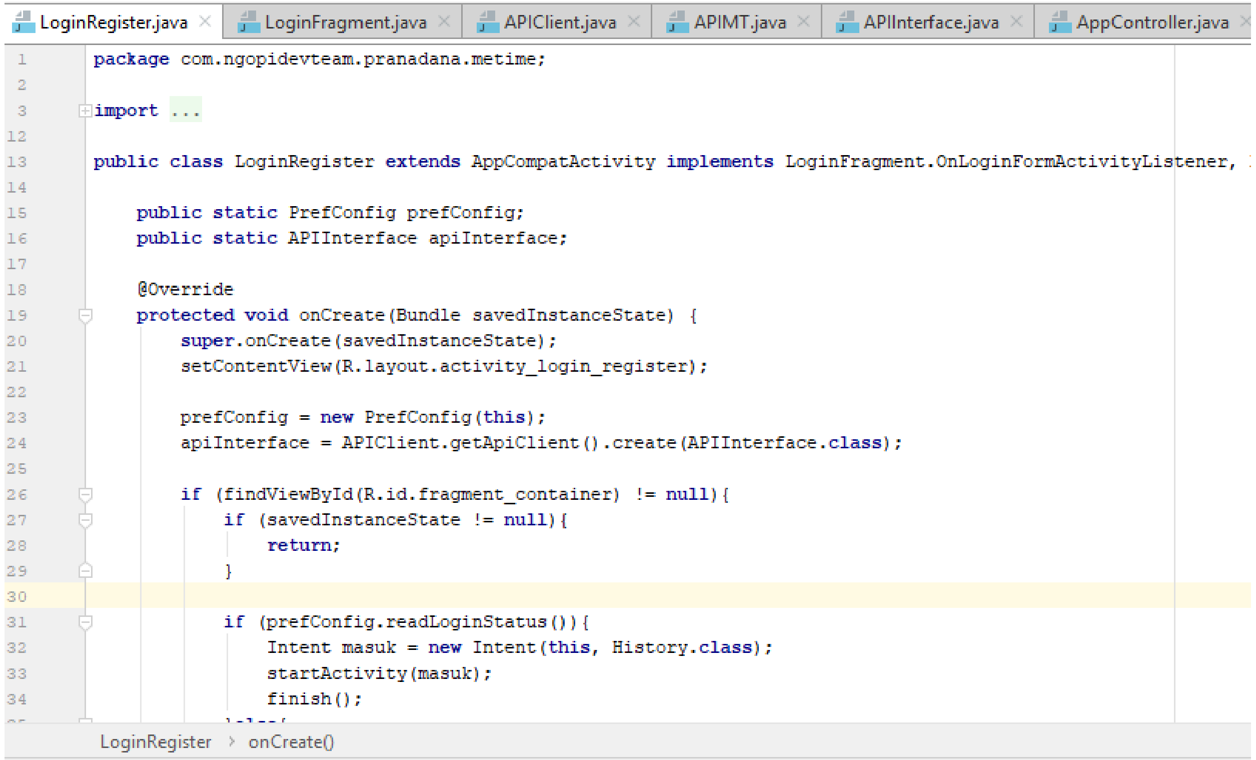
Android Programming

Android Activity

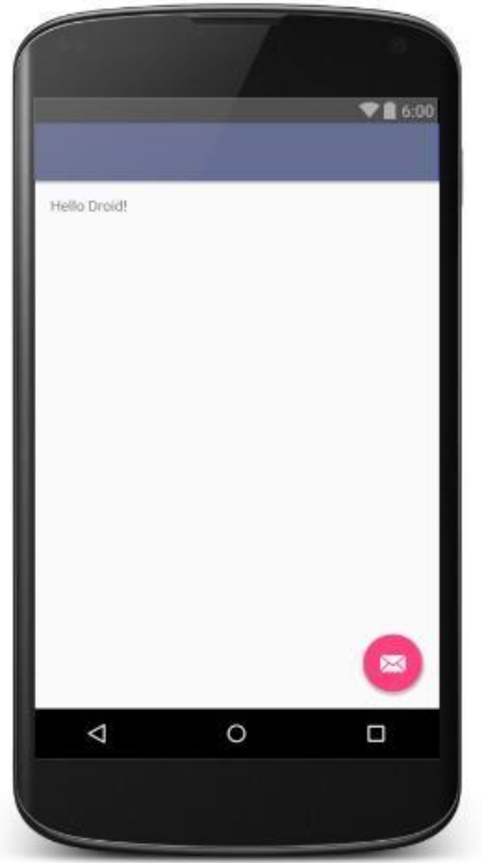
Activity

- ❑ Sebuah Activity adalah application component yang menyediakan sebuah screen (layar UI) sehingga user bisa berinteraksi



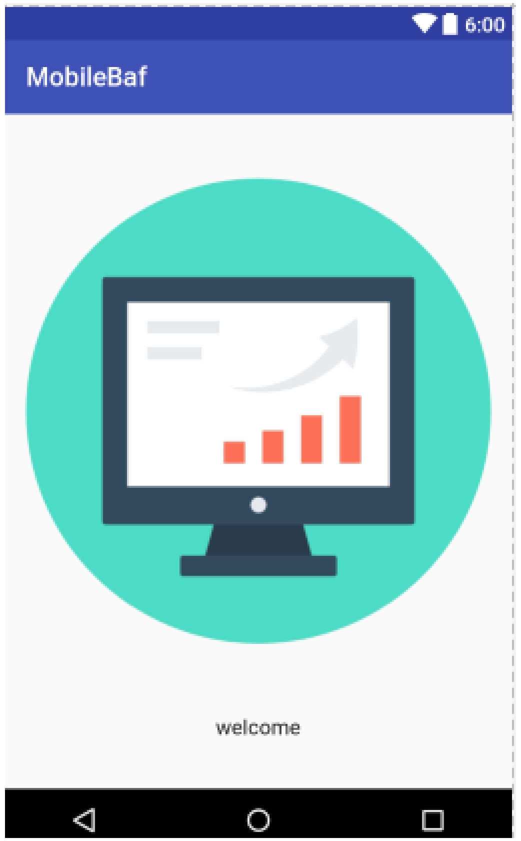
```java
package com.ngopidevteam.pranadana.metime;

import ...

public class LoginRegister extends AppCompatActivity implements LoginFragment.OnLoginFormActivityListener, 

    public static PrefConfig prefConfig;
    public static APIInterface apiInterface;

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

        prefConfig = new PrefConfig(this);
        apiInterface = APIClient.getApiClient().create(APIInterface.class);

        if (findViewById(R.id.fragment_container) != null){
            if (savedInstanceState != null){
                return;
            }

            if (prefConfig.readLoginStatus()){
                Intent masuk = new Intent(this, History.class);
                startActivity(masuk);
                finish();
```

Activity (.java)



Layout (.xml)

Layout : activity_main



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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:id="@+id/activity_main">

    <ImageView
        app:srcCompat="@drawable/logo"
        android:id="@+id/imgLogo"
        android:layout_above="@+id/txtWelcome"/>
    <TextView
        android:text="welcome"
        android:id="@+id/txtWelcome"
        android:textAppearance="@style/TextAppearance.AppCompat"
        android:textSize="16pt" />
</RelativeLayout>
```

Design Layout : GUI - click drag - rename ID

Text Layout : xml

Binding Object : XML - Java

```java
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    public ImageView img1;
    public TextView txt1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // binding object from xml layout
        img1 = (ImageView) findViewById(R.id.imglogo);
        txt1 = (TextView) findViewById(R.id.txtWelcome);

        txt1.setText("Selamat Datang !!!");
    }
}
```

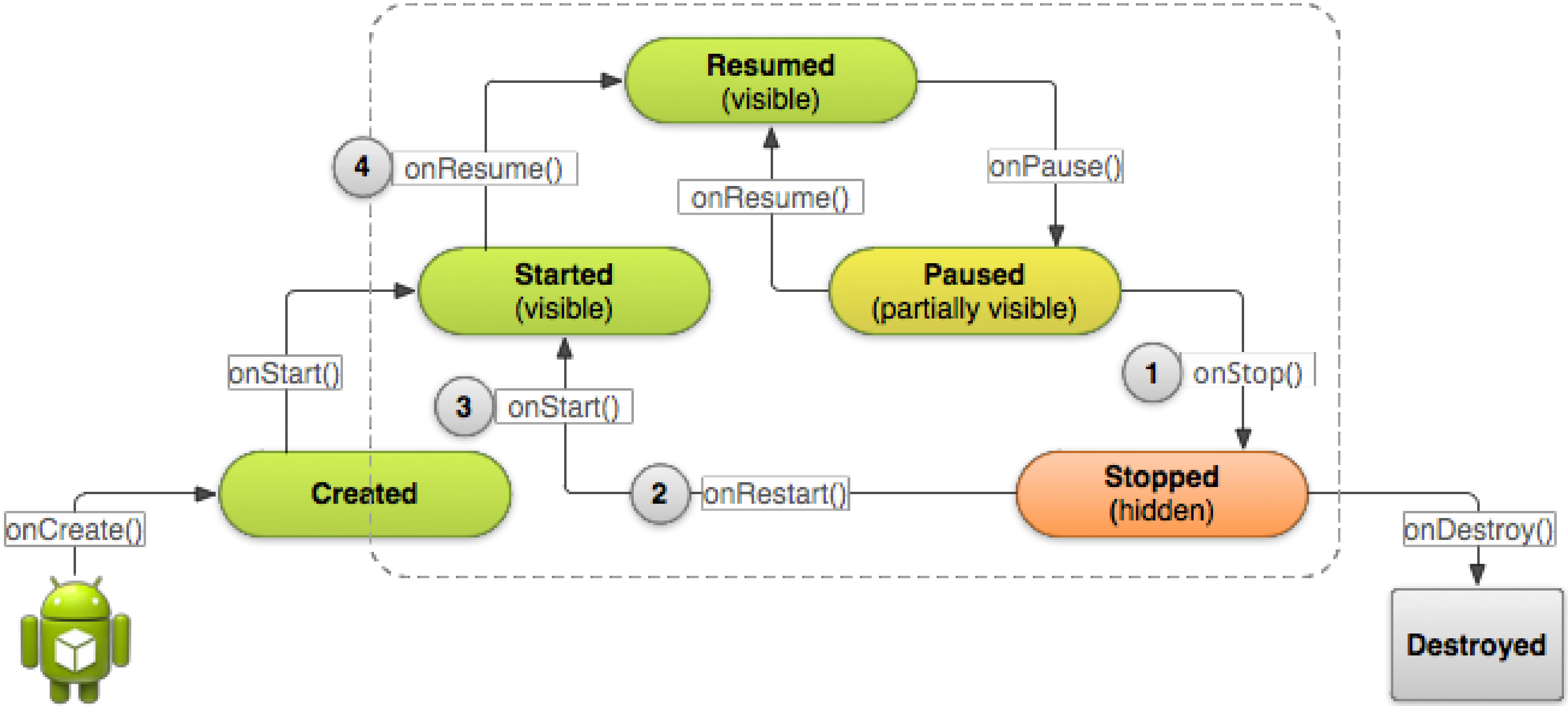
Event Handler : Listener behaviour

```
public class MainActivity extends AppCompatActivity {
    public ImageView img1;
    public int jumlah = 0 ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
       // kode program lainnya
       img1 = (ImageView) findViewById(R.id.imglogo);
       img1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                jumlah++;
                Toast.makeText(getApplicationContext(),
                        "ANDA CLICK GAMBAR !! " + jumlah + " kali",
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

ImageView :

Listener - Click Event

Activity LifeCycle



Creating Activity

- To create an activity, you must create a subclass of Activity class.
- In your subclass you need to implement callback methods that the system calls when the activity transitions between various states of its lifecycle.
- Two most important methods are:
  - onCreate()
  - onPause()

Method : onCreate()

- You must implement this method on your activity class.
- The system call this method when creating your activity.
- Within your implementation, you should initialize the essential components of your activity.
- Most importantly, this is where you must call `setContentView()` to define the layout for the activity's user interface.

Method : onPause()

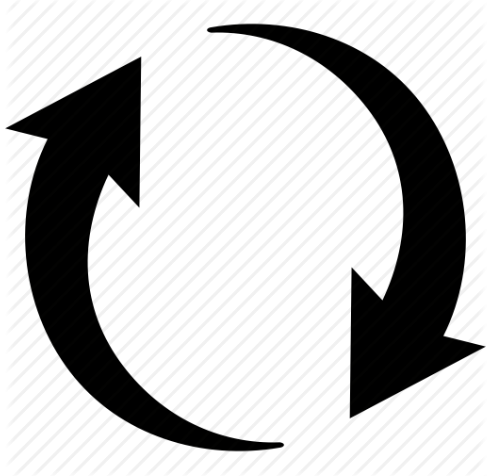
- ❑ The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed).

- ❑ This is usually where you should commit any changes that should be persisted beyond the current user session (because the user might not come back).

Method : CallBack

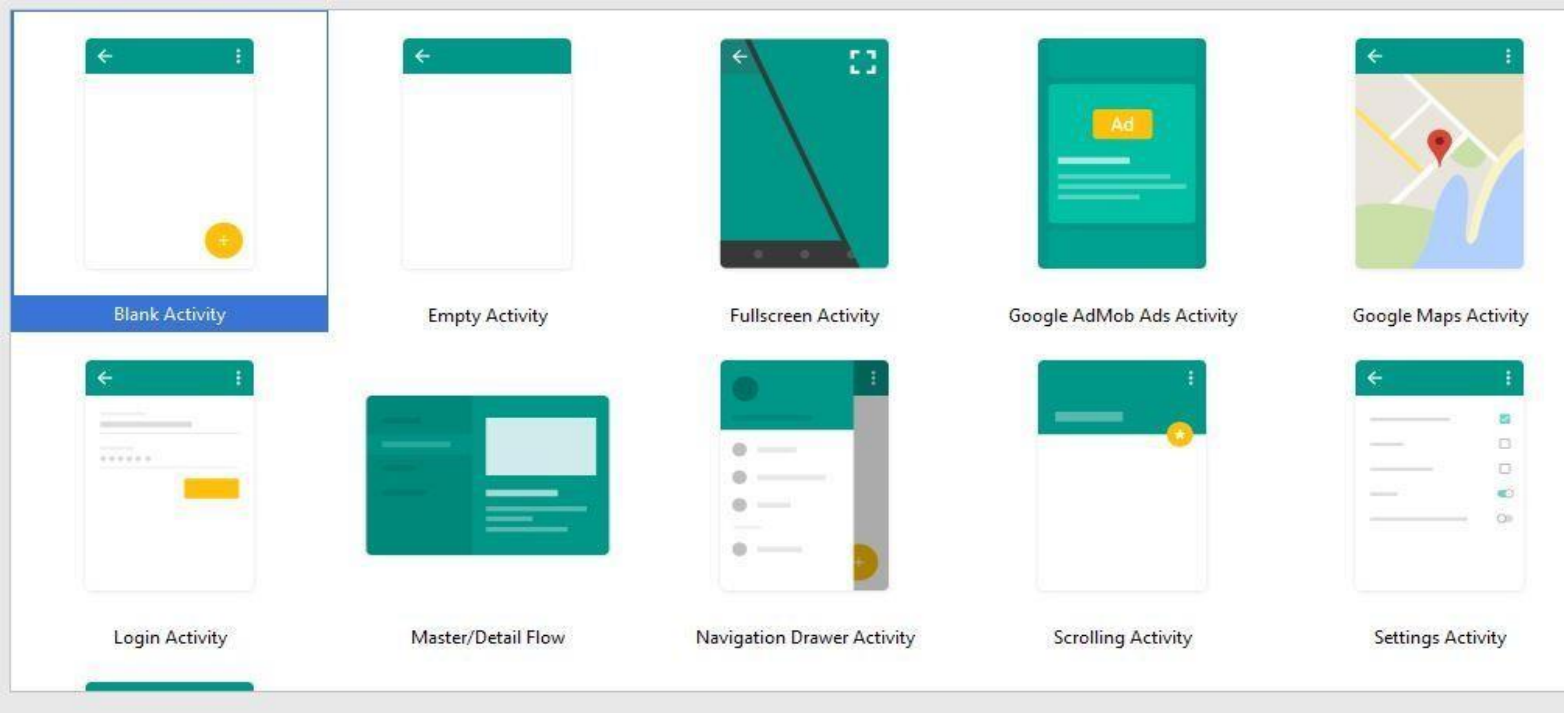
- There are several other lifecycle callback methods that you should use in order to provide a fluid user experience between activities and handle unexpected interruptions that cause your activity to be stopped and even destroyed.
- onRestart(), onStart(), onResume(), onStop(), onDestroy()

Contoh : Restart Activity

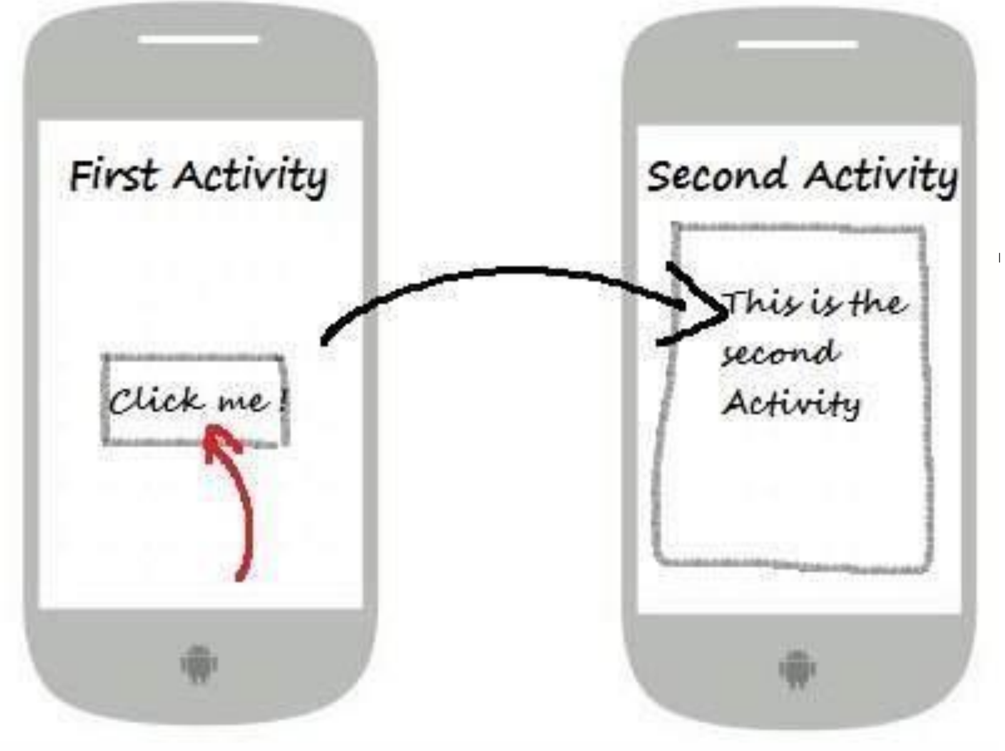


```
public void restartActivity()
{
   Intent mIntent = getIntent();
   finish();
   startActivity(mIntent);
}
```

Template Activity : Android Studio

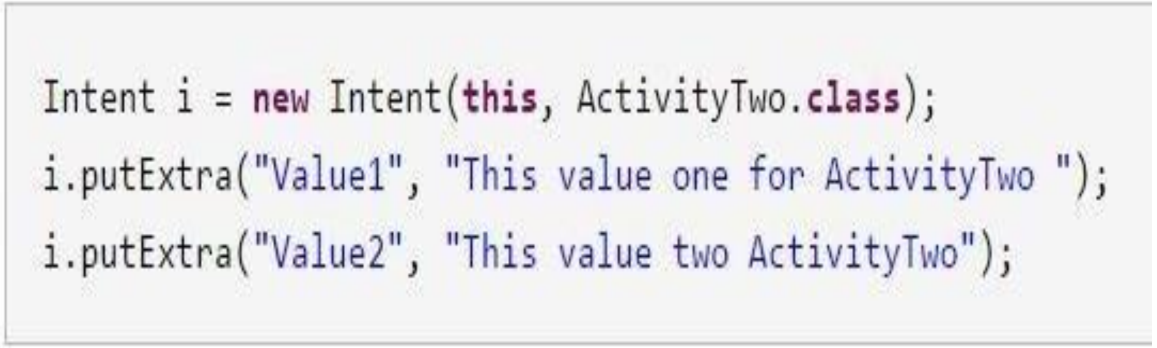


Intent

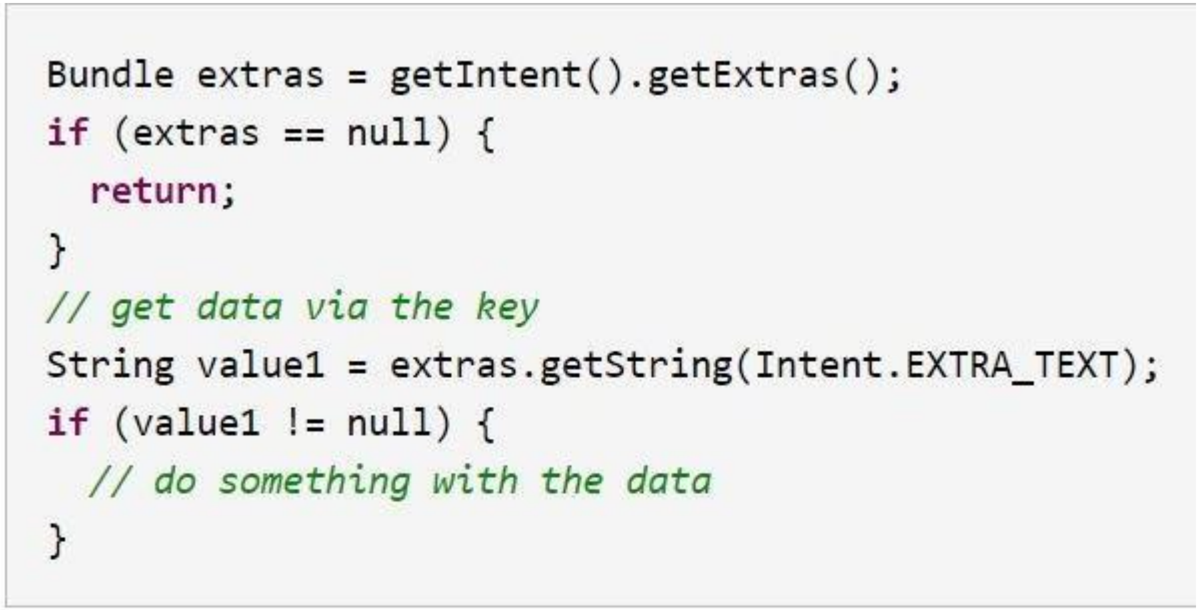


```
Intent itn = new Intent( getApplicationContext(),
                                   SecondActivity.class);
Itn.putExtra(“NOMOR”,20041);//kirim data
startActivity(itn);
```

Intent Sharing Data

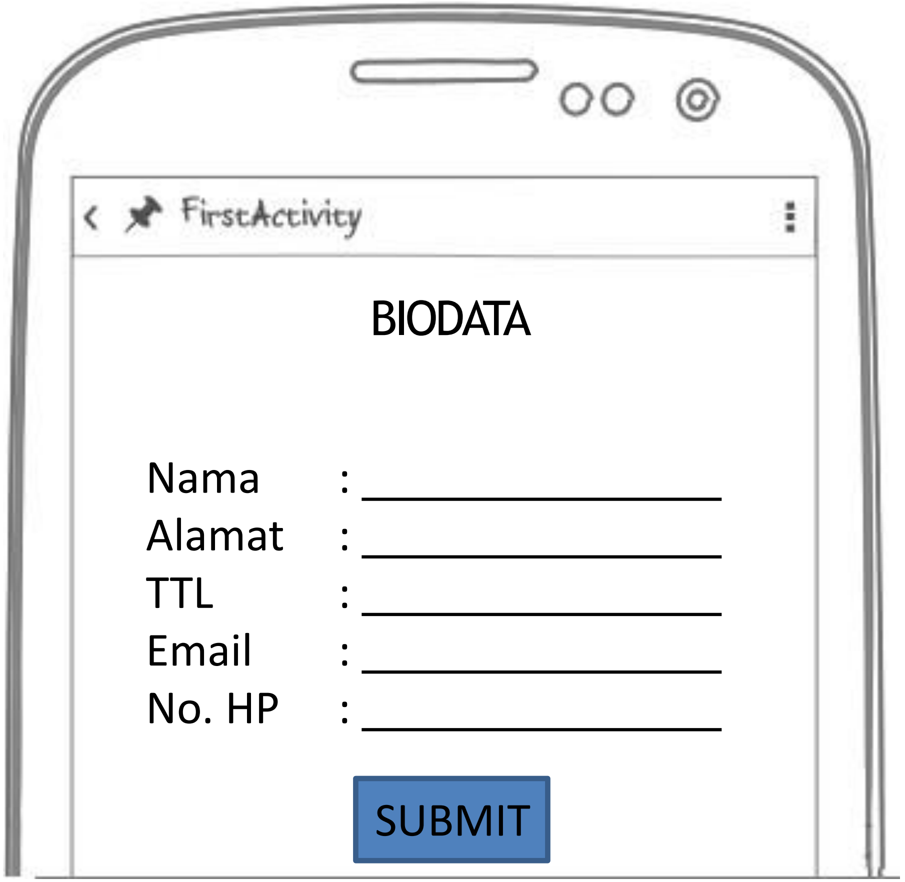


```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
i.putExtra("Value2", "This value two ActivityTwo");
```



```
Bundle extras = getIntent().getExtras();
if (extras == null) {
  return;
}
// get data via the key
String value1 = extras.getString(Intent.EXTRA_TEXT);
if (value1 != null) {
  // do something with the data
}
```

Latihan



BIODATA

Nama : ________________
Alamat : ________________
TTL : ________________
Email : ________________
No. HP : ________________

1. Buatlah Second Activity yang menampilkan data-data yang telah diisi pada First Activity
2. Beri nama project yang dibuat : NamaMahasantriActAndroid
3. Project dikirimkan dalam bentuk zip paling lambat sebelum pertemuan tanggal 1 Februari 2021 ke pijar
4. Dilarang copy paste project orang lain, jika terbukti maka nilai pemilik project dan pelaku copy-paste akan dikurangi sebanyak 50%