

## 2.1: Membuat dan Memulai Aktivitas

### Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Membuat proyek `TwoActivities`
- Tugas 2. Membuat dan meluncurkan aktivitas kedua
- Tugas 3. Mengirim data dari aktivitas utama ke aktivitas kedua
- Tugas 4. Mengembalikan data ke aktivitas utama
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Sebuah aktivitas mewakili satu layar di aplikasi, yang digunakan pengguna untuk melakukan satu tugas terfokus seperti menelepon, mengambil foto, mengirim email, atau melihat peta. Aktivitas biasanya ditampilkan ke pengguna dalam jendela layar penuh.

Sebuah aplikasi biasanya terdiri atas beberapa aktivitas yang terikat secara longgar satu sama lain. Biasanya, satu aktivitas dalam aplikasi ditetapkan sebagai aktivitas "utama", yang ditampilkan kepada pengguna saat membuka aplikasi. Tiap aktivitas kemudian bisa memulai aktivitas lain untuk melakukan berbagai tindakan.

Tiap kali aktivitas baru dimulai, aktivitas sebelumnya akan dihentikan, namun sistem mempertahankan aktivitas dalam sebuah tumpukan ("back stack"). Saat sebuah aktivitas baru dimulai, aktivitas baru itu akan didorong ke atas back-stack dan mengambil fokus pengguna. Back-stack mematuhi mekanisme dasar tumpukan "masuk terakhir, keluar pertama", jadi jika pengguna selesai dengan aktivitas saat ini dan menekan tombol Back, aktivitas saat ini akan dikeluarkan dari tumpukan (dan dimusnahkan) dan aktivitas sebelumnya akan dilanjutkan.

Aktivitas Android dimulai atau diaktifkan dengan *intent*. Intent adalah pesan asinkron yang bisa Anda gunakan dalam aktivitas untuk meminta tindakan dari aktivitas lain (atau komponen aplikasi lainnya). Anda menggunakan intent untuk memulai satu aktivitas dari aktivitas lainnya dan meneruskan data di antara aktivitas.

Ada dua jenis intent: *eksplisit* dan *implisit*. Intent eksplisit adalah intent yang targetnya Anda ketahui. Anda sudah mengetahui nama kelas yang sepenuhnya memenuhi syarat dari aktivitas spesifik tersebut. Intent implisit adalah intent yang nama komponen targetnya tidak Anda ketahui, namun memiliki tindakan umum untuk dikerjakan. Dalam praktik ini Anda akan belajar tentang intent eksplisit. Anda akan mempelajari tentang intent implisit pada praktik berikutnya.

### Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat dan menjalankan aplikasi dalam Android Studio.
- Membuat dan mengedit elemen UI dengan Layout Editor grafis atau langsung di file layout XML.
- Menambahkan fungsionalitas `onClick` ke tombol.

### Yang akan Anda PELAJARI

Anda akan belajar:

- Membuat aktivitas baru di Android Studio.
- Mendefinisikan aktivitas induk dan anak untuk navigasi "Up".

- Memulai aktivitas dengan intent eksplisit.
- Meneruskan data di antara aktivitas dengan ekstra intent.

## Yang akan Anda LAKUKAN

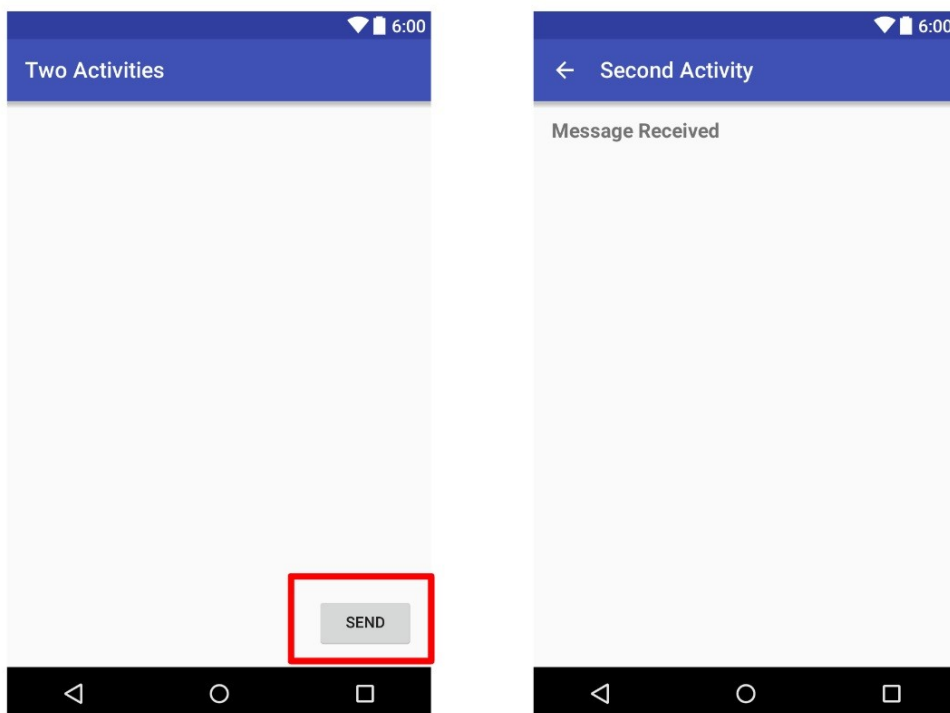
Dalam praktik ini Anda akan:

- Membuat aplikasi Android baru dengan dua aktivitas.
- Meneruskan beberapa data (string) dari aktivitas utama ke yang kedua menggunakan intent, dan menampilkan data tersebut dalam aktivitas kedua.
- Mengirimkan bit data kedua yang berbeda kembali ke aktivitas utama, menggunakan intent juga.

## Ringkasan Aplikasi

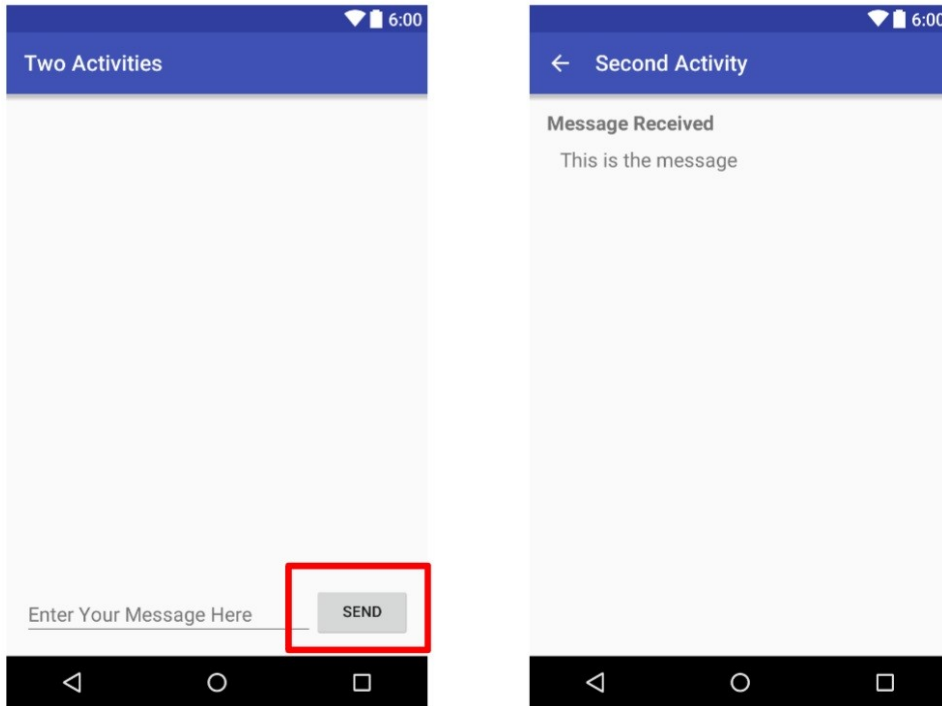
Dalam bab ini Anda akan membuat dan membangun aplikasi bernama TwoActivities yang, ternyata, berisi dua aktivitas. Aplikasi ini akan dibangun dalam tiga tahap.

Pada tahap pertama, buat aplikasi yang aktivitas utamanya hanya berisi satu tombol (Send). Saat pengguna mengeklik tombol ini, aktivitas utama menggunakan intent untuk memulai aktivitas kedua.



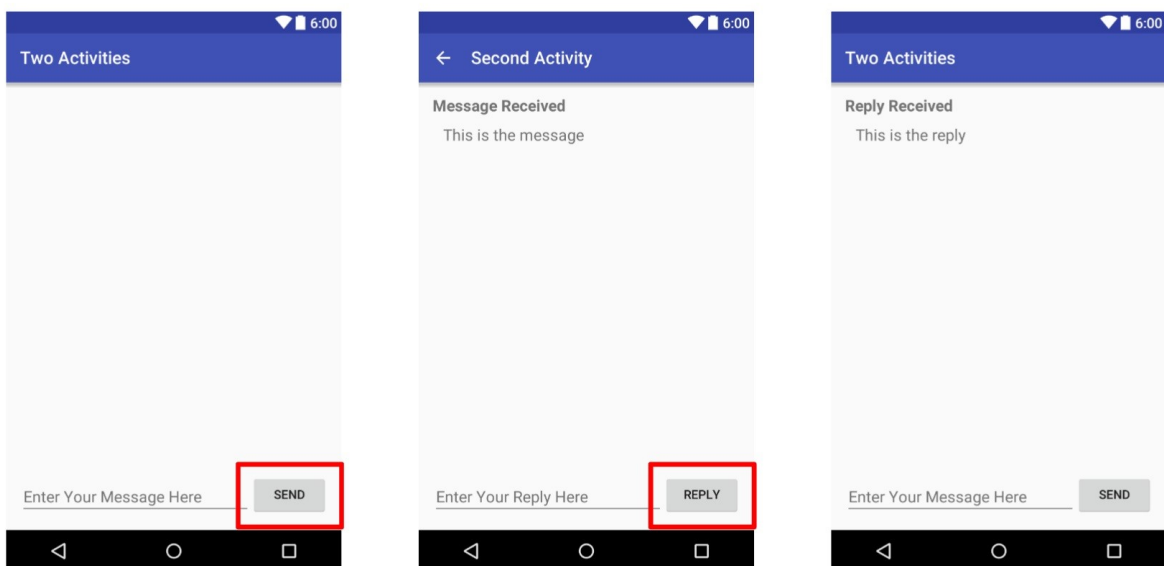
Main activity  Second activity

Dalam tahap kedua, Anda akan menambahkan tampilan EditText ke aktivitas utama. Pengguna memasukkan pesan dan mengeklik Send. Aktivitas utama menggunakan intent untuk memulai aktivitas kedua dan untuk mengirimkan pesan pengguna ke aktivitas tersebut. Aktivitas kedua menampilkan pesan yang diterimanya.



Main activity → Second activity

Pada tahap terakhir aplikasi TwoActivities, tambahkan tampilan EditText dan tombol Reply ke aktivitas kedua. Sekarang pengguna bisa mengetik pesan balasan dan mengklik Reply, dan balasannya ditampilkan di aktivitas utama. Pada titik ini, gunakan intent di sini untuk meneruskan pesan balasan kembali dari aktivitas kedua ke aktivitas utama.



Main activity → Second activity → Back to Main activity

# Tugas 1. Membuat proyek TwoActivities

Dalam tugas ini Anda akan menyiapkan proyek awal dengan aktivitas utama, mendefinisikan layout, dan mendefinisikan metode kerangka untuk peristiwa tombol `onClick`.

## 1.1 Membuat proyek TwoActivities

1. Mulai Android Studio dan buat proyek Android Studio baru.

Panggil aplikasi "TwoActivities" dan ubah domain perusahaan ke "android.example.com." Pilih SDK Minimum yang Anda gunakan di proyek sebelumnya.

2. Pilih **Empty Activity** untuk template proyek. Klik **Next**.
3. Terima nama aktivitas default (MainActivity). Pastikan kotak **Generate Layout file** dicentang. Klik **Finish**.

## 1.2 Definisikan layout untuk aktivitas utama

1. Buka `res/layout/activity_main.xml`. Pada Layout Editor, klik tab Design di bagian bawah layar.
2. Hapus TextView yang berbunyi "Hello World."
3. Tambahkan Tombol ke layout dalam posisi apa pun.
4. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut ini di dalam Tombol:

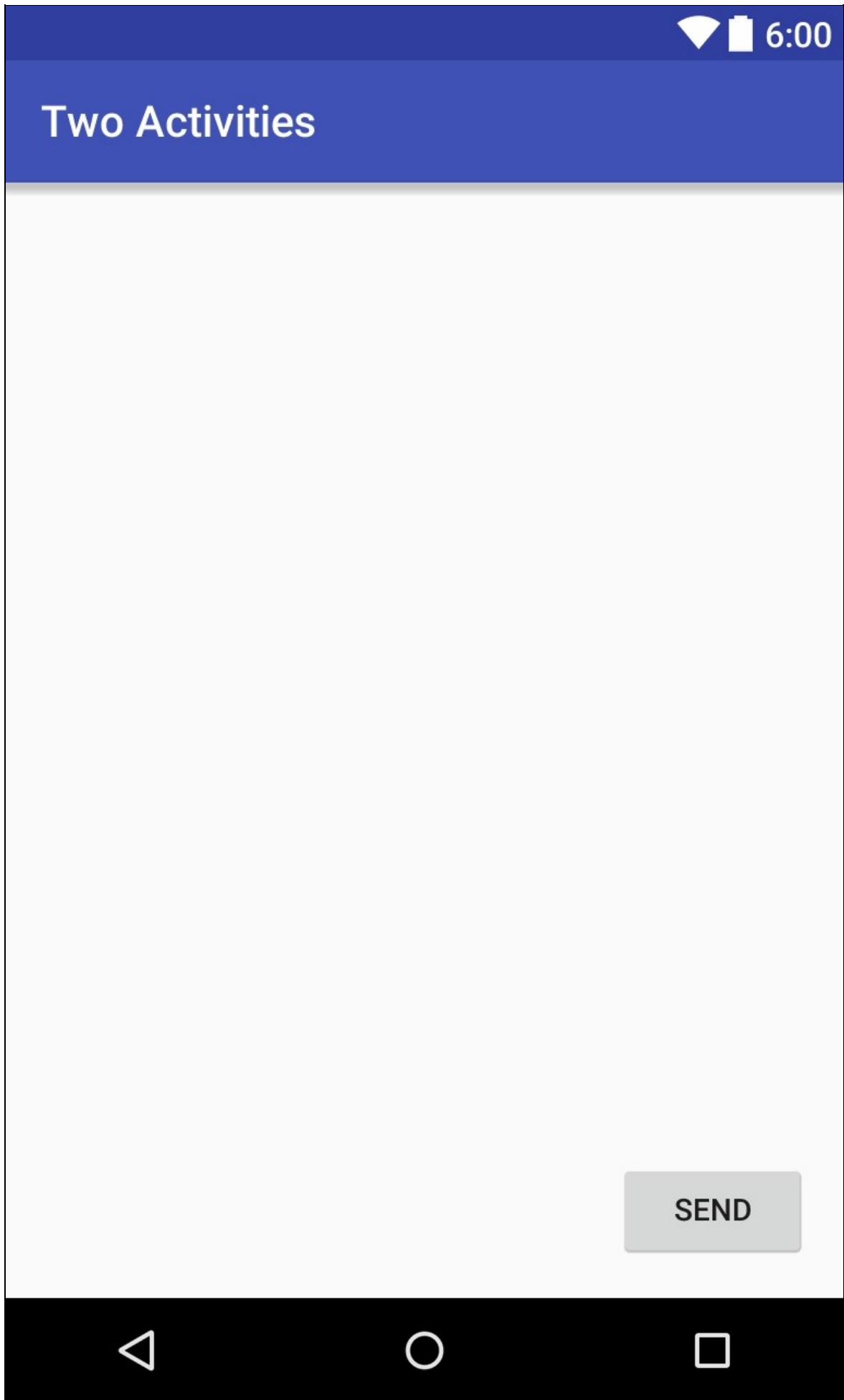
Atribut	Nilai
<code>android:id</code>	<code>"@+id/button_main"</code>
<code>android:layout_width</code>	<code>wrap_content</code>
<code>android:layout_height</code>	<code>wrap_content</code>
<code>android:layout_alignParentRight</code>	<code>"true"</code>
<code>android:layout_alignParentBottom</code>	<code>"true"</code>
<code>android:layout_alignParentEnd</code>	<code>"true"</code>
<code>android:text</code>	<code>"Send"</code>
<code>android:onClick</code>	<code>"launchSecondActivity"</code>

Ini mungkin membuat kesalahan "Method `launchSecondActivity` is missing in MainActivity." Abaikan kesalahan ini sekarang. Masalah ini akan diatasi pada tugas berikutnya.

5. Letakkan kursor pada kata `"Send"`.
6. Tekan **Alt-Enter** (**Option-Enter** di Mac), dan pilih **Extract string resources**.
7. Setel nama **Sumber Daya** ke `button_main` dan klik **OK**.

Proses ini membuat sumber daya string dalam file `values/res/string.xml`, dan string dalam kode Anda digantikan dengan referensi ke sumber daya string tersebut.

8. Pilih **Code > Reformat Code** untuk memformat kode XML, jika perlu.
9. Pratinjau layout aktivitas utama menggunakan Layout Editor. Layout harus terlihat seperti ini:



**Kode Solusi:**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>
</RelativeLayout>
```

**1.3 Mendefinisikan tindakan tombol**

Dalam tugas ini Anda akan mengimplementasikan metode `onClick` yang didefinisikan dalam layout.

1. Dalam Editor XML, letakkan kursor pada kata `"launchSecondActivity"` .
2. Tekan **Alt-Enter** (**Option-Enter** di Mac) dan pilih **Create 'launchSecondActivity(View)' dalam 'MainActivity'**.

File `MainActivity.java` terbuka, dan Android Studio membuat metode kerangka untuk handler `onClick`.

3. Dalam `launchSecondActivity` , tambahkan pernyataan log yang bernama `"Button Clicked!"`

```
Log.d(LOG_TAG, "Button clicked!");
```

`LOG_TAG` akan terlihat berwarna merah. Definisi untuk variabel tersebut akan ditambahkan dalam langkah selanjutnya.

4. Tempatkan kursor pada kata `"Log"` dan tekan **Alt-Enter** (**Option-Enter** di Mac). Android Studio menambahkan pernyataan impor untuk `android.util.Log`.
5. Di bagian atas kelas, tambahkan konstanta untuk variabel `LOG_TAG`:

```
private static final String LOG_TAG =
    MainActivity.class.getSimpleName();
```

Konstanta ini menggunakan nama kelas itu sendiri sebagai tag-nya.

6. Jalankan aplikasi Anda. Saat mengklik tombol `"Send"`, Anda akan melihat pesan `"Button Clicked!"` di Android Monitor (logcat). Jika ada terlalu banyak keluaran di monitor, ketik `MainActivity` di dalam kotak pencarian dan log hanya akan menunjukkan baris yang cocok dengan tag tersebut.

**Kode Solusi:**

```

package com.example.android.twoactivities;

import android.support.v7.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!");
    }
}

```

## Tugas 2. Membuat dan meluncurkan aktivitas kedua

Setiap aktivitas baru yang Anda tambahkan ke proyek memiliki layout dan file Java-nya sendiri, terpisah dari milik aktivitas utama. Aktivitas tersebut juga memiliki elemen `<activity>` sendiri dalam manifes Android. Seperti aktivitas utama, aktivitas baru yang Anda buat di Android Studio juga meluas dari kelas `AppCompatActivity`.

Semua aktivitas di aplikasi terhubung secara longgar dengan satu sama lain. Namun, Anda dapat mendefinisikan sebuah aktivitas sebagai induk dari akti lain dalam file `AndroidManifest.xml`. Hubungan induk-anak ini memungkinkan Android untuk menambahkan petunjuk navigasi seperti panah yang menghadap ke kiri pada bilah judul untuk setiap aktivitas.

Aktivitas berkomunikasi dengan satu sama lain (di dalam aplikasi yang sama dan di berbagai aplikasi yang berbeda) dengan *intent*. Ada dua tipe intent, eksplisit dan implisit. Intent eksplisit adalah intent yang targetnya Anda ketahui. Anda sudah mengetahui nama kelas yang sepenuhnya memenuhi syarat dari aktivitas spesifik tersebut. Intent implisit adalah intent yang nama komponen targetnya tidak Anda ketahui, namun memiliki tindakan umum untuk dikerjakan. Anda akan mempelajari tentang intent implisit pada praktik berikutnya.

Pada tugas ini Anda akan menambahkan aktivitas kedua pada aplikasi, dengan layout-nya sendiri. Anda akan memodifikasi manifes Android untuk mendefinisikan aktivitas utama sebaga induk aktivitas kedua. Lalu Anda akan memodifikasi metode kejadian `onClick` dalam aktivitas utama untuk menyertakan intent yang meluncurkan aktivitas kedua saat tombol diklik.

### 2.1 Membuat aktivitas kedua

1. Klik folder **app** untuk proyek Anda dan pilih **File > New > Activity > Empty Activity**.
2. Namakan aktivitas baru "SecondActivity." Pastikan **Generate Layout File** dicentang, dan nama layout akan diisi dengan `activity_second`.
3. Klik **Finish**. Android Studio menambahkan layout aktivitas baru (`activity_second`) dan file Java baru (`SecondActivity`) ke proyek Anda untuk aktivitas baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan aktivitas baru.

### 2.2 Memodifikasi manifes Android

1. Buka `manifests/AndroidManifest.xml`.
2. Temukan elemen `<activity>` yang dibuat Android Studio untuk aktivitas kedua.

```
<activity android:name=".SecondActivity"></activity>
```

3. Tambahkan atribut ini ke elemen `<activity>` :

Atribut	Nilai
android:label	"Second Activity"
android:parentActivityName	".MainActivity"

Atribut `label` menambahkan judul aktivitas ke bilah tindakan.

Atribut `parentActivityName` menandakan bahwa aktivitas utama adalah induk aktivitas kedua. Hubungan aktivitas induk ini digunakan untuk navigasi "ke atas" di dalam aplikasi Anda. Dengan mendefinisikan atribut ini, bilah tindakan untuk aktivitas kedua akan muncul dengan panah yang menghadap ke kiri untuk memungkinkan pengguna untuk bergerak "ke atas" ke aktivitas utama.

- Tempatkan kursor di "Second Activity" dan tekan **Alt-Enter** (**Option-Enter** di Mac).
- Pilih **Extract string resource**, namakan sumber daya "activity2\_name", dan klik **OK**. Android Studio menambahkan sumber daya string untuk label aktivitas.
- Tambahkan elemen `<meta-data>` element inside the `<activity>` di dalam elemen untuk aktivitas kedua. Gunakan atribut ini:

Atribut	Nilai
android:name	"android.support.PARENT_ACTIVITY"
android:value	"com.example.android.twoactivities.MainActivity"

Elemen `<meta-data>` menyediakan informasi arbitrer tambahan tentang aktivitas sebagai pasangan kunci-nilai. Dalam hal ini, atribut ini mencapai hal yang sama dengan atribut `android:parentActivityName`, yakni mendefinisikan hubungan antara dua aktivitas untuk tujuan navigasi ke atas. Atribut ini diperlukan untuk versi Android lebih lama. `android:parentActivityName` hanya tersedia untuk API level 16 dan yang lebih tinggi.

#### Kode Solusi:

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SecondActivity"
            android:label="@string/activity2_name"
            android:parentActivityName=".MainActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="com.example.android.twoactivities.MainActivity" />
            </activity>
        </application>
    </manifest>
```

## 2.3 Mendefinisikan layout untuk aktivitas utama

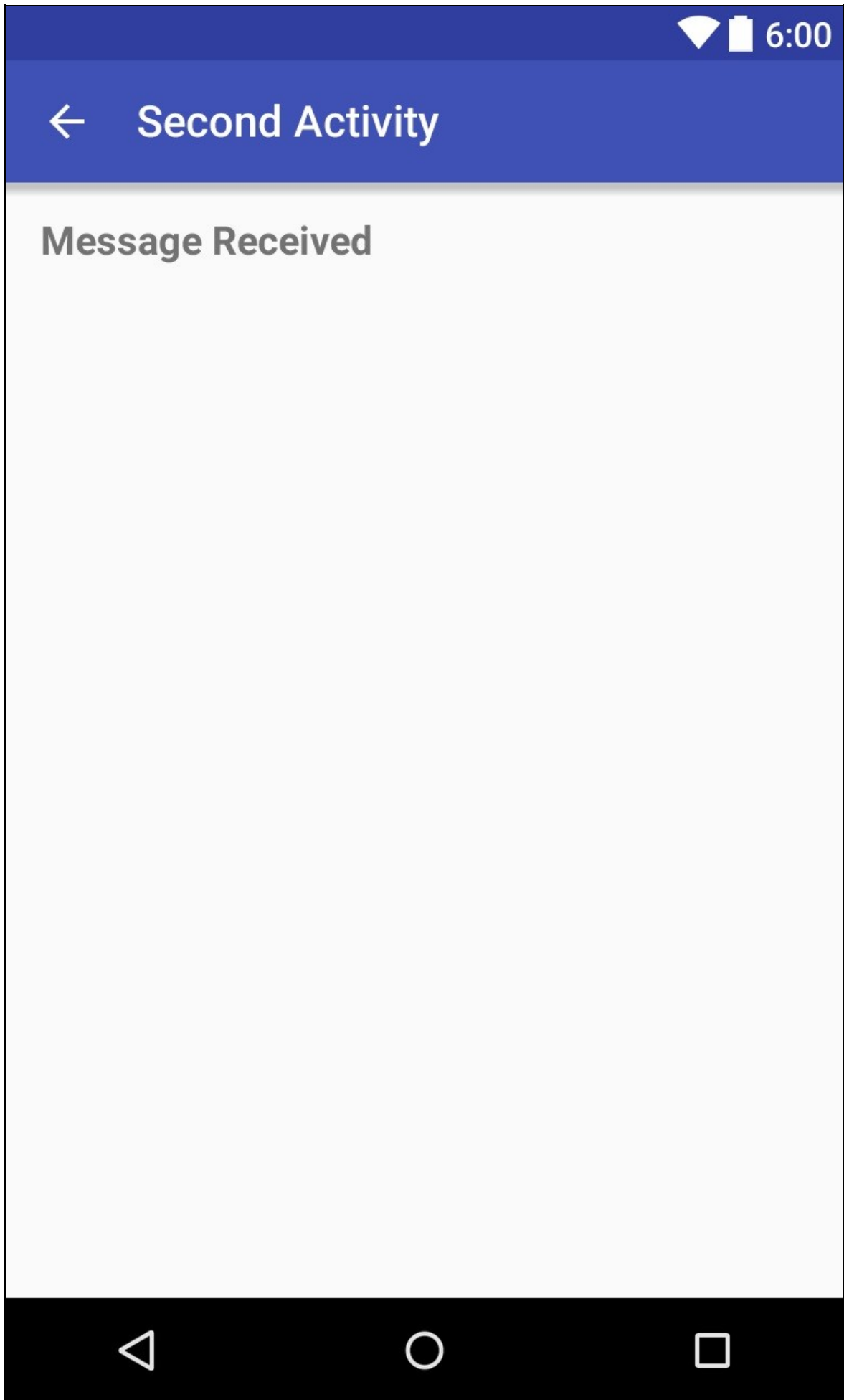
- Buka `res/layout/activity_second.xml`.
- Tambahkan `TextView` ("Plain Textview" dalam Layout Editor). Berikan atribut berikut pada `TextView`:



Atribut	Nilai
android:id	"@+id/text_header"
android:layout_width	wrap_content
android:layout_height	wrap_content
android:layout_marginBottom	"@dimen/activity_vertical_margin"
android:text	"Message Received"
android:textAppearance	"?android:attr/textAppearanceMedium"
android:textStyle	"bold"

Nilai `textAppearance` adalah atribut bertema Android yang mendefinisikan gaya font dasar font kecil, medium, dan besar. Anda akan mempelajari selengkapnya tentang tema pada pelajaran berikutnya.

3. Ekstrak string "Message Received" ke sumber daya bernama `text_header`.
4. Pratinjau layout dalam Layout Editor. Layout harus terlihat seperti ini:



**Kode Solusi:**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:text="@string/text_header"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold" />
</RelativeLayout>
```

## 2.4 Menambahkan intent ke aktivitas utama

Dalam aktivitas ini Anda akan menambahkan intent eksplisit ke aktivitas utama. Intent ini digunakan untuk mengaktifkan aktivitas kedua saat tombol Send diklik.

1. Buka file Java untuk `MainActivity` (`java/com.example.android.twoactivities/MainActivity`).
2. Buat intent baru dalam metode `launchSecondActivity()`.

Konstruktor intent memerlukan dua argumen untuk intent eksplisit: konteks [Aplikasi](#) dan komponen spesifik yang akan menerima intent tersebut. Di sini Anda harus menggunakan `this` sebagai konteksnya, dan `SecondActivity.class` sebagai kelas spesifiknya.

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

3. Tempatkan cursor di intent dan tekan **Alt-Enter** (**Option-Enter** di Mac) untuk menambahkan impor ke kelas intent.
4. Panggil metode `startActivity()` dengan intent baru sebagai argumennya.

```
startActivity(intent);
```

5. Jalankan aplikasi.

Saat mengklik tombol Send, aktivitas utama mengirim intent dan sistem Android membuka aktivitas kedua. Aktivitas kedua muncul di layar. Untuk kembali ke aktivitas utama, klik tombol Kembali Android di kiri bawah layar, Anda bisa menggunakan panah kiri di atas aktivitas kedua untuk kembali ke aktivitas utama.

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode bersifat opsional.

**Tantangan:** Apa yang terjadi jika Anda menghapus `android:parentActivityName` dan elemen `<meta-data>` dari manifest? Buat perubahan ini dan jalankan aplikasi.

## Tugas 3. Mengirim data dari aktivitas utama ke aktivitas kedua

Dalam tugas terakhir, Anda akan menambahkan intent eksplisit ke aktivitas utama yang mengaktifkan aktivitas kedua. Anda juga bisa menggunakan intent untuk mengambil data dari aktivitas satu ke aktivitas lainnya.

Dalam tugas ini, Anda akan memodifikasi intent eksplisit dalam aktivitas utama untuk menyertakan data tambahan (dalam kasus ini, string yang dimasukkan pengguna) dalam ekstra intent. Lalu Anda akan memodifikasi aktivitas kedua untuk mendapatkan data kembali dari ekstra intent dan menampilkannya di layar.

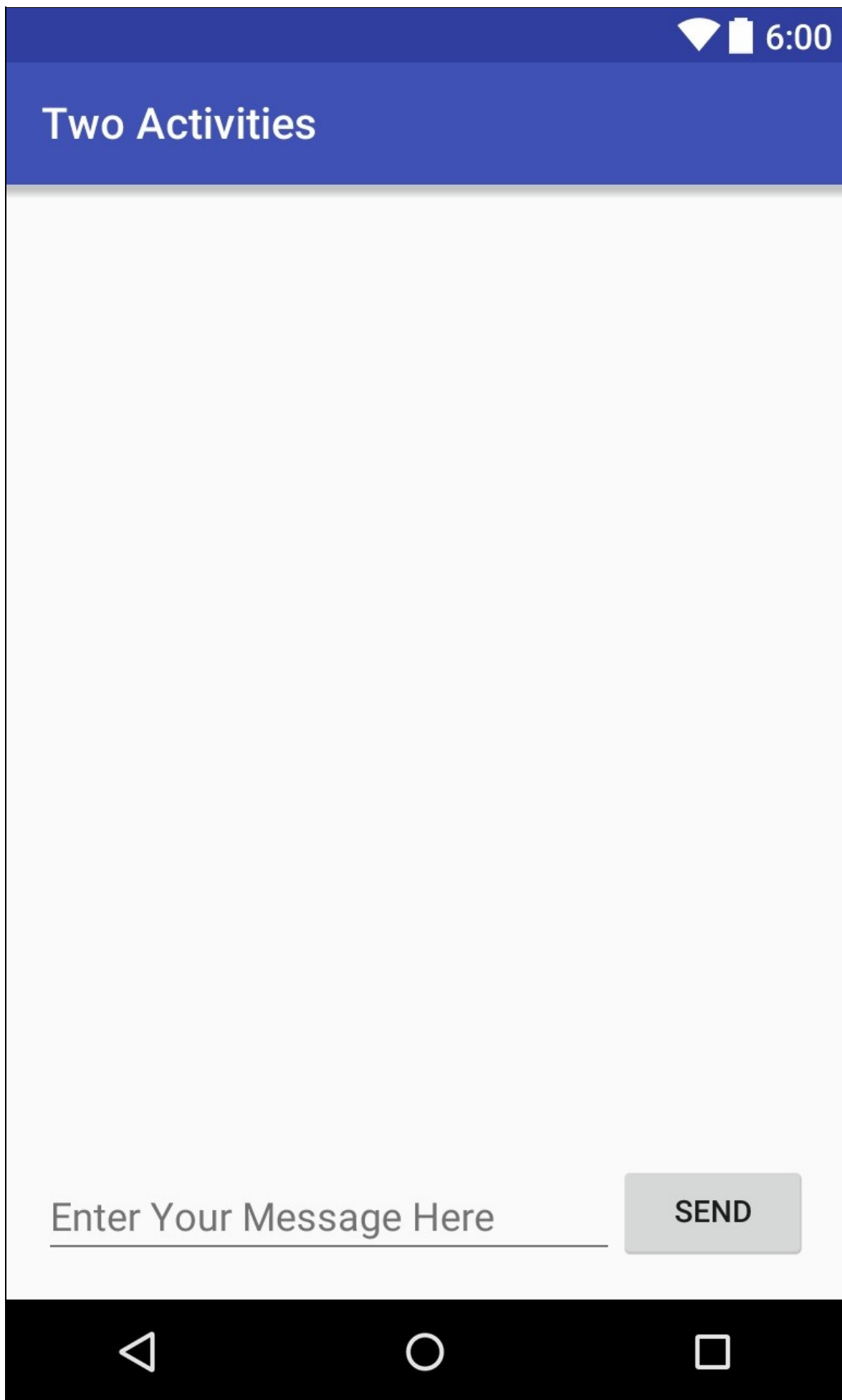
### 3.1 Menambahkan EditText ke layout aktivitas utama

1. Buka `res/layout/activity_main.xml`.
2. Tambahkan tampilan EditText (Plain Text dalam Layout Editor.) Berikan EditText atribut berikut:

Atribut	Nilai
<code>android:id</code>	<code>"@+id/editText_main"</code>
<code>android:layout_width</code>	<code>match_parent</code>
<code>android:layout_height</code>	<code>wrap_content</code>
<code>android:layout_toLeftOf</code>	<code>"@+id/button_main"</code>
<code>android:layout_toStartOf</code>	<code>"@+id/button_main"</code>
<code>android:layout_alignParentBottom</code>	<code>"true"</code>
<code>android:hint</code>	<code>"Enter Your Message Here"</code>

3. Hapus `android:text` attribute.
4. Ekstrak string "Enter Your Message Here" ke sumber daya bernama `editText_main`.

Layout baru untuk aktivitas utama harus terlihat seperti ini:



**Kode Solusi:**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_main"
        android:layout_toStartOf="@+id/button_main"
        android:hint="@string/editText_main" />
</RelativeLayout>
```

**3.2 Menambahkan string ke ekstra intent aktivitas utama**

Objek intent dapat meneruskan data ke aktivitas target dalam dua cara: dalam bidang data, atau dalam *ekstra* intent. Data intent adalah URI yang menandakan data spesifik untuk dikerjakan. Jika informasi yang ingin diteruskan ke aktivitas melalui intent bukan URI, atau jika Anda memiliki lebih dari satu informasi yang ingin dikirimkan, Anda bisa meletakkan informasi tambahan tersebut ke dalam *ekstra* intent.

*Ekstra* intent adalah pasangan kunci/nilai dalam **Bundel**. *Bundel* adalah kumpulan data, yang disimpan sebagai pasangan kunci/nilai. Untuk meneruskan informasi dari satu aktivitas ke yang lainnya, letakkan kunci dan nilai ke dalam bundel ekstra intent dari aktivitas mengirim, dan keluarkan lagi ke aktivitas menerima.

1. Buka `java/com.example.android.twoactivities/MainActivity`.
2. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk ekstra intent:

```
public static final String EXTRA_MESSAGE =
    "com.example.android.twoactivities.extra.MESSAGE";
```

3. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText. Mengimpor kelas EditText.

```
private EditText mMessageEditText;
```

4. Dalam metode `onCreate()`, gunakan `findViewById` untuk mendapatkan referensi tentang contoh EditText dan menetapkannya ke variabel privat tersebut:

```
mMessageEditText = (EditText) findViewById(R.id.editText_main);
```

5. Dalam metode `launchSecondActivity()`, di bawah intent baru, dapatkan teks dari EditText sebagai string:

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

- Tambahkan string tersebut ke intent sebagai sebuah ekstra dengan konstanta EXTRA\_MESSAGE sebagai kunci dan string-nya sebagai nilai:

```
intent.putExtra(EXTRA_MESSAGE, message);
```

#### Kode Solusi:

```
package com.example.android.twoactivities;

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.EditText;

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();

    public static final String EXTRA_MESSAGE =
"com.example.android.twoactivities.extra.MESSAGE";

    private EditText mMessageEditText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mMessageEditText = (EditText) findViewById(R.id.editText_main);
    }

    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);
    }
}
```

### 3.3 Menambahkan TextView ke aktivitas kedua untuk pesan

- Buka `res/layout/activity_second.xml`.
- Tambahkan TextView kedua. Berikan atribut berikut pada TextView:

Atribut	Nilai
android:id	"@+id/text_message"
android:layout_width	wrap_content
android:layout_height	wrap_content
android:layout_below	"@+id/text_header"
android:layout_marginLeft	"@dimen/activity_horizontal_margin"
android:layout_marginStart	"@dimen/activity_horizontal_margin"
android:textSize	"?android:attr/textAppearanceMedium"

- Hapus atribut `android:text` (jika ada).

Layout baru untuk aktivitas kedua terlihat sama dengan layout pada tugas sebelumnya karena TextView baru tidak berisi (belum) teks apa pun, sehingga tidak muncul di layar.

#### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.SecondActivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />

</RelativeLayout>
```

### 3.4 Memodifikasi aktivitas kedua untuk mendapatkan ekstra dan menampilkan pesan

1. Buka `java/com.example.android.twoactivities/SecondActivity`.
2. Dalam metode `onCreate()`, dapatkan intent yang mengaktifkan aktivitas ini:

```
Intent intent = getIntent();
```

3. Dapatkan string berisi pesan dari ekstra intent menggunakan variabel statis `MainActivity.EXTRA_MESSAGE` sebagai kuncinya:

```
String message =
    intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
```

4. Gunakan `findViewById` untuk mendapatkan referensi ke `TextView` untuk pesan dari layout (Anda mungkin perlu mengimpor kelas `TextView`):

```
TextView textView = (TextView) findViewById(R.id.text_message);
```

5. Setel teks `TextView` tersebut ke string dari ekstra intent:

```
textView.setText(message);
```

6. Jalankan aplikasi. Saat Anda mengetik pesan dalam aktivitas utama dan mengklik Send, aktivitas kedua terbuka dan menampilkan pesan tersebut.

#### Kode Solusi:



```

package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @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 = (TextView) findViewById(R.id.text_message);
        textView.setText(message);
    }
}

```

## Tugas 4. Mengembalikan data ke aktivitas utama

Sekarang karena Anda telah memiliki aplikasi yang meluncurkan aktivitas baru dan mengirimkan data ke aplikasi, langkah terakhir adalah mengembalikan data ke aktivitas utama. Anda juga akan menggunakan intent dan *ekstra* intent untuk tugas ini.

### 4.1 Menambahkan EditText dan Tombol ke layout aktivitas kedua

1. Salin EditText dan Tombol dari file layout aktivitas utama dan tempel ke layout kedua.
2. Dalam file `activity_second.xml`, modifikasi nilai atribut untuk Tombol dan tampilan EditText. Gunakan nilai ini:

Atribut Lama (Tombol)	Atribut Baru (Tombol)
<code>android:id="@+id/button_main"</code>	<code>android:id="@+id/button_second"</code>
<code>android:onClick="launchSecondActivity"</code>	<code>android:onClick="returnReply"</code>
<code>android:text="@string/button_main"</code>	<code>android:text="@string/button_second"</code>
Atribut Lama (EditText)	Atribut Baru (EditText)
<code>android:id="@+id/editText_main"</code>	<code>android:id="@+id/editText_second"</code>
<code>android:layout_toLeftOf="@+id/button_main"</code>	<code>android:layout_toLeftOf="@+id/button_second"</code>
<code>android:layout_toStartOf="@+id/button_main"</code>	<code>android:layout_toStartOf="@+id/button_second"</code>
<code>android:hint="@string/editText_main"</code>	<code>android:hint="@string/editText_second"</code>

3. Buka `res/values/strings.xml` dan tambahkan sumber daya String untuk teks tombol dan petunjuknya di EditText:

```

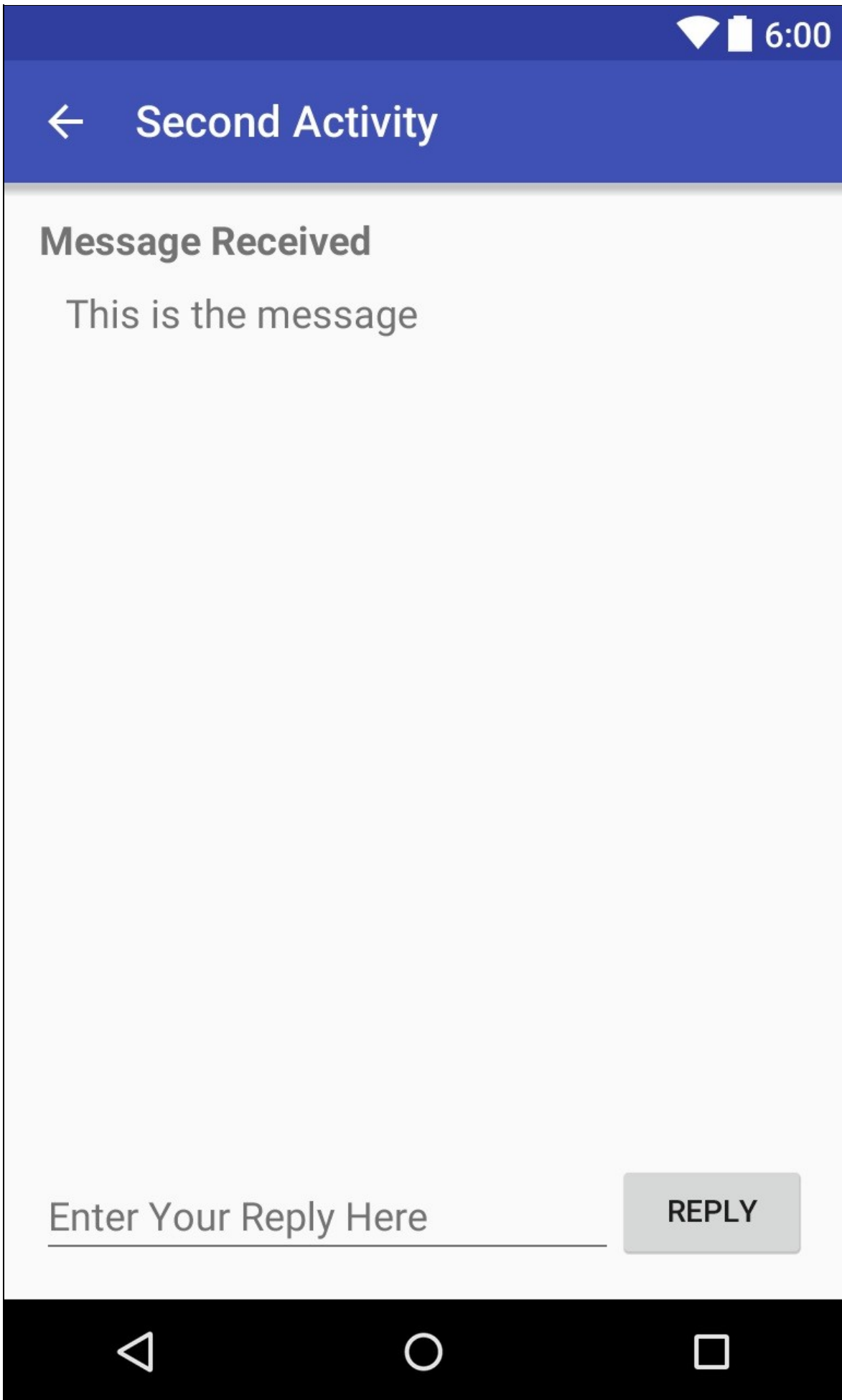
<string name="button_second">Reply</string>
<string name="editText_second">Enter Your Reply Here</string>

```

4. Dalam editor layout XML, letakkan kursor pada `"returnReply"`, tekan **Alt-Enter** (**Option-Enter** di Mac) dan pilih **Create 'launchSecondActivity(View)' dalam 'SecondActivity'**.

File `SecondActivity.java` terbuka, dan Android Studio membuat metode kerangka untuk handler `onClick`. Anda akan mengimplementasikan metode ini dalam tugas berikutnya.

Layout baru untuk aktivitas kedua terlihat seperti ini:



**Kode Solusi:**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.SecondActivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_second"
        android:id="@+id/button_second"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="returnReply"/>

    <EditText
        android:id="@+id/editText_second"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_second"
        android:layout_toStartOf="@+id/button_second"
        android:hint="@string/editText_second" />

</RelativeLayout>

```

**4.2 Membuat intent respons dalam aktivitas kedua**

1. Buka `java/com.example.android.twoactivities/SecondActivity`.
2. Di atas kelas, tambahkan konstanta publik untuk mendefinisikan kunci untuk ekstra intent:

```

public static final String EXTRA_REPLY =
    "com.example.android.twoactivities.extra.REPLY";

```

3. Tambahkan variabel privat di bagian atas kelas untuk menampung objek `EditText`.

```

private EditText mReply;

```

4. Dalam metode `onCreate()`, gunakan `findViewById()` untuk mendapatkan referensi tentang contoh `EditText` dan menetapkannya ke variabel privat tersebut:

```
mReply = (EditText) findViewById(R.id.editText_second);
```

5. Dalam metode `returnReply()`, dapatkan teks `EditText` sebagai string:

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

6. Membuat intent baru untuk respons tersebut.

**Catatan:** Jangan gunakan ulang objek intent yang diterima dari permintaan asalnya. Buat intent baru untuk respons tersebut.

```
Intent replyIntent = new Intent();
```

7. Tambahkan string balasan dari `EditText` ke intent baru sebagai *ekstra* intent. Karena *ekstra* adalah pasangan kunci/nilai, di sini kuncinya adalah `EXTRA_REPLY` dan nilainya adalah balasan:

```
replyIntent.putExtra(EXTRA_REPLY, reply);
```

8. Setel hasilnya ke `RESULT_OK` untuk menunjukkan bahwa responsnya berhasil. Kode hasil (termasuk `RESULT_OK` dan `RESULT_CANCELLED`) didefinisikan oleh kelas [Aktivitas](#).

```
setResult(RESULT_OK, replyIntent);
```

9. Panggil `finish()` untuk menutup aktivitas dan kembali ke aktivitas utama.

```
finish();
```

#### Kode Solusi:

```

package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {
    public static final String EXTRA_REPLY =
        "com.example.android.twoactivities.extra.REPLY";

    private EditText mReply;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        mReply = (EditText) findViewById(R.id.editText_second);

        Intent intent = getIntent();
        String message =
            intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = (TextView) findViewById(R.id.text_message);
        textView.setText(message);
    }

    public void returnReply(View view) {
        String reply = mReply.getText().toString();

        Intent replyIntent = new Intent();
        replyIntent.putExtra(EXTRA_REPLY, reply);
        setResult(RESULT_OK, replyIntent);
        finish();
    }
}

```

### 4.3 Menambahkan TextViews ke layout aktivitas utama untuk menampilkan balasan

Aktivitas utama memerlukan cara untuk menampilkan balasan yang dikirim kembali ke aktivitas kedua. Dalam tugas ini Anda akan menambahkan TextViews ke layout aktivitas utama untuk menampilkan balasan tersebut. Untuk memudahkannya, Anda dapat menyalin TextViews yang digunakan dalam aktivitas kedua.

1. Salin dua TextViews untuk tampilan pesan dari file layout aktivitas kedua dan tempel ke layout utama di atas EditText yang ada dan tampilan Tombol.
2. Modifikasi nilai atribut untuk kedua TextViews baru ini. Gunakan nilai ini:

Atribut Lama (header TextView)	Atribut Baru (header TextView)
android:id="@+id/text_header"	android:id="@+id/text_header_reply"
android:text="@string/text_header"	android:text="@string/text_header_reply"
Atribut Lama (TextView pesan)	Atribut Baru (TextView pesan)
android:id="@+id/text_message"	android:id="@+id/text_message_reply"
android:layout_below="@+id/text_header"	android:layout_below="@+id/text_header_reply"

3. Tambahkan atribut `android:visibility` ke setiap TextView agar tidak terlihat di awal. (Membuatnya terlihat di layar tanpa konten apa pun dapat membuat pengguna bingung.) Anda akan menjadikan TextView terlihat setelah data respons diteruskan kembali dari aktivitas kedua.

```
android:visibility="invisible"
```

4. Buka `res/values/strings.xml` dan tambahkan sumber daya string untuk header balasan:

```
<string name="text_header_reply">Reply Received</string>
```

Layout untuk aktivitas utama terlihat sama dengan tugas sebelumnya, walaupun Anda sudah menambahkan dua `TextView` baru ke layout. Namun, karena Anda menyetel `TextView` menjadi tidak terlihat, `TextView` tidak muncul di layar.

#### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.MainActivity">
    <TextView
        android:id="@+id/text_header_reply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header_reply"
        android:visibility="invisible"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message_reply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header_reply"
        android:visibility="invisible"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_main"
        android:layout_toStartOf="@+id/button_main"
        android:hint="@string/editText_main" />
</RelativeLayout>
```

## 4.4 Mendapatkan balasan dari ekstra intent dan menampilkannya

Saat Anda menggunakan intent eksplisit untuk memulai aktivitas, Anda mungkin tidak berharap untuk mendapatkan data kembali, Anda hanya mengaktifkan aktivitas tersebut. Dalam kasus seperti ini, gunakan `StartActivity()` untuk memulai aktivitas baru, seperti yang Anda lakukan sebelumnya pada pelajaran ini. Tetapi jika ingin mendapatkan data kembali dari

aktivitas yang diaktifkan, Anda perlu memulainya dengan `startActivityForResult()`.

Dalam tugas ini Anda akan memodifikasi aplikasi untuk memulai aktivitas kedua dan mengharapkan hasil, untuk mengekstrak data pengembalian itu dari intent, dan untuk menampilkan data tersebut dalam `TextView` yang dibuat dalam tugas terakhir.

1. Buka `java/com.example.android.twoactivities/MainActivity`.
2. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk tipe respons tertentu yang Anda inginkan:

```
public static final int TEXT_REQUEST = 1;
```

3. Tambahkan dua variabel privat untuk menampung header balasan dan `TextView` balasan:

```
private TextView mReplyHeadTextView;
private TextView mReplyTextView;
```

4. dalam metode `onCreate()`, gunakan `findViewById` untuk mendapatkan referensi dari layout ke header balasan dan `TextView` balasan. Tetapkan dua contoh tampilan ke variabel privat:

```
mReplyHeadTextView = (TextView) findViewById(R.id.text_header_reply);
mReplyTextView = (TextView) findViewById(R.id.text_message_reply);
```

5. Dalam metode `launchSecondActivity()`, modifikasi panggilan ke `startActivity()` menjadi `startActivityForResult()`, dan sertakan kunci `TEXT_REQUEST` sebagai argumen:

```
startActivityForResult(intent, TEXT_REQUEST);
```

6. Buat metode callback `onActivityResult()` dengan tanda tangan ini:

```
public void onActivityResult(int requestCode, int resultCode,
    Intent data) {}
```

7. Dalam `onActivityResult()`, panggil `super.onActivityResult()`:

```
super.onActivityResult(requestCode, resultCode, data);
```

8. Tambahkan kode untuk `TEXT_REQUEST` (untuk memproses hasil intent yang benar, jika ada beberapa) dan `RESULT_CODE` (untuk memastikan permintaannya sukses):

```
if (requestCode == TEXT_REQUEST) {
    if (resultCode == RESULT_OK) {
    }
}
}
```

9. Di bagian dalam blok if, dapatkan ekstra intent dari intent respons (`data`). Di sini, kunci untuk ekstra adalah konstanta `EXTRA_REPLY` dari `SecondActivity`:

```
String reply = data.getStringExtra(SecondActivity.EXTRA_REPLY);
```

10. Setel; visibilitas header balasan ke benar:

```
mReplyHeadTextView.setVisibility(View.VISIBLE);
```

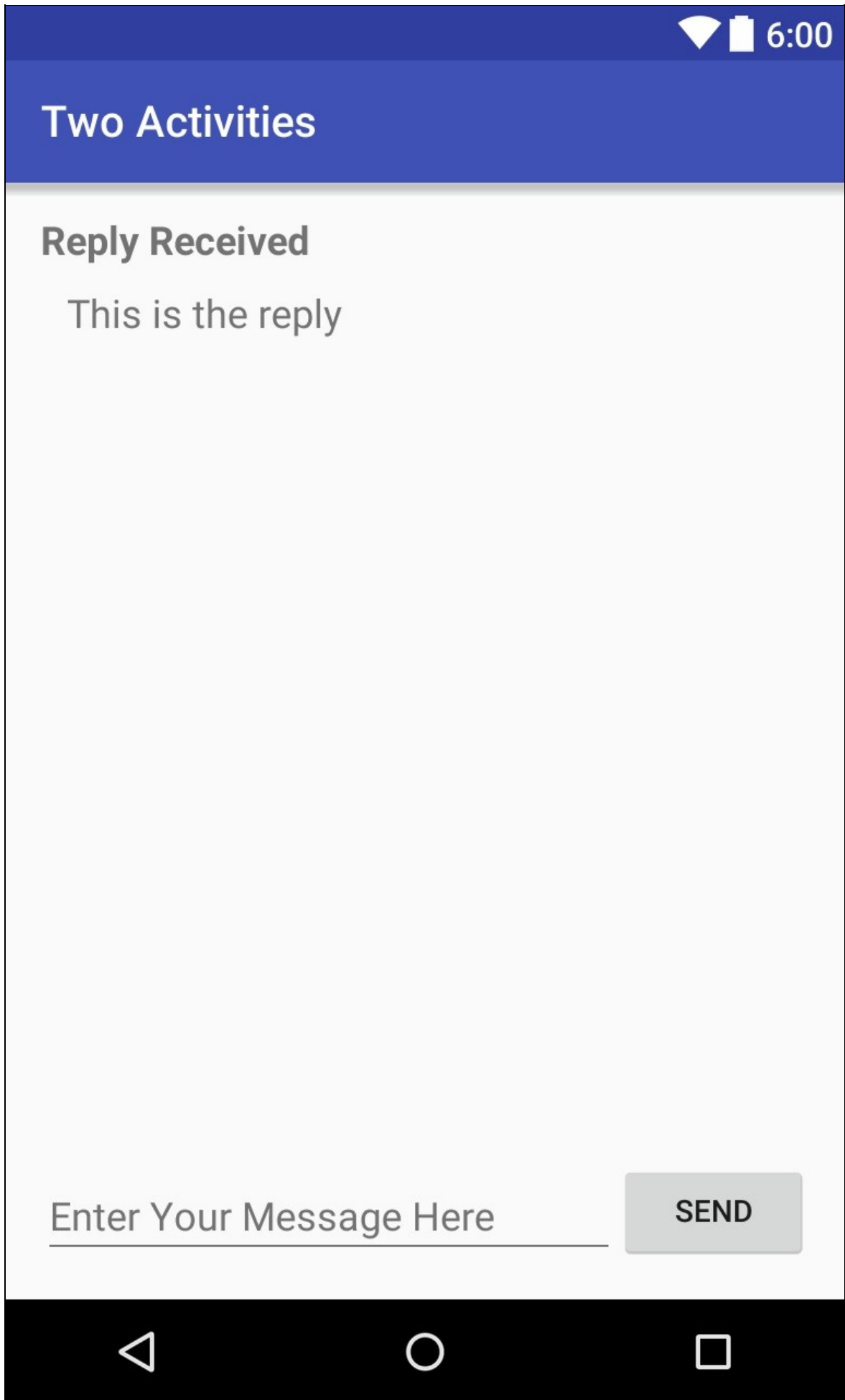
11. Setel teks `textview` balasan ke balasan, dan setel visibilitasnya ke benar:

```
mReplyTextView.setText(reply);
mReplyTextView.setVisibility(View.VISIBLE);
```

12. Jalankan aplikasi.

Sekarang, saat Anda mengirimkan pesan ke aktivitas kedua dan mendapatkan balasan kembali, aktivitas utama diperbarui untuk menampilkan balasan.





**Kode Solusi:**

```
package com.example.android.twoactivities;

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.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();

    public static final String EXTRA_MESSAGE =
        "com.example.android.twoactivities.extra.MESSAGE";

    public static final int TEXT_REQUEST = 1;

    private EditText mMessageEditText;
    private TextView mReplyHeadTextView;
    private TextView mReplyTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mMessageEditText = (EditText) findViewById(R.id.editText_main);
        mReplyHeadTextView = (TextView) findViewById(R.id.text_header_reply);
        mReplyTextView = (TextView) findViewById(R.id.text_message_reply);
    }

    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);
        startActivityForResult(intent, TEXT_REQUEST);
    }

    public void onActivityResult(int requestCode, int resultCode,
        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);
            }
        }
    }
}
```

**Kode Solusi**Proyek Android Studio: [TwoActivities](#)**Tantangan penyusunan kode**

**Catatan:**Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

**Tantangan:** Buat aplikasi dengan tiga tombol yang diberi label: Text One, Text Two, dan Text Three. Saat tombol mana saja diklik, aktivitas kedua terbuka. Aktivitas kedua itu harus berisi ScrollView yang menampilkan salah satu dari tiga bagian teks (Anda bisa menyertakan pilihan bagian). Gunakan intent untuk membuka aktivitas kedua dan ekstra intent untuk menunjukkan mana dari tiga bagian tersebut yang akan ditampilkan.

## Rangkuman

Dalam praktik ini Anda telah belajar:

- Sebuah Aktivitas adalah komponen aplikasi yang menyediakan layar tunggal yang fokus pada tugas seorang pengguna.
  - Setiap aktivitas memiliki file layout antarmuka pengguna masing-masing.
  - Anda bisa menetapkan hubungan induk/anak pada aktivitas untuk mengaktifkan navigasi "ke atas" di dalam aplikasi.
  - Untuk mengimplementasikan aktivitas di dalam aplikasi, lakukan hal berikut:
- Buat kelas Java aktivitas.
- Implementasikan antarmuka pengguna untuk aktivitas itu.
- Deklarasikan aktivitas baru itu dalam manifes aplikasi.
- Saat membuat proyek baru untuk aplikasi, atau menambahkan aktivitas baru ke aplikasi, dalam Android Studio (dengan File > new > Activity), kode template untuk setiap tugas ini disediakan untuk Anda.
- Intent mengizinkan Anda meminta tindakan dari komponen lain dalam aplikasi, misalnya, untuk memulai satu aktivitas dari lainnya. Intent dapat menjadi eksplisit atau implisit.
  - Dengan intent eksplisit Anda mengindikasikan komponen target spesifik untuk menerima data.
  - Dengan intent implisit Anda menetapkan fungsionalitas yang Anda inginkan tetapi tidak dengan komponen target.
  - Intent dapat menyertakan data tempat melakukan tindakan (sebagai URI) atau informasi tambahan sebagai *ekstra* intent.
  - *Ekstra* intent adalah pasangan kunci/nilai dalam bundel yang dikirimkan bersama intent-nya.
- Tampilan dapat dijadikan terlihat atau tidak terlihat dengan atribut `android:visibility`

## [Konsep terkait]

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Aktivitas dan Intent](#)

## Ketahui selengkapnya

- [Dasar-Dasar Aplikasi Android](#)
- [Memulai Aktivitas Lain](#)
- [Aktivitas](#) (Panduan API)
- [Aktivitas](#) (Referensi API)
- [Intent Filter Intent API Guide](#) (Panduan API)
- [Intent](#) (Referensi API)