# Materi 3: Activities dan Intents

## Pokok Bahasan

1. Mengenal cara menggunakan Activity
2. Mengimplementasikan Intent dengan menggunakan Explicit Intent
3. Melewatkan variabel antar Activity

## Tujuan

Di Akhir pertemuan ini, diharapkan mahasiswa dapat mengenal dan menggunakan Activity disertai Explicit dan Implicit Intent yang digunakan pada program/ aplikasi yang dibuat. Selain itu, mahasiswa mengetahui bagaimana cara kerja pengembangan Aplikasi Android dengan mengetahui lifecycle pada saat aplikasi/ program itu dibuat.

## Penjelasan Singkat

Activity merupakan komponen dari aplikasi yang merepresentasikan dari satu tampilan/ window dimana dalam tampilan tersebut terdapat aksi yang dilakukan seperti mengambil foto, mengirimkan pesan, membuka map, dan sebagainya.

Didalam sebuah aplikasi membuat banyak tampilan (multiple screen). Setiap tampilan tersebut mewakili satu Activity. Secara default, tampilan yang pertama kali pada saat project dibuat menghasilkan main activity dengan nama MainActivity.java. Activity tersebut dijalankan pada saat pertama kali project dijalankan. Activity tersebut dapat terhubung dengan Activity lainnya dengan cara menggunakan Intents.

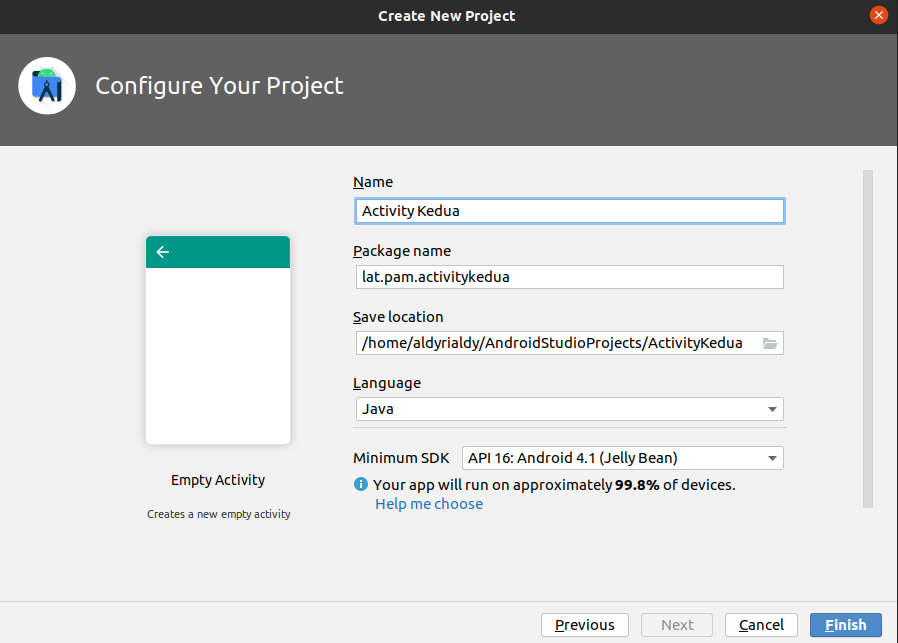
Sedangkan Intents merupakan objek yang digunakan untuk meminta tindakan dari komponen aplikasi lain melalui sistem Android. Misalnya tindakan untuk menghubungkan satu activity ke Activity yang lain. Terdapat dua jenis Intent yakni Explicit Intent dan Implicit Intent.

1. Explicit intents secara spesifik menyebutkan target (nama Activity), seperti mengarahkan halaman utama (Main activity) ke halaman untuk lihat barang yang sudah dibeli pada halaman Shopping Cart (ViewShoppingCart Activity).
2. Implicit intents tidak secara spesifik menyebutkan target (nama Activity), hanya mendefinisikan aksi yang akan direspon oleh Android runtime.

Pada modul pertama ini anda diminta membuat sebuah aplikasi sederhana dengan dua activity. Untuk menghubungkan activity satu dengan activity kedua, anda akan menggunakan Intent yang melewatkan parameter (passing parameter). Kemudian, anda akan mengirim data balik lagi kepada activity pertama.

## Modul 1: Membuat activity pertama dan menampilkan Log

1. Hal pertama yang perlu dilakukan pada praktikum kali ini anda dapat membuat project terlebih dahulu dan pilihlah template Empty Activity seperti pada tampilan berikut. Jika sudah maka beri nama dengan Activity Kedua, seperti pada tampilan berikut ini.



1. Langkah selanjutnya buka project tersebut dan pilihlah activity\_main.xml yang terletak pada app > res > layout.
2. Kemudian hapuslah TextView yang ada pada tampilan tersebut dan masukkan Button kedalam project tersebut. Anda dapat menuliskan atribut dari Button yang dibuat sepeti berikut.

<Button

android:id="@+id/button\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:text="@string/button\_main"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintRight\_toRightOf="parent" />

1. Selanjutnya definisikan resource value untuk string button\_main dengan tulisan “Kirim”. Sehingga atribut <string name="button\_main">Kirim</string> ada pada string.xml.
2. Tambahkan atribut onClick yang diisi dengan launchSecondActivity. Sehingga secara keseluruhan maka Button akan seperti berikut.

<Button

android:id="@+id/button\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="16dp"

android:layout\_marginRight="16dp"

android:text="@string/button\_main"

android:onClick="launchSecondActivity"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintRight\_toRightOf="parent" />

1. Selanjutnya, buatlah method baru dengan nama launchSecondActivity pada MainActivity.java. Anda dapat membuatnya dengan menekan Alt+Enter pada onclick tersebut secara otomatis tersebut dapat dibuat.
2. Pada bagian atas dari class MainActivity tambahkan kode berikut:

private static final String LOG\_TAG = MainActivity.class.getSimpleName();

1. Tambahkan kode Log.*d*(*LOG\_TAG*, "Button clicked!"); pada method yang baru dibuat tadi, sehingga kode program java tersebut seperti berikut.

package lat.pam.activitykedua;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

public class MainActivity extends AppCompatActivity {

private static final String *LOG\_TAG* =

MainActivity.class.getSimpleName();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

}

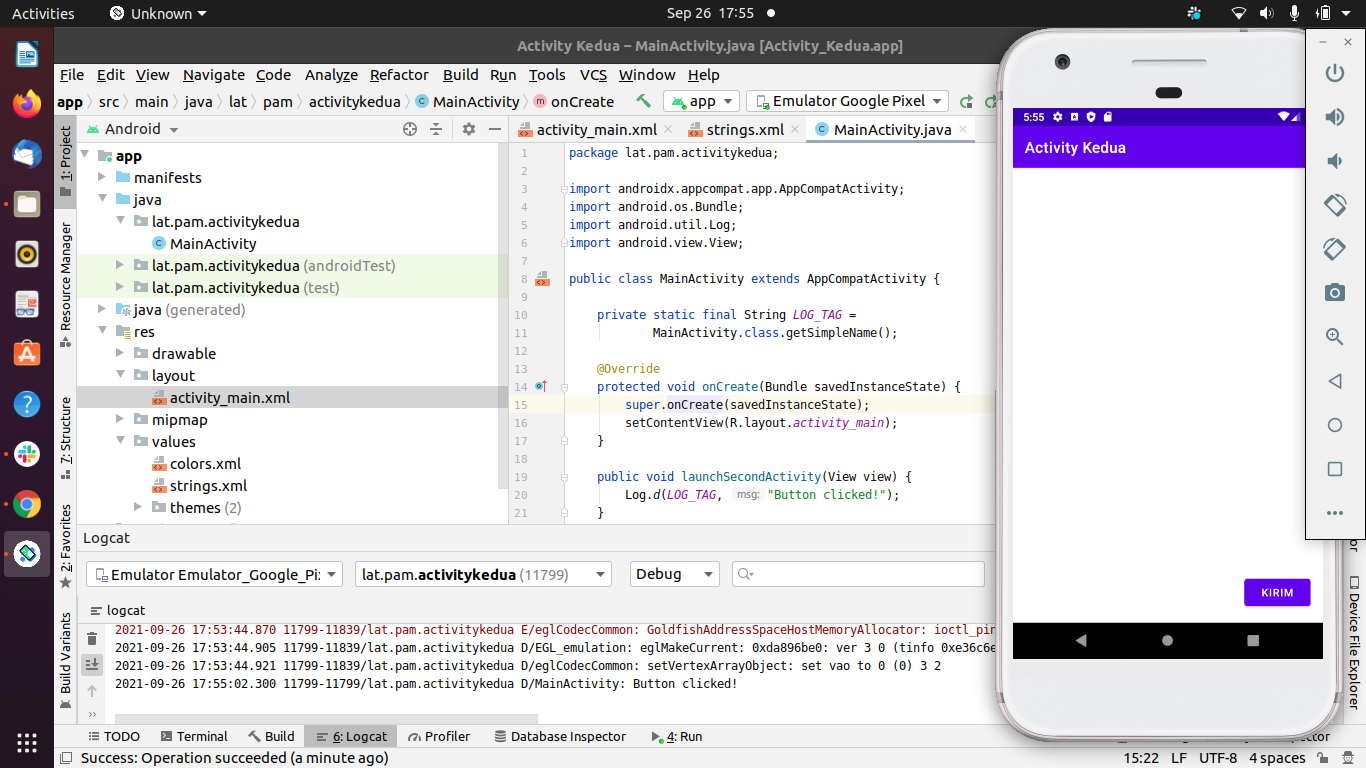
public void launchSecondActivity(View view) {

Log.*d*(*LOG\_TAG*, "Button clicked!");

}

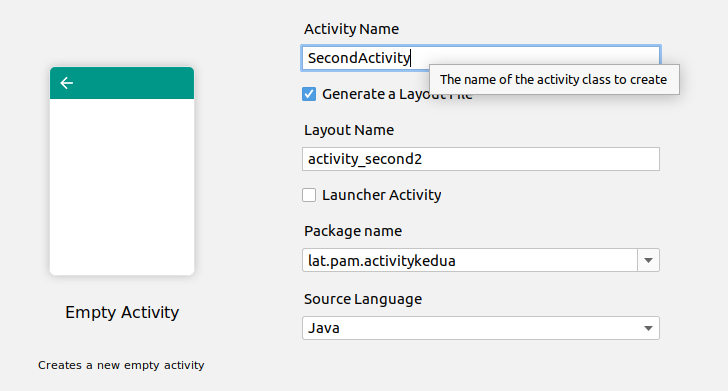
}

1. Jalankan program tersebut dan pastikan log tersebut muncul pada Logcat seperti pada gambar ini yang ditandai warna kotak berwarna merah berikut.



## Modul 2: Membuat Activity kedua dan menambahkan Intent pada MainActivity

1. Untuk membuat activity yang kedua dapat dilakukan dengan melakukan klik kanan pada package, kemudian pilih New > Activity > Empty Activity. Setelah itu, buatlah satu Activity baru dengan nama SecondActivity.



1. Tambahkan atribut TextView pada layout activity\_second.xml yang sudah dibuat dengan menambahkan kode berikut.

<TextView

android:id="@+id/text\_header"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginTop="16dp"

android:text="@string/text\_header"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"

android:textStyle="bold"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

1. Tambahkan properti string text\_header pada string.xml, sehingga pada string.xml dituliskan sebagai berikut.

<resources>

<string name="app\_name">Activity Kedua</string>

<string name="button\_main">Kirim</string>

<string name="text\_header">Message Received</string>

</resources>

1. Kemudian buka activity\_main.xml dan tambahkan kembali EditText pada tampilan layout tersebut. Sehingga tampilan keseluruhan pada activity\_main.xml sebagai berikut.

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

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<Button

android:id="@+id/button\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="16dp"

android:layout\_marginRight="16dp"

android:text="@string/button\_main"

android:onClick="launchSecondActivity"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintRight\_toRightOf="parent" />

<EditText

android:id="@+id/editText\_main"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="16dp"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:ems="10"

android:hint="@string/editText\_main"

android:inputType="textLongMessage"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toStartOf="@+id/button\_main"

app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

1. Tambahkan variabel EXTRA\_MESSAGE pada MainActivity.java yang ditulis sebagai berikut.

public static final String *EXTRA\_MESSAGE* =

"com.example.android.twoactivities.extra.MESSAGE";

1. Kemudian tambahkan variabel EditText pada activity\_main.xml.

private EditText mMessageEditText;

1. Inisialisasikan variabel tersebut dengan atribut edit text pada activity\_main.xml. Dengan menambahkan kode pada onCreate.

mMessageEditText = findViewById(R.id.*editText\_main*);

1. Setelah itu, tambahkan kode program ini pada launchSecondActivity sehingga kode programnya sebagai berikut.

public void launchSecondActivity(View view) {

Log.*d*(*LOG\_TAG*, "Button clicked!");

Intent intent = new Intent(this, SecondActivity.class);

String message = mMessageEditText.getText().toString();

intent.putExtra(*EXTRA\_MESSAGE*, message);

startActivity(intent);

}

1. Tambahkan atribut tersebut pada activity\_second.xml dengan menambahkan TextView, sehingga pada kode XML tertulis sebagai berikut.

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

<androidx.constraintlayout.widget.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=".SecondActivity">

<TextView

android:id="@+id/text\_header"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="16dp"

android:text="@string/text\_header"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"

android:textStyle="bold"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

android:layout\_marginLeft="8dp" />

<TextView

android:id="@+id/text\_message"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/text\_header"

android:layout\_marginLeft="8dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

1. Tambahkan kode program ini pada OnCreate untuk mengambil parameter yang dikirimkan dari MainActivity.java. Sehingga pada method OnCreate ditulis sebagai berikut.

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_second*);

Intent intent = getIntent();

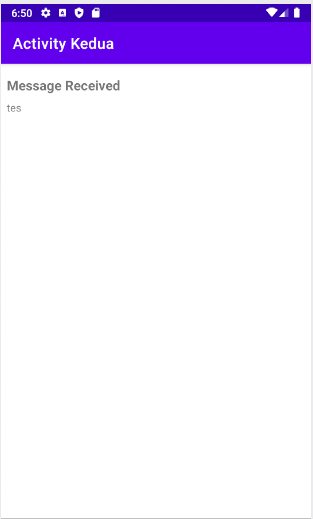
String message = intent.getStringExtra(MainActivity.*EXTRA\_MESSAGE*);

TextView textView = findViewById(R.id.*text\_message*);

textView.setText(message);

}

1. Jalankan aplikasi tersebut dan pastikan tampilan pertama (MainActivity.java) dapat mengirimkan pesan kepada tampilan kedua (SecondActivity.java)

## Modul 3: Menambahkan respon balik ke MainActivity

1. Pada modul ke 3 ini masih pada project yang sama, anda dapat menambahkan string terlebih dahulu pada string.xml. Berikut kode xml yang ditambahkan.

<string name="button\_second">Reply</string>

<string name="editText\_second">Enter Your Reply Here</string>

1. Pada activity\_second.xml tersebut tambahkan pula atribut EditText dan Button untuk mengirimkan respon balik kepada main activity. Berikut kode program yang ditambahkan kedalam activity\_second.xml.

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

<androidx.constraintlayout.widget.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=".SecondActivity">

<TextView

android:id="@+id/text\_header"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginTop="16dp"

android:text="@string/text\_header"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"

android:textStyle="bold"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/text\_message"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginTop="8dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/text\_header" />

<Button

android:id="@+id/button\_second"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:onClick="returnReply"

android:text="@string/button\_second"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintRight\_toRightOf="parent" />

<EditText

android:id="@+id/editText\_second"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="16dp"

android:ems="10"

android:hint="@string/editText\_second"

android:inputType="textLongMessage"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toStartOf="@+id/button\_second"

app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

1. Kemudian, pada SecondActivity.java tambahkan kode untuk mendefinisikan variabel static EXTRA\_REPLY dan EditText.

public static final String *EXTRA\_REPLY* =

"com.example.android.twoactivities.extra.REPLY";

private EditText mReply;

1. Selanjutnya, pada method OnCreate, anda dapat menambahkan untuk memanggil EditText tersebut dengan menambahkan kode program berikut.

mReply = findViewById(R.id.*editText\_second*);

1. Tambahkan pula kode program didalam method returnReply, sehingga method tersebut menjadi sebagai berikut.

public void returnReply(View view) {

String reply = mReply.getText().toString();

Intent replyIntent = new Intent();

replyIntent.putExtra(*EXTRA\_REPLY*, reply);

setResult(*RESULT\_OK*, replyIntent);

finish();

}

1. Tambahkan string tambahan header reply pada string.xml.

<string name="text\_header\_reply">Reply Received</string>

1. Lakukan copy textView dari second\_activity.xml kepada activity\_main.xml dan ubahlah id TextView tersebut, serta tambahkan atribut android:visibility="invisible", sehingga kode xml menjadi berikut.

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

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/text\_header\_reply"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginTop="16dp"

android:text="@string/text\_header\_reply"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"

android:textStyle="bold"

android:visibility="invisible"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/text\_message\_reply"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginTop="8dp"

android:visibility="invisible"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/text\_header\_reply" />

<Button

android:id="@+id/button\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="16dp"

android:layout\_marginRight="16dp"

android:text="@string/button\_main"

android:onClick="launchSecondActivity"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintRight\_toRightOf="parent" />

<EditText

android:id="@+id/editText\_main"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="16dp"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:ems="10"

android:hint="@string/editText\_main"

android:inputType="textLongMessage"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toStartOf="@+id/button\_main"

app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

1. Untuk mendapatkan respon balik dari SecondActivity, kita perlu menambahkan sedikit program dengan cara tambahkan konstanta dan definisikan TextView pada MainActivity.

public static final int *TEXT\_REQUEST* = 1;

private TextView mReplyHeadTextView;

private TextView mReplyTextView;

1. Tambahkan kode program ini untuk memanggil atribut TextView pada OnCreate.

mReplyHeadTextView = findViewById(R.id.*text\_header\_reply*);

mReplyTextView = findViewById(R.id.*text\_message\_reply*);

1. Kemudian ubahlah pada bagian startActivity menjadi

startActivityForResult(intent, *TEXT\_REQUEST*);

1. Tambahkan method onActivityResult seperti dibawah ini.

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == *TEXT\_REQUEST*) {

if (resultCode == *RESULT\_OK*) {

String reply =

data.getStringExtra(SecondActivity.*EXTRA\_REPLY*);

mReplyHeadTextView.setVisibility(View.*VISIBLE*);

mReplyTextView.setText(reply);

mReplyTextView.setVisibility(View.*VISIBLE*);

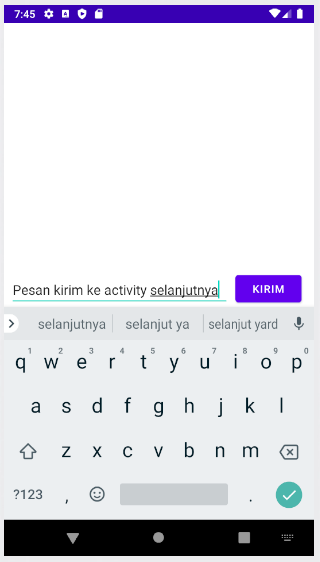
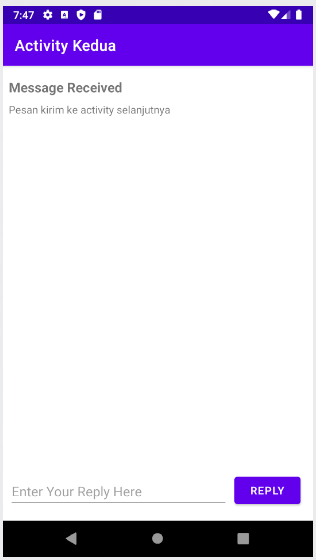
}

}

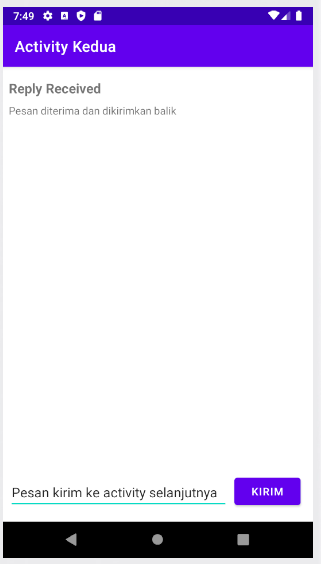
}

1. Jalankan aplikasi ini dan cobalah pada emulator atau perangkat anda.
2. Berikut tampilan dari project setelah dieksekusi.

Tampilan 1: Tampilan 2:

Tampilan 3:



1. Untuk selengkapnya project ini dapat anda unduh pada link berikut [[link]](https://drive.google.com/file/d/1Hk6oesFBpKs0aftsrciEABQK6TB_4qFz/view?usp=sharing).