {
```

Selanjutnya kita menggunakan keyword `super`

```
super.onCreate(savedInstanceState);
```

Dengan keyword `super` di sini metode callback `OnCreate` yang kita override akan dijalankan terlebih dahulu barulah menjalankan rutin yang kita tambahkan di bawahnya.

```
setContentView(R.layout.activity_main);
```

Selanjutnya baris di atas digunakan untuk merender `activity_main.xml`

```
_popup("Contoh menampilkan toast di android");
```

Metode `_popup` merupakan metode yang kita buat untuk menampilkan pesan popup berupa `Toast`.

H. Contoh Aplikasi Spinner

Aplikasi spinner merupakan komponen yang sering digunakan saat membuat formulir isian pada aplikasi android. Komponen ini berguna untuk menampilkan pilihan pada isian formulir yang bisa dipilih user.

Berikut ini adalah contoh pembuatan spinner. Buka android studio lalu start new project dengan nama SpinnerDemo

Bagian XML

Pertama tama isikan strings.xml berikut ini di res → values → strings.xml

```
<resources>
    <string name="app_name">SpinnerDemo</string>
    <string name="pilih">pilih</string>
    <string-array name="array_pilihan">
        <item>Kue</item>
        <item>Cake</item>
        <item>Susu</item>
    </string-array>
</resources>
```

Berikut ini isi lengkap Strings.xml

```
<resources>
    <string name="app_name">SpinnerDemo</string>
    <string name="pilih">pilih</string>
    <string-array name="array_pilihan">
        <item>Kue</item>
        <item>Cake</item>
        <item>Susu</item>
    </string-array>
</resources>
```

Berikut ini contoh isi activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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"
    tools:context="com.ringlayer.spinnerdemo.MainActivity">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:padding="10dip"
        android:gravity="center_vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Spinner
            android:layout_gravity="center_vertical"
            android:id="@+id/spinner"
            android:entries="@array/array_pilihan"
            android:layout_width="fill_parent"
```

```

        android:layout_height="wrap_content"
        android:prompt="@string/pilih"/>
    <Button
        android:layout_gravity="center_vertical"
        android:id="@+id/tombol"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/pilih"></Button>
</LinearLayout>
</android.support.constraint.ConstraintLayout>
```

Perhatian !!! pada bagian :

tools:context="com.ringlayer.spinnerdemo.MainActivity"

Pastikan Anda sesuaikan dengan nama package dan activity Anda.

Pada baris ini kita menggunakan spinner :

```

<Spinner
    android:layout_gravity="center_vertical"
    android:id="@+id/spinner"
    android:entries="@array/array_pilihan"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:prompt="@string/pilih"/>
```

Spinner merupakan komponen user interface yang digunakan untuk menampilkan pilihan. Spinner di atas mengambil data array “array_pilihan” dari strings untuk kemudian ditampilkan.

Bagian Source Code

Langkah akhir, berikut ini isi MainActivity.java :

```

package com.ringlayer.spinnerdemo;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private Toast mToastToShow;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final Spinner spinner;
```

```

        spinner = (Spinner) findViewById(R.id.spinner);

        Button tombol = (Button) findViewById(R.id.tombol);
        tombol.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                _popup(spinner.getSelectedItem().toString());
            }
        });
    }

    public void _popup(String pesan) {
        try {
            mToastToShow = Toast.makeText(getApplicationContext(), pesan,
            Toast.LENGTH_LONG);
            mToastToShow.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL,
            0, 0);

            new Thread(new Runnable() {
                public void run() {
                    try {
                        Thread.sleep(4000);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    mToastToShow.show();
                }
            }).start();
        }
        catch(Exception e) {
            Log.e("error", e.toString());
        }
    }
}

```

Perhatian ! Sesuaikan nama package dengan nama package Anda, pada line ini :

```
package com.ringlayer.spinnerdemo;
```

Pada baris ini :

```
final Spinner spinner;
```

Kita deklarasi objek dengan jenis Spinner dengan nama spinner. Di sini menggunakan keyword final . Keyword final saat digunakan pada deklarasi kelas menyebabkan kelas tersebut tidak bisa diturunkan, sedangkan pada contoh ini kita gunakan pada deklarasi variabel yang artinya variabel tersebut sudah tidak bisa diubah.

```
spinner = (Spinner) findViewById(R.id.spinner);
```

Pada baris di atas kita inisialisasi objek spinner dengan referensi dari resource layout saat ini, di mana metode `findViewById` digunakan mencari referensi di layout dengan id spinner.

(Spinner)

fungsinya digunakan untuk melakukan casting dari referensi yang telah ditemukan agar berupa objek Spinner.

```
Button tombol = (Button) findViewById(R.id.tombol);
```

Pada baris diatas kita mencari referensi dengan id tombol

```
tombol.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        _popup(spinner.getSelectedItem().toString());  
    }  
});
```

Setelah didapat referensi dengan id tombol maka kita register OnClickListener yang akan ditrigger saat user mengklik tombol.

```
_popup(spinner.getSelectedItem().toString());
```

Fungsinya adalah menampilkan pesan toast data di spinner yang dipilih.

I. Contoh Aplikasi DatePicker

Aplikasi datepicker merupakan komponen yang sering digunakan saat pembuatan form pada aplikasi android. Komponen ini berguna untuk menampilkan user interface yang berguna untuk menampilkan pilihan tanggal di form.

Pada contoh kali ini, kita akan membuat aplikasi datepicker sederhana. Pertama tama buat proyek aplikasi android baru dengan nama `datepicker`

Bagian XML

Berikut ini contoh layout activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/background"
    tools:context="com.ringlayer.datepicker.MainActivity"
    android:gravity="top|center_horizontal">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="50dp"></TextView>
    <EditText
        android:focusable="false"
        android:paddingLeft="10dp"
        android:id="@+id/tgl"
        android:layout_width="300dp"
        android:layout_height="45dp"
        android:layout_gravity="center"
        android:inputType="textPersonName"
        android:hint="Tanggal"
        android:textColor="@color/colorAccent" />
</LinearLayout>
```

Perhatian !

Pada bagian edittext pastikan fokus dimatikan agar keyboard bisa digantikan dengan listener on click:

`android:focusable="false"`

Bagian Source Code

Pada bagian MainActivity isikan kode ini:

```
package com.ringlayer.datepicker;
import android.app.DatePickerDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.widget.DatePicker;
import android.widget.EditText;
```

```
import java.util.Calendar;
import static android.view.View.GONE;
public class MainActivity extends AppCompatActivity {
    EditText tgl;
    DatePickerDialog picker;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tgl = (EditText)findViewById(R.id.tgl);
        tgl.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final Calendar cldr = Calendar.getInstance();
                int day = cldr.get(Calendar.DAY_OF_MONTH);
                int month = cldr.get(Calendar.MONTH);
                int year = cldr.get(Calendar.YEAR);
                // date picker dialog
                picker = new DatePickerDialog(MainActivity.this,
                    new DatePickerDialog.OnDateSetListener() {
                        @Override
                        public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {
                            tgl.setText(dayOfMonth + "/" + (monthOfYear + 1) + "/" +
year);
                        }
                    }, year, month, day);
                picker.show();
            }
        });
    }
}
```

Keterangan kode java:

EditText **tgl**;

berguna untuk membuat objek berupa edittext dengan nama tgl

DatePickerDialog **picker**;

Berguna untuk membuat objek berupa datepickerdialog

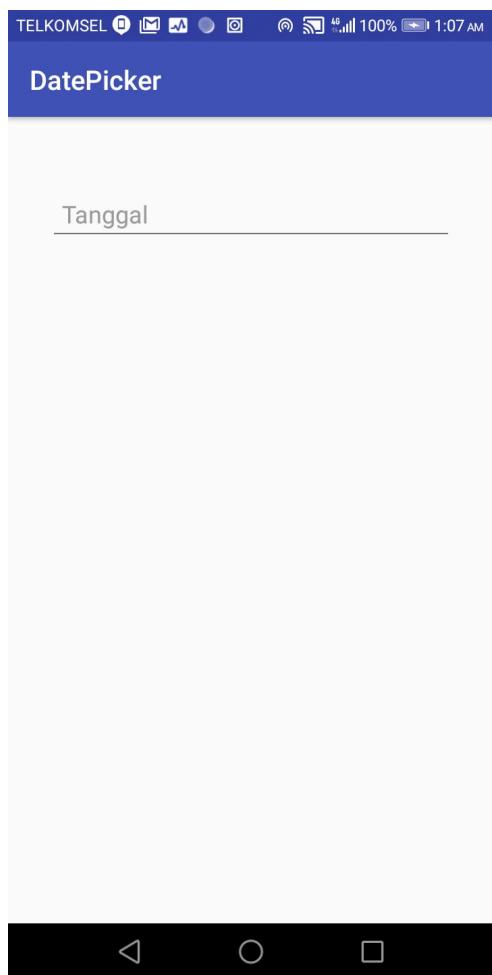
tgl = (EditText)findViewById(R.id.**tgl**);

Kode ini berguna untuk mencari view yang berupa objek edittext di dalam layout dengan id : tgl

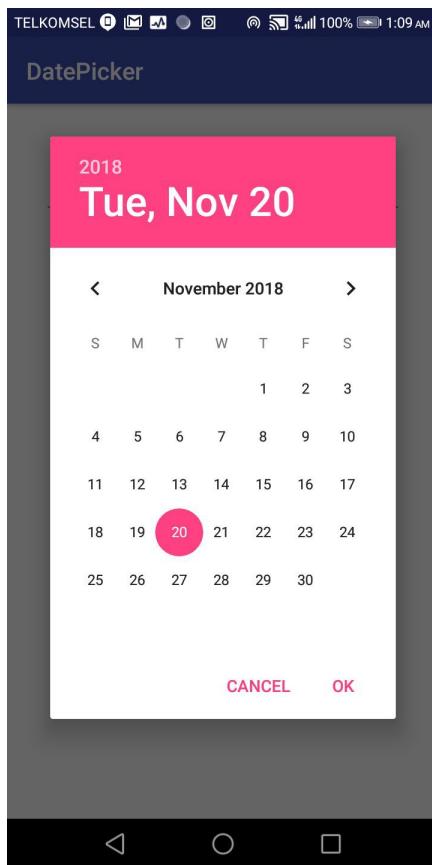
Selanjutnya isikan implementasi metode setOnClickListener agar objek edittext tersebut saat diklik menampilkan dialog datepicker :

```
tgl.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        final Calendar cldr = Calendar.getInstance();
        int day = cldr.get(Calendar.DAY_OF_MONTH);
        int month = cldr.get(Calendar.MONTH);
        int year = cldr.get(Calendar.YEAR);
        // date picker dialog
        picker = new DatePickerDialog(MainActivity.this,
            new DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
                    tgl.setText(dayOfMonth + "/" + (monthOfYear + 1) + "/" + year);
                }
            }, year, month, day);
        picker.show();
    }
});
```

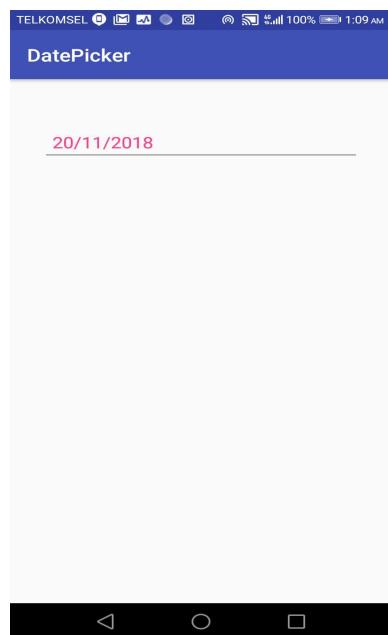
Jalankan aplikasi di atas, jika berhasil maka berikut ini tampilan awalnya :



Untuk menguji onclicklistener, klik pada edittext , jika aplikasi berjalan sempurna akan muncul kotak dialog pemilihan tanggal (datepicker dialog)



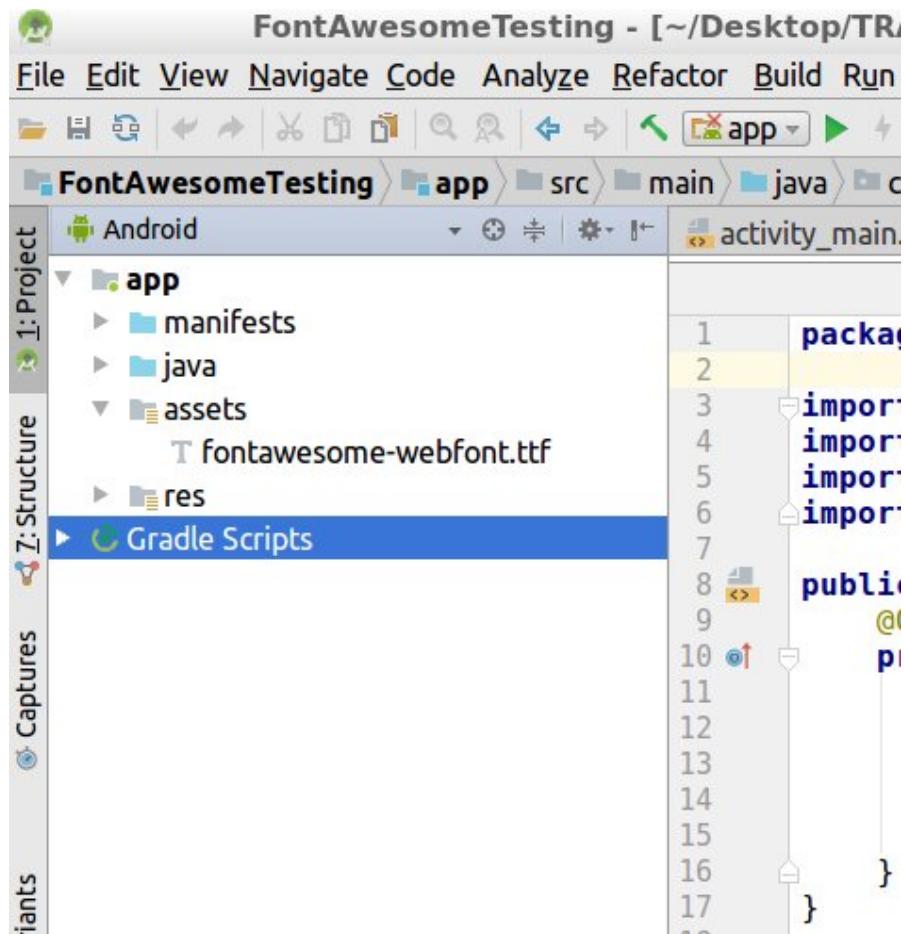
Pilih tanggal, lalu klik ok, maka datepicker akan mengisikan kotak edittext dengan tanggal terpilih



J. Font Awesome di Aplikasi Android

Font awesome merupakan kumpulan set ikon yang berguna untuk mempercantik tampilan komponen komponen layout pada aplikasi android. Pada contoh kali ini, kita akan menggunakan fontawesome pada tombol / button.

Pertama tama buat proyek baru dengan nama FontAwesomeTesting. Lalu buat folder dengan nama assets pada app :



Download font awesome ikon dari :

<https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/fonts/fontawesome-webfont.ttf>

Lalu simpan di folder assets yang sudah kita siapkan sebelumnya.

Bagian XML

Selanjutnya isikan strings.xml dengan isi berikut ini :

```
<resources>
    <string name="app_name">FontAwesomeTesting</string>
```

```
<string name="font_awesome_unlock_icon_and_string">✉</string>
</resources>
```

Perhatikan ! Pada contoh di atas kita menggunakan kode font awesome f09c. Untuk melihat kode kode lengkap yang bisa digunakan, Anda bisa melihat daftar lengkapnya di

<https://fontawesome.com/cheatsheet>

Langkah selanjutnya siapkan isi layout activity_main.xml berikut ini :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:="com.ringlayer.fontawesometesting.MainActivity">
    <LinearLayout
        android:gravity="center_horizontal"
        android:paddingTop="50dp"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/tombol_unlock"
            android:layout_width="300dp"
            android:layout_height="50dp"
            android:layout_gravity="center"
            android:textColor="@color/colorAccent"
            android:text="@string/font_awesome_unlock_icon_and_string" />
    </LinearLayout>
</LinearLayout>
```

Bagian Source Code

Selanjutnya, isikan kode berikut ini pada MainActivity.java :

```
package com.ringlayer.fontawesometesting;

import android.graphics.Typeface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Typeface fontAwesomeFont = Typeface.createFromAsset(getAssets(), "fontawesome-webfont.ttf");
        Button fontAwesomeUnLockAndString = (Button) findViewById(R.id.tombol_unlock);
        fontAwesomeUnLockAndString.setTypeface(fontAwesomeFont);
    }
}
```

Perhatian ! Sesuaikan nama package dengan nama package Anda, pada line ini :

```
package com.ringlayer.fontawesometesting;
```

```
Typeface fontAwesomeFont = Typeface.createFromAsset(getAssets(), "fontawesome-webfont.ttf");
```

Baris ini digunakan untuk memanggil metode `createFromAsset` pada kelas `Typeface`, metode ini akan membuat objek `typeface` dari data font yang dispesifikasi di parameter.

```
Button fontAwesomeUnLockAndString = (Button) findViewById(R.id.tombol_unlock);
```

Baris ini digunakan untuk mencari referensi di layout dengan id `tombol_unlock` kemudian diinisialisasi di objek berupa `Button` dengan nama `fontAwesomeUnLockAndString`.

```
fontAwesomeUnLockAndString.setTypeface(fontAwesomeFont);
```

Baris ini digunakan untuk memanggil metode `setTypeface` pada objek `fontAwesomeUnLockAndString` di mana pada metode ini resource dari font yang sebelumnya kita spesifikasi di baris diatasnya akan dirender.