

MODUL PRAKTIKUM
MOBILE PROGRAMMING



OLEH:
TIM PEMBUAT MODUL PRAKTIKUM
LABORATORIUM TEKNIK
INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BHAYANGKARA
SURABAYA 2021

MODUL 1

LAYOUT

1. LINEAR LAYOUT

Vertical Linear Layout

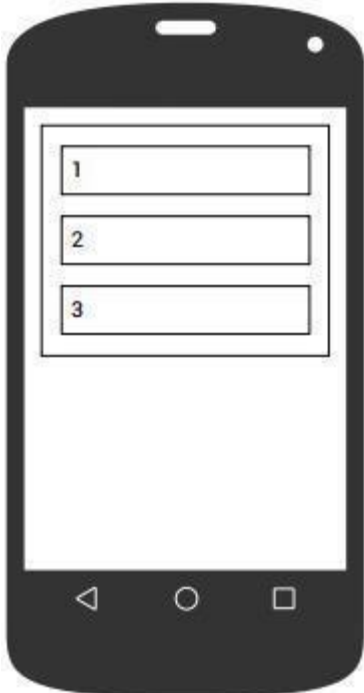


Figure 1.1 Vertical Linear Layout

A vertical LinearLayout arranges its children in a column.

```
<LinearLayout  
  android:layout_width="match_parent"  
  android:layout_height="wrap_content"  
  android:orientation="vertical">  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="1"/>  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="2"/>  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="3"/>  
</LinearLayout>
```

Horizontal Linear Layout

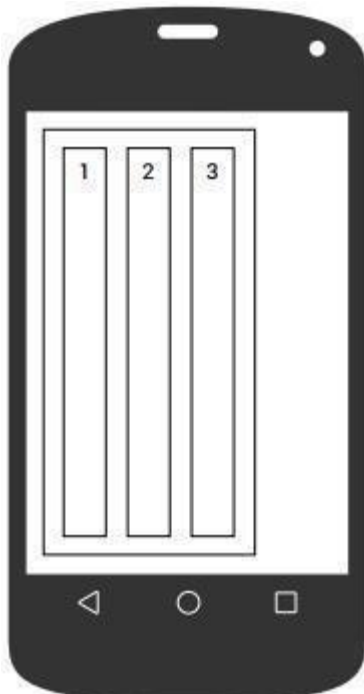
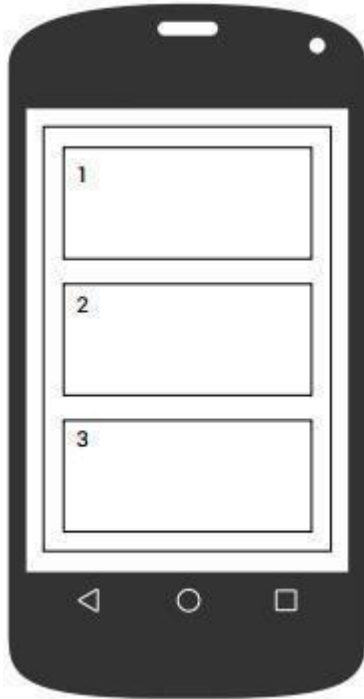


Figure 1.2 Horizontal Linear Layout

A horizontal LinearLayout arranges its children in a row.

```
<LinearLayout  
  android:layout_width="wrap_content"  
  android:layout_height="match_parent"  
  android:orientation="horizontal">  
  <TextView  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:text="1"/>  
  <TextView  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:text="2"/>  
  <TextView  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:text="3"/>  
</LinearLayout>
```

Vertical Linear Layout: Equal High

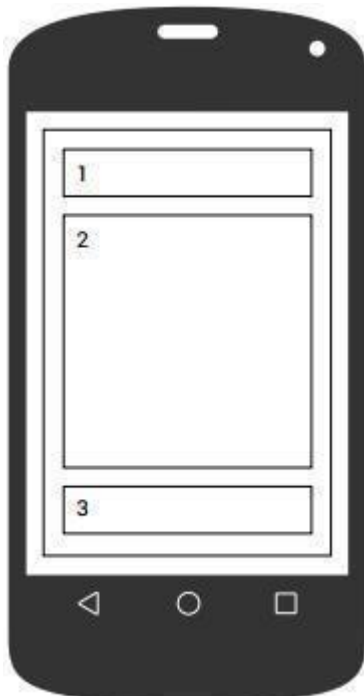


A vertical LinearLayout can give all of its children equal height.

```
<LinearLayout  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:orientation="vertical">  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:text="1"/>  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:text="2"/>  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:text="3"/>  
</LinearLayout>
```

Figure 1.3 Vertical Linear Layout: Equal High

Vertical Linear Layout: Leftover High

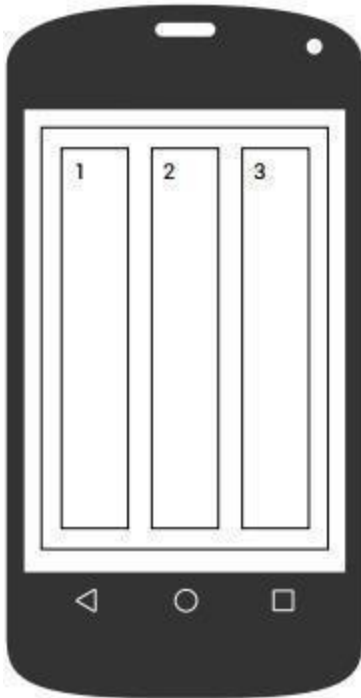


A vertical LinearLayout can give one of its children all the leftover height.

```
<LinearLayout  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:orientation="vertical">  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="0"  
    android:text="1"/>  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:text="2"/>  
  <TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="0"  
    android:text="3"/>  
</LinearLayout>
```

Figure 1.4 Vertical Linear Layout: Leftover High

Horizontal Linear Layout: Equal Width

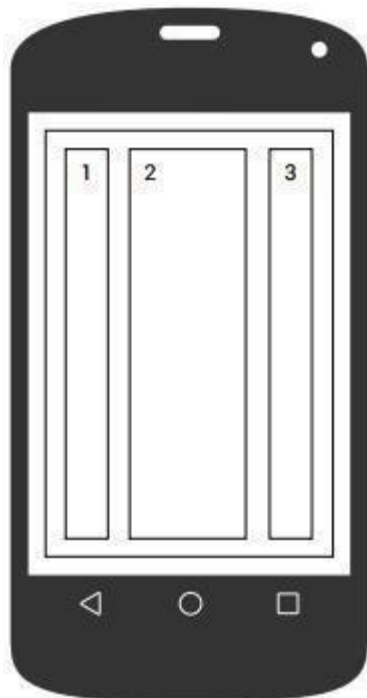


A horizontal LinearLayout can give all of its children equal width.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="1"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="2"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="3"/>
</LinearLayout>
```

Figure 1.5 Horizontal Linear Layout: Equal Width

Horizontal Linear Layout: Leftover Width



A horizontal LinearLayout can give one of its children all the leftover width.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="0"
        android:text="1"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="2"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="0"
        android:text="3"/>
</LinearLayout>
```

Figure 1.6 Horizontal Linear Layout: Leftover Width

Relative Layout

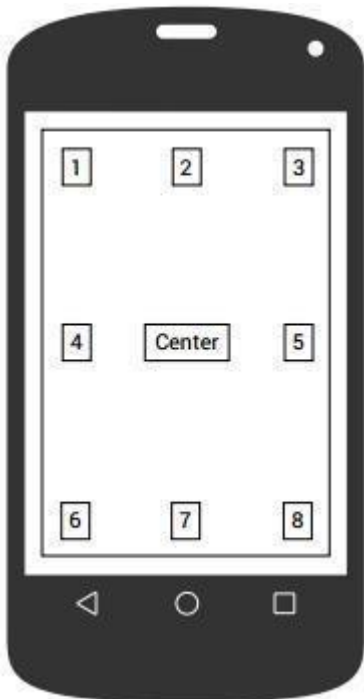


Figure 1.7 Relative Layout model 1

A RelativeLayout can position a child relative to the RelativeLayout.

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:text="1"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="2"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:text="3"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:text="4"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Center"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"
        android:text="5"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:text="6"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="7"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:text="8"/>
</RelativeLayout>
```

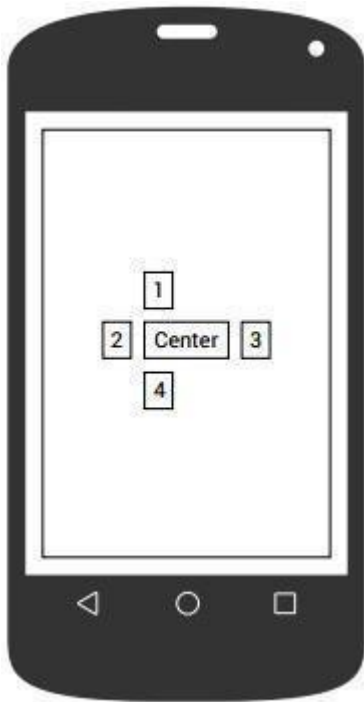


Figure 1.8Relative Layout model 2

```
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
android:id="@+id/center"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true"
android:text="Center"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_toLeftOf="@id/center"
android:layout_alignBottom="@id/center"
android:text="2"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_toRightOf="@id/center"
android:layout_alignBottom="@id/center"
android:text="3"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@id/center"
android:layout_alignLeft="@id/center"
android:text="1"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/center"
android:layout_alignLeft="@id/center"
android:text="4"/>
</RelativeLayout>
```

See the UI Overview and Layouts Guide. The attributes for the children of each type of layout are listed in the LayoutParams classes.

- For LinearLayout, see the LinearLayout Guide, class LinearLayout and its source code, and class LinearLayout.LayoutParams and its source code.
- For RelativeLayout, see the RelativeLayout guide, class RelativeLayout and its source code, and class RelativeLayout.LayoutParams and its source code. The Views in the above images are outlined only for clarity. Additional code would be needed to draw the black borders and the margins between them.

2. INFLATER LAYOUT

LayoutInflater is one of the classes or libraries used to create or convert xml layout files, as new View objects, in the main layout or can be termed layout stack. The Inflater layout is needed when someone wants to build the user interface dynamically (programmatically). The use of Layout Inflater has become a common thing when it wants to develop more complex Applications.

Suppose that in the creation of an application that uses a ListView there are two layouts, the first is the layout in charge of the main layout (eg: activity_main.xml), and has a ListView container, then the layout containing the content (eg item_list.xml), which can load TextView, ImageView and others.

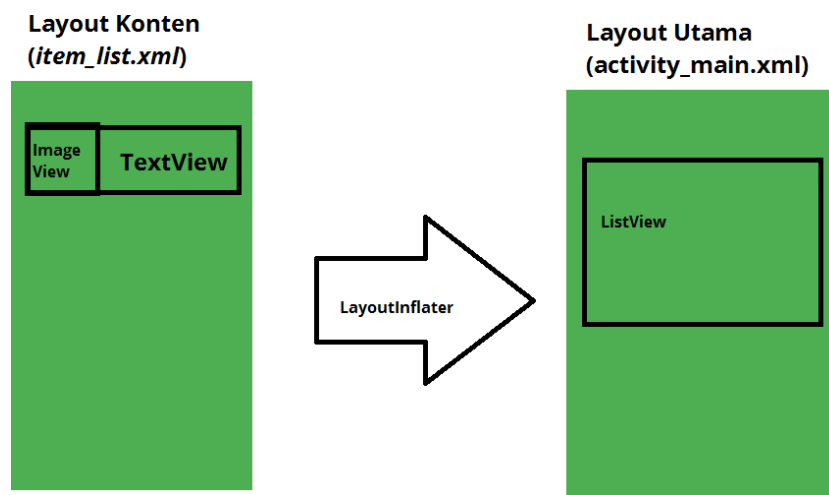


Figure 1.9 InflaterLayout Example

InflaterLayout java source code example:

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.RelativeLayout;

public class MainActivity extends AppCompatActivity {

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

        getSupportActionBar().setTitle("Contoh LayoutInflater");
        getSupportActionBar().setSubtitle("Belajar di okedroid.com");

        RelativeLayout relativeLayout =
        (RelativeLayout)findViewById(R.id.activity_main);
        //1 inisialisasi layout utama
        View view =
        getLayoutInflater().inflate(R.layout.konten,relativeLayout,false);
        // 2 mengconvert layout konten
        relativeLayout.addView(view);
        //3 method dari object relativelayout untuk menambahkan child view
    }
}
```

3. DIALOG

Alert Dialog is one of the most important and fundamental components of the Android App. Which serves to alert / alert to the user, and receive confirmation of the button action of the user of the Application. Here's how to create it.

1. First we create 1 xml layout file then copy the instruction line below:

main_alert.xml

```
<< <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"
tools:context=".AlertMain" >
<Button android:id="@+id/one_button_alert" android:layout_width="200dip"
android:layout_height="wrap_content" android:layout_alignParentLeft="true"
android:layout_marginTop="25dip" android:layout_marginLeft="20dip"
android:text="One Button Alert" />
<Button android:id="@+id/two_button_alert" android:layout_width="200dip"
android:layout_height="wrap_content" android:layout_alignParentLeft="true"
android:layout_below="@id/one_button_alert" android:layout_marginLeft="20dip"
android:text="Two Button Alert" />
<Button android:id="@+id/three_button_alert" android:layout_width="200dip"
android:layout_height="wrap_content" android:layout_alignParentLeft="true"
android:layout_below="@id/two_button_alert" android:layout_marginLeft="20dip"
android:text="Three Button Alert" />
</RelativeLayout>
```


2. Second we create the java file and copy the instruction line below:

```

package com.okedroid.myapplication;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
/** * Created by FATHUR on 12/16/2015. */public class AlertMain extends
Activity {
private Button button1;
private Button button2;
private Button button3;
@Override protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main_alert);
button1 = (Button) findViewById(R.id.one_button_alert);
button2 = (Button) findViewById(R.id.two_button_alert);
button3 = (Button) findViewById(R.id.three_button_alert);
button1.setOnClickListener(new View.OnClickListener() {
@Override public void onClick(View v) {
AlertDialog.Builder builder = new AlertDialog.Builder(
AlertMain.this);
builder.setTitle("Contoh Alert");
builder.setMessage("Alert dengan 1 Action Button ");
builder.setPositiveButton("OK",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int which) {
Log.e("info", "OK");
}
});
builder.show();
}
});

button2.setOnClickListener(new View.OnClickListener() {
@Override public void onClick(View v) {
AlertDialog.Builder builder = new AlertDialog.Builder(
AlertMain.this);
builder.setTitle("Contoh Alert");
builder.setMessage("Alert dengan 2 Action Button ");
builder.setNegativeButton("NO",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int which) {
Log.e("info", "NO");
}
});
builder.setPositiveButton("YES",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int which) {
Log.e("info", "YES");
}
});
builder.show();
}
});
}
}

```

```

button3.setOnClickListener(new View.OnClickListener() {
    @Override public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(
            AlertMain.this);
        builder.setTitle("Contoh Alert");
        builder.setMessage("Alert dengan 3 Action Button ");
        builder.setNegativeButton("NO",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int which) {
                    Log.e("info", "NO");
                }
            });
        builder.setPositiveButton("YES",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int which) {
                    Log.e("info", "YES");
                }
            });
        builder.setNeutralButton("BATAL",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int which) {
                    Log.e("info", "BATAL");
                }
            });
        builder.show();
    }
});
}

@Override public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
}

```

4. CONSTRAIN

ConstraintLayout is used to facilitate android developers, in designing user interfaces or interfaces on the Application, flexibly and dynamically, by drag and drop, without having to involve many Nested Multiple Layouts. So we no longer need to build a hierarchical layout, as in RelativeLayout, there is LinearLayout. Some View Content will be tied together and side by side. If you've ever built a UI in xcode, on an iOS device, maybe you're very familiar and familiar, and it does not take long to adapt and adjust to this layout.

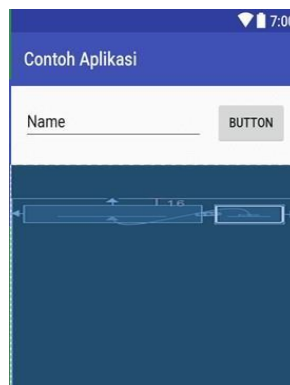


Figure 1.10designing user interfaces

In this section, explained how to build User Interface components such as EditText and Button, using the ConstraintLayout manager layout.

1. First open a project that already exists in Android Studio.
2. In build.gradle (Module App) add library from ConstraintLayout independencies section as shown below:

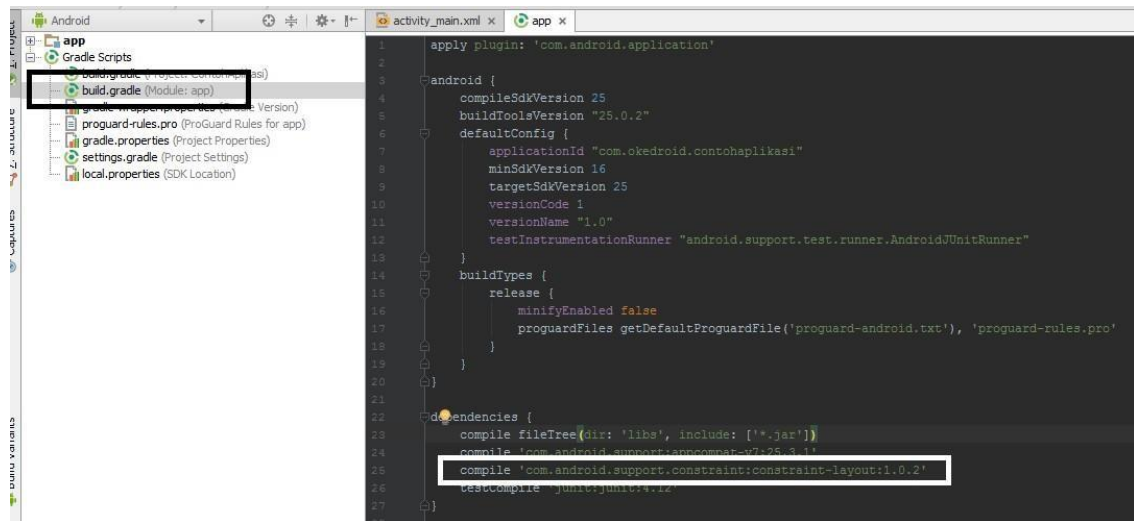


Figure 1.11add library from ConstraintLayout in dependencies

3. Then drag the ConstraintLayout on the left window of the Palette, then select Layouts, and then select ConstraintLayout. As shown below:

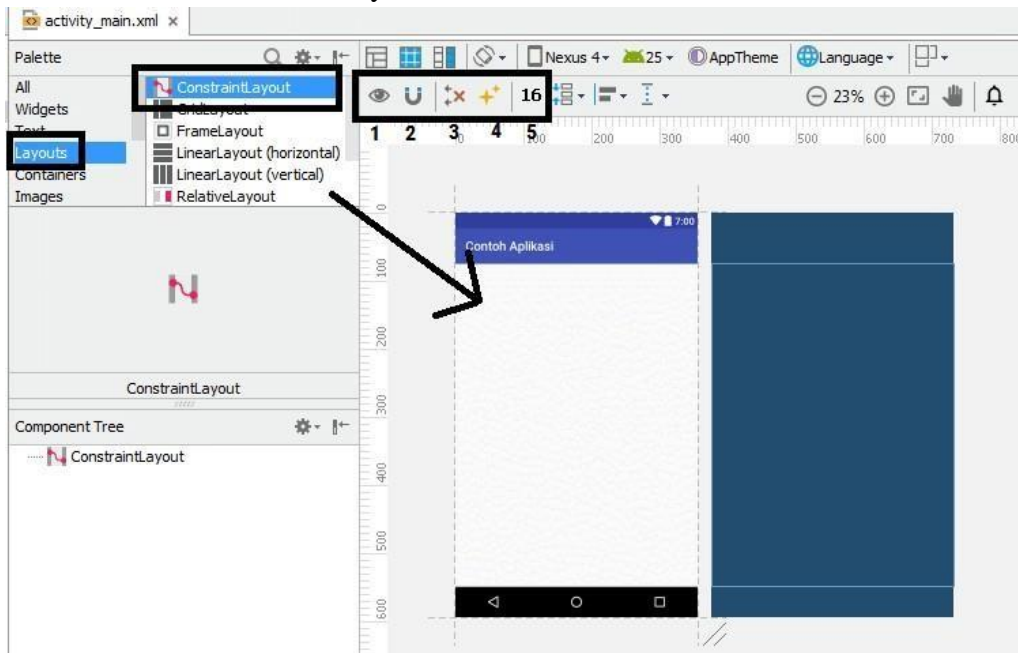


Figure 1.12Palette

There are special icons or toolbars while building User Interface with ConstraintLayout like:

- To turn it on or off, hide the constraint
- To enable or disable, Autoconnect constraint
- To delete the corresponding constraint
- To relate the constraint
- Standard size Margin in ConstraintLayout

Or you can also convert RelativeLayout to ConstraintLayout in the Component Tree window by right-clicking on the mouse, then select Convert RelativeLayout to ConstraintLayout.

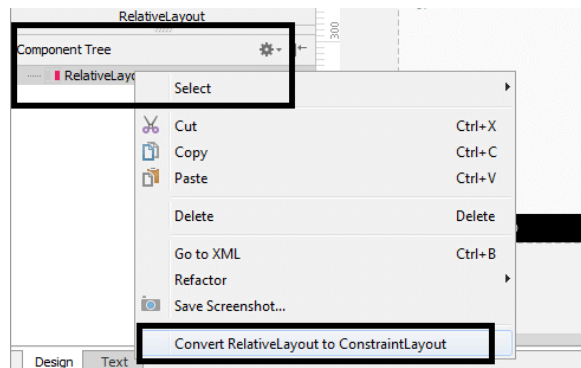


Figure 1.13 convert RelativeLayout to ConstraintLayout

4. For the source code xmlnya, you can see in the picture below:

Here we can use a special namespace, such as

xmlns: app = "http://schemas.android.com/apk/res-auto" and
 xmlns: tools = "http://schemas.android.com/tools" To use
 the attributes available in the
 ConstraintLayout library.

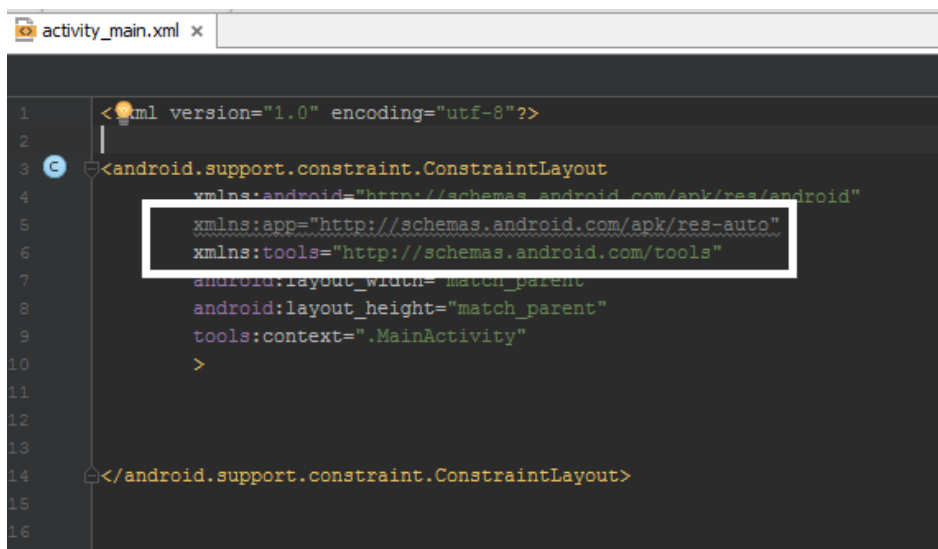


Figure 1.14 ConstraintLayout library

5. Now we will start trying to design the User Interface by drag and dro. We will try to design, using EditText and also Button on ConstraintLayout, and how to be consistent across all screen sizes on Android devices.

The source code will look like this:

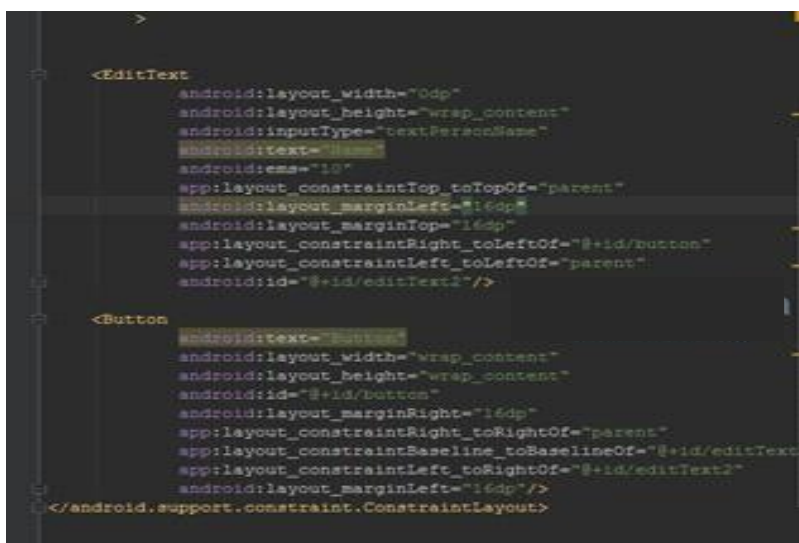


Figure 1.15 ConstraintLayout

To understand this, consider the example illustrations:

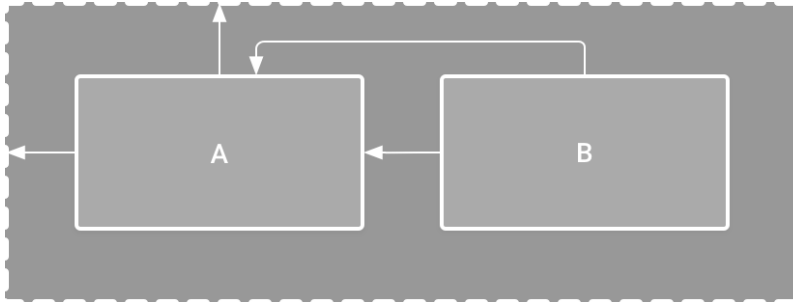


Figure 1.16 illustrations

Information :

- View A with Parent Layout has size spacing or margin 16 dp and above
- View A with Parent Layout has a spacing or margin size of 16 dp to the left
- View B with View A has a spacing or margin of 16 dp to the left
- View B is aligned at the top with View A

MODUL 2 WIDGET

1. Widget Concept

- a. Android widget has been little implemented in the activity section.
- b. Basically the widget package is a visualization of the user interface elements used on Android application screen where the programmer can design itself according to the needs of the user / client.
- c. Inside android, widgets are displayed with the View concept, where android apps use widgets as an XML layout in general.

2. The types of android widgets are as follows:

a. TextView

To display text on the screen and optionally can be edited, but by default TextView can not be edited, to be able to edit must use subclass that serves to display the contents.

Example property of TextView is android: textSize to set size, android: textStyle to set whether bold or italic font, android: textColor.

```
<!--Email label-->
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="Email"
  android:textSize="12dp"
  android:textColor="#4c4c4c"
  android:textStyle="bold"
  android:id="@+id/textView1"
  android:paddingTop="10dp"
/>
```

Figure 2.1TextView

b. EditText

EditText is a TextView customization which becomes a TextView that can configure itself so that it can be edited. Please try one by one and note the resulting XML.

```
<EditText
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_marginTop="5dip"
  android:layout_marginBottom="20dip"
  android:singleLine="true"
  android:id="@+id/editText1"
  android:layout_weight="0.06"
  android:background="#ffffff"
  android:foregroundTint="#bf2c2c" />
```

Figure 2.2EditText

c. Radio Button/ Radio Group

RadioButton is used in conjunction with RadioGroup. Inside a RadioGroup there are several RadioButton, and in one RadioGroup user can only do one check / RadioButton selection.



```

<RadioGroup
    android:id="@+id/rdgSekolah"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <RadioButton
        android:text="SMP"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/rdbSMP"
        android:layout_weight="1" />

    <RadioButton
        android:text="SMU"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/rdbSMU"
        android:layout_weight="1" />

    <RadioButton
        android:text="S1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/rdbS1"
        android:layout_weight="1" />

</RadioGroup>
    
```

Figure 2.3RadioButton

d. Image View/ Image Button

ImageView is a widget that displays images like icons.

1. ImageView can load images from various sources (resource or content Providers), ImageButton is a widget that displays the contains button Images (not text) that can be pushed or clicked by the user.
2. By default, ImageButton is almost the same as the regular button



```

<ImageView
    android:layout_width="87dp"
    android:layout_gravity="center"
    android:layout_height="78dp"
    android:src="@mipmap/ic_launcher"
    android:id="@+id/imageViewAPPLE" />
    
```

Figure 2.4Image View

e. Spinner/ ComboList

This widget is almost similar to ListView. The difference is that the ListView comes from a single combo or known as a spinner.



```

<Spinner
    android:id="@+id/planets_spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
    
```

Figure 2.5 Spinner

f. Button

Button is a derivative of TextView so applicable in textView also applies to the button. The most important addition is onClick.

```
<Button
    android:id="@+id/button"
    android:layout_width="95dp"
    android:layout_height="38dp"
    android:text="KLIK"
    tools:layout_editor_absoluteX="66dp"
    tools:layout_editor_absoluteY="41dp" />
```

Figure 2.6 Button

The result (note the use of fill_parent for layout_width attribute):



g. CheckBox

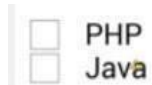
User can select more than one option with checkbox. On the palette, checkbox is in the section from widgets.



Try adding two checkboxes then set the id and text attributes through the property window:



So the result,



An example of creating a Login Form

Create Layout for login page, res / layout -> Android XML file. Name login.xml, then type the code.

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

<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"
android:padding="10dp"
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.projek.uinmaliki.tutorial.Main2Activity"
android:weightSum="1">

<!--Email Label-->
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Email"
    android:textSize="12dp"
    android:textColor="#4c4c4c"
    android:textStyle="bold"
    android:id="@+id/textView1"
    android:paddingTop="10dp"
    />

<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="20dp"
    android:singleLine="true"
    android:id="@+id/editText1"
    android:layout_weight="0.06"
    android:background="#ffffff"
    android:foregroundTint="#bf2c2c" />

<!--Password Label-->
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Password"
    android:textSize="12dp"
    android:id="@+id/textView2" />

<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText2"
    android:layout_marginTop="5dp"
    android:singleLine="true"
    android:password="true"
    android:layout_weight="0.06"
    android:background="#ffffff"
    android:foregroundTint="#bf2c2c" />
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/editText5" />

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Login"
    android:textSize="15dp"
    android:gravity="center"
    android:layout_marginTop="15dp"
    android:id="@+id/button"
    android:layout_weight="0.02"
    android:onClick="bSapaClick"
    android:background="#b4b4b4" />

<TextView
    android:id="@+id/link_to_register"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="40dip"
    android:layout_marginTop="40dip"
    android:gravity="center"
    android:text="Register, for new user !"
    android:textColor="#0b84aa"
    android:textSize="20dip" />

</LinearLayout>
```

And the results on the display of the above code is

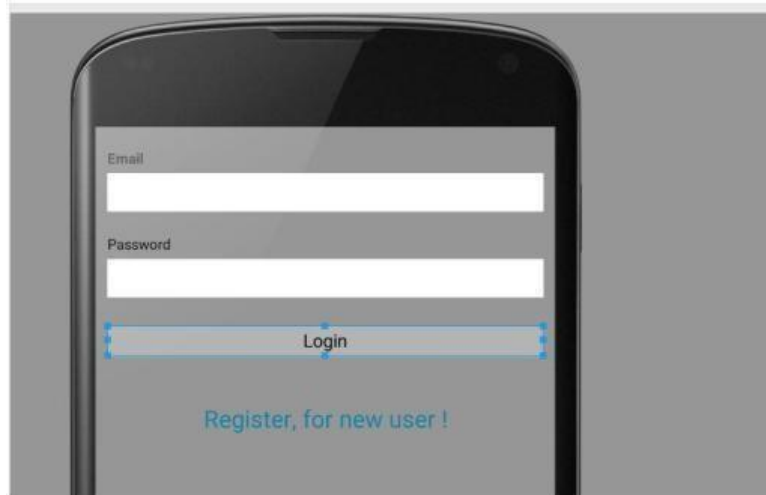


Figure 2.6 Result Display

MODUL 3 VIEW

1. SPLASH SCREEN

Splash Screen is an activity that is set as an initial layer or an opening. Splash Screen only appears in a short time and then move on to the next activity. Splash Screen is used as a means of checking whether the application is ready or not to load the data. Examples of Splash screen quite a lot, such as screen loading, screen welcome, and others. Along with the antenna intent can also be used to send data from activity to other activity.

Create a Splash Screen

1. Create an activity with an empty activity with the name of MainActivity that will be Used as a Splash screen, then edit its xml layout.

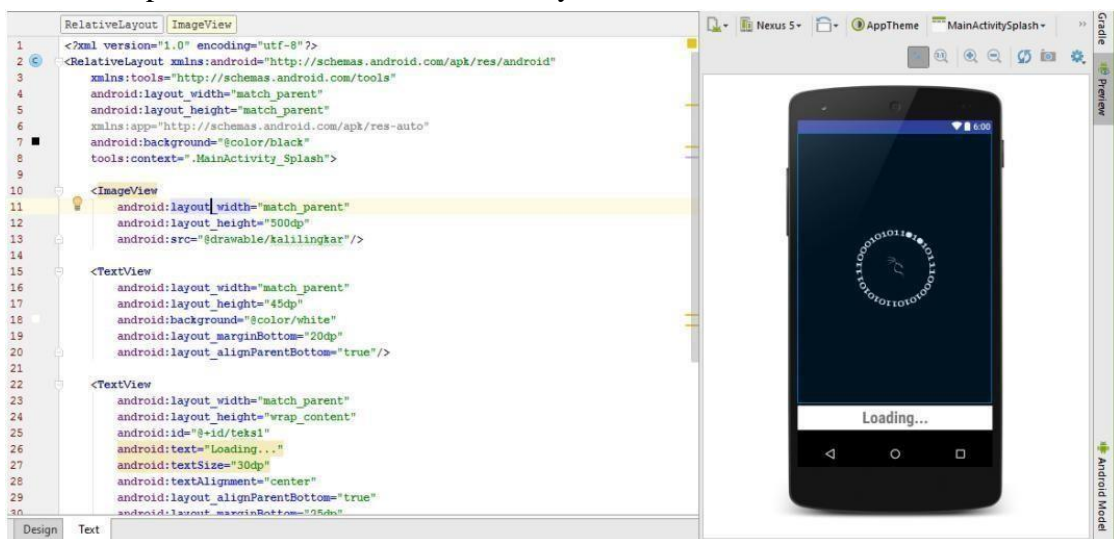


Figure 3.1 Splash Screen

2. In the java code activity_main to be used as Splash Screen, type source code for splash screen and intent, then run observe what happened.

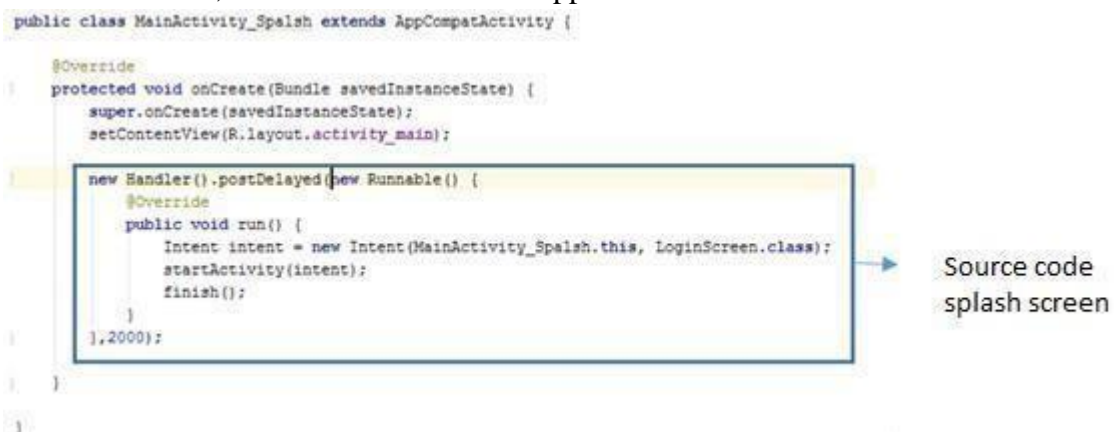


Figure 3.2 Source Code Splash Screen

From the source code above the "(MainActivity_Splash.this, LoginScreen.class)" section it shows that the intent starts from this Activity which is MainActivity_Splash and will end in Activity LoginScreen.

3. After layout and code display for Splash Screen is complete, now create a new Activity with LoginScreen name.

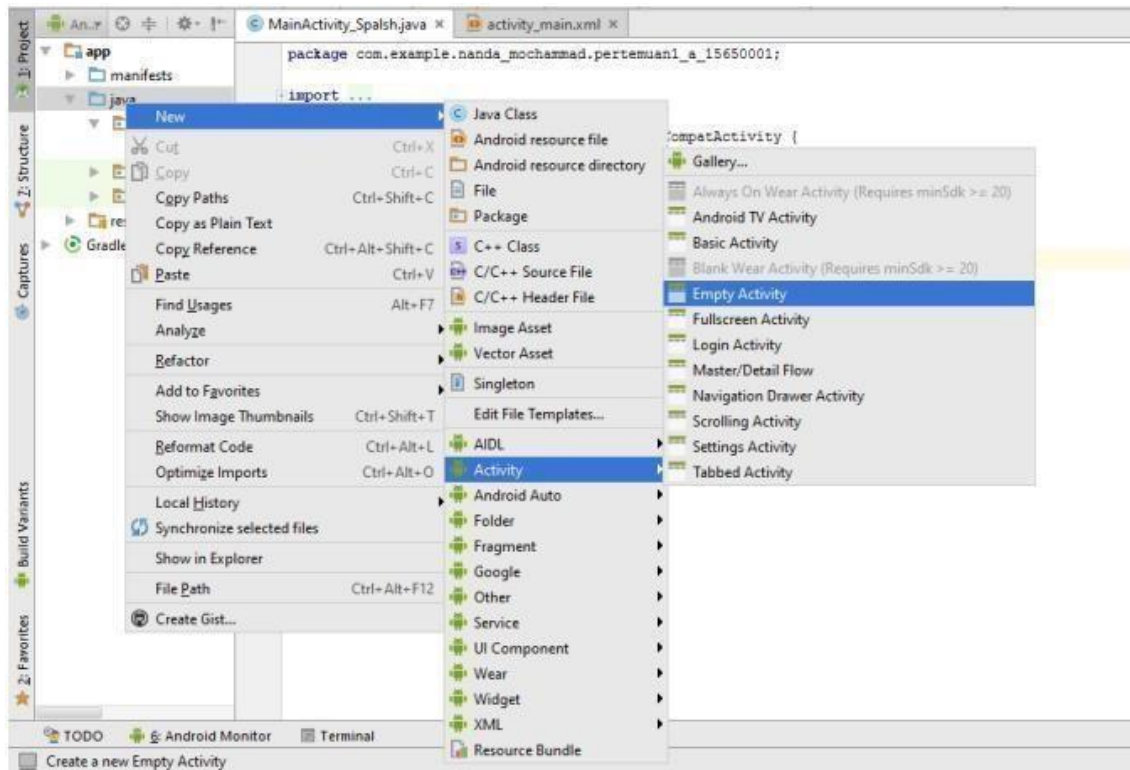


Figure 3.3 Create new empty activity

4. Then edit its xml Layout on Login Screen

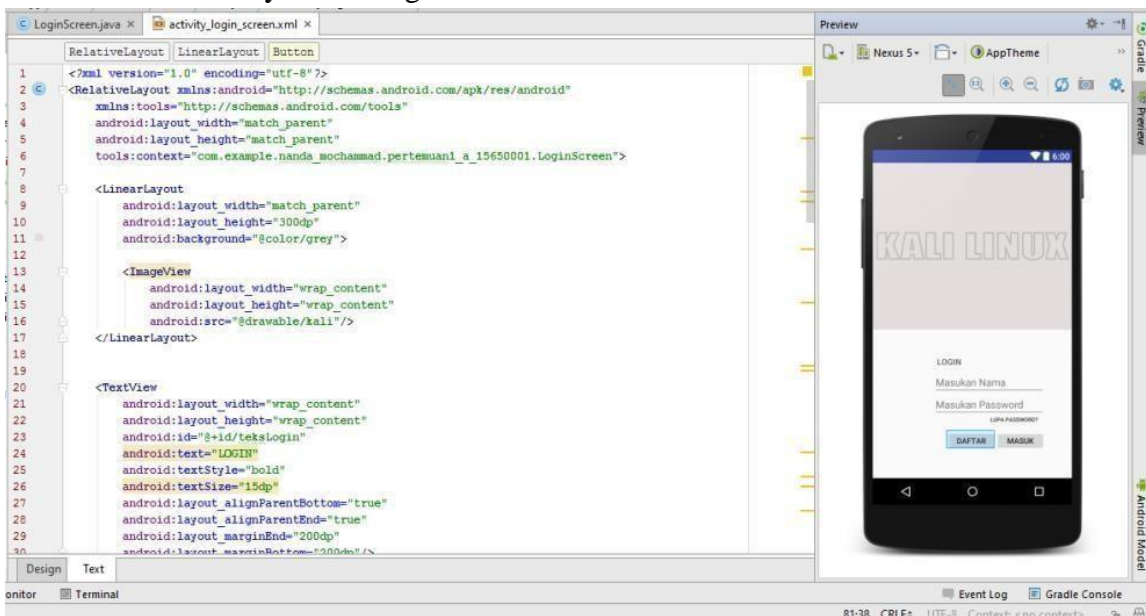


Figure 3.4 xml Layout on Login Screen

```
<EditText
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/editLogin"
    android:hint="Masukan Nama"
    android:textAlignment="textStart"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="50dp"
    android:layout_marginBottom="150dp"/>

<EditText
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/editLogin1"
    android:hint="Masukan Password"
    android:textAlignment="textStart"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="50dp"
    android:layout_marginBottom="110dp"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:id="@+id/buttonForgot"
    android:text="Lupa password?"
```

Figure 3.5 source code xml on loginscreen

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:id="@+id/buttonForgot"
    android:text="Lupa password?"
    android:textSize="10dp"
    android:keepScreenOn="true"
    style="@style/BorderlessButtonStyle"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="85dp"
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="50dp"/>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="35dp"
    android:weightSum="2"
    android:id="@+id/linearLogin"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="50dp"
    android:layout_marginEnd="50dp">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/buttonLogin"
        android:text="Daftar"/>
```

Figure 3.6 source code xml on loginscreen

```

        android:layout_alignParentEnd="true"
        android:layout_marginEnd="50dp"/>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="35dp"
            android:weightSum="2"
            android:id="@+id/linearLogin"
            android:layout_alignParentEnd="true"
            android:layout_alignParentBottom="true"
            android:layout_marginBottom="50dp"
            android:layout_marginEnd="50dp">
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:id="@+id/buttonLogin"
                android:text="Daftar"/>

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:id="@+id/buttonLogin2"
                android:text="masuk"/>
        </LinearLayout>
    </RelativeLayout>

```

Figure 3.7 source code xml on login screen

5. After Edit Layout to Login Screen, now type the java code for LoginScreen To send data.

```

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

public class LoginScreen extends AppCompatActivity {

    //Deklarasi Object yang sudah dibuat di layout agar dapat diberi aksi.
    EditText nama, password;
    Button daftar, masuk, lupa;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_screen);

        //Ini juga deklarasi nya
        nama = (EditText) findViewById(R.id.editLogin);
        password = (EditText) findViewById(R.id.editLogin1);
        daftar = (Button) findViewById(R.id.buttonLogin);
        masuk = (Button) findViewById(R.id.buttonLogin2);
        lupa = (Button) findViewById(R.id.buttonForgot);

        //Kode untuk intent. Dan mengirimkan data ke activity selanjutnya
        masuk.setOnClickListener(v -> {
            Intent intent = new Intent(LoginScreen.this, firstScreen.class);
            intent.putExtra("Nama ", nama.getText().toString());
            startActivity(intent);
        });
    }
}

```

6. Create one Activity that will be the destination of the intent. Name the Activity FirstScreen. And type java code firstScreen in accordance with the code below. Then Run from the beginning of Activity and observe what happened.

```
import ...

public class firstScreen extends AppCompatActivity {

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

        //Berikut kode menerima data dari activity sebelumnya
        String namaUser = getIntent().getStringExtra("nama");

        //Cobaan ini kode untuk menampilkan TOAST
        Toast.makeText(getApplicationContext(), "Selamat Datang" + namaUser, Toast.LENGTH_LONG).show();
    }
}
```

To receive data sent from Activity before it can be used `.getStringExtra ("DataData")`. In this example Toast used to display data that has been obtained from the previous Activity.

2. LISTVIEW

- a. Original Listview = Creating a Listview using a template already provided by Android Studio
- b. Custom List View = Create List View with customization from the developer itself

ORIGINAL LIST VIEW

Adapter is a bridge between AdapterView (sample listview) with data. This adapter provides access to data items and is also responsible for creating a View of each item in the data set.

- 1) Prepare some XML and Class needed.

Here we need 2 Class in the java folder that is "Detail Activity" and "Main Activity". Then we enter in the res - layout folder and create 3 xml ie "activity_detail.xml", "Activity_main.xml" and "item.xml".

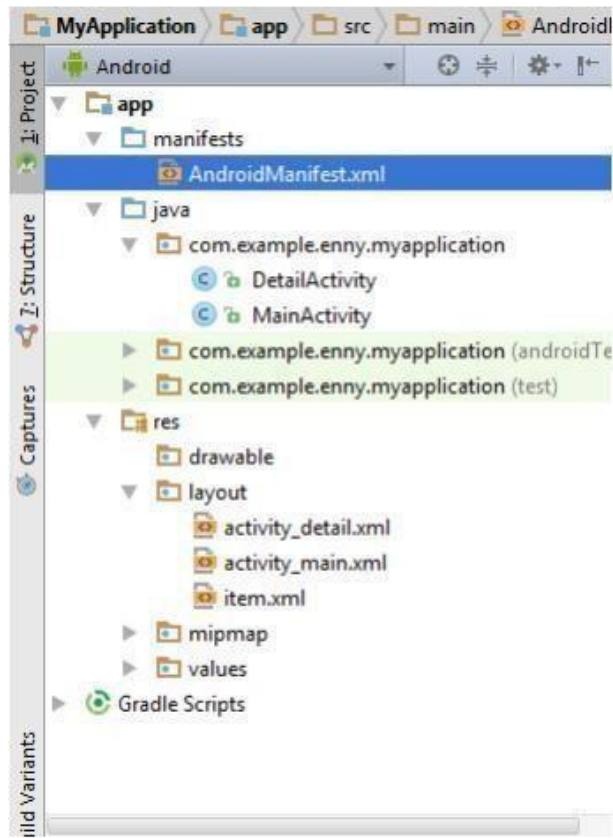


Figure 3.8 Android Manifest

2) Creating ListView

After creating some required attributes, then click the res - layout - activity_main.xml folder, write the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res-
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.enny.myapplication.MainActivity">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true">

    </ListView>
</RelativeLayout>
```



Figure 3.9 Source code android manifest

3) Create an Array then we create MainActivity.java class to create data of fruit names in array and enter the following code:


```
public class MainActivity extends AppCompatActivity {  
    Context context;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        context=MainActivity.this;  
        final String ArrayBuah[]={ "Apel", "Manggis", "Melon", "Nanas", "Jeruk" };  
    }  
}
```

Figure 3.9 source code array on MainActivity.java

4) TextView in detail.

Then we go to Activity_Detail.xml and add textView as follows:

```
<?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="16dp"  
    android:paddingLeft="16dp"  
    android:paddingRight="16dp"  
    android:paddingTop="16dp"  
    tools:context="com.example.enny.myapplication.DetailActivity">  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:id="@+id/detail"  
        android:text="Berhasil"  
        android:textStyle="bold"  
        android:textSize="35dp" />  
</RelativeLayout>
```



Figure 3.10 source code Activity_Detail.xml

5) Creating data.

In DetailActivity.java we fill in the following code:

```
public class DetailActivity extends AppCompatActivity {

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

        TextView textViewDetail = (TextView) findViewById(R.id.detail);
        textViewDetail.setText(getIntent().getStringExtra("data"));
    }
}
```

6) Creating Adapters.

In MainActivity.java add the following adapter code:

```
ListView listView =(ListView) findViewById(R.id.ListView);
ArrayAdapter<String>arrayAdapter=new ArrayAdapter<String>
    (context, R.layout.activity_main, ArrayBuah);
listView.setAdapter(arrayAdapter);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(context, "Anda Memilih: "+ ArrayBuah[position], Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(context, DetailActivity.class );
        intent.putExtra("data", ArrayBuah[position]);
        startActivity(intent);
    }
});
BaseAdapter baseAdapter = new BaseAdapter() {
    @Override
    public int getCount() {
        return ArrayBuah.length;
    }
    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View itemCustom = getLayoutInflater().inflate(R.layout.item, null);
        TextView textViewContent =(TextView)itemCustom.findViewById(R.id.textData);
        textViewContent.setText(ArrayBuah[position]);
        return itemCustom;
    }
};
listView.setAdapter(baseAdapter);
}
```

Then run. And will display as below:

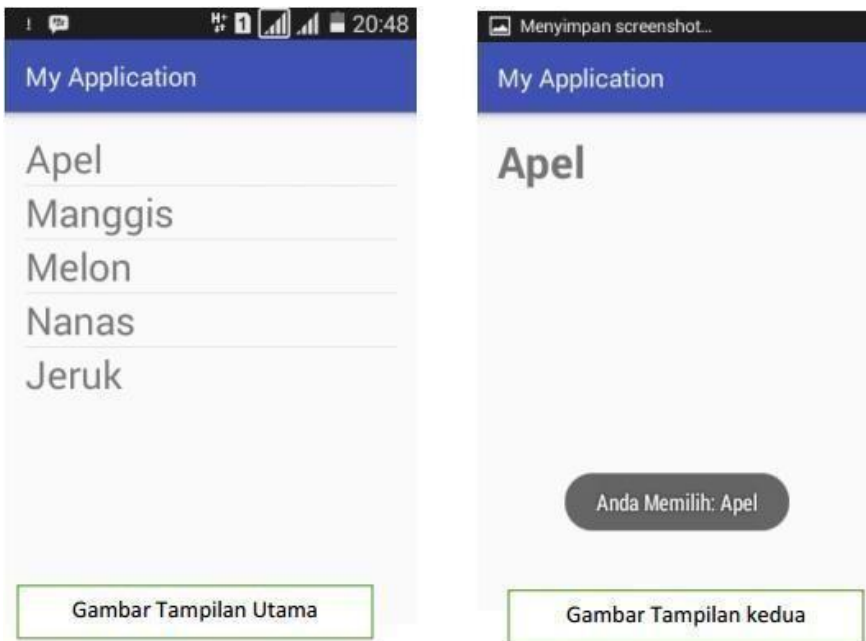


Figure 3.11 Result Display

3. CARDVIEW

CardView function as a wrapper or frame layout that will wrap the layout inside with a card-like design. If the reader sees a list of apps on the Google Play Store, the view used is CardView. The hallmark of CardView design is the presence of rounded corners and the presence of shadows for elevation effects.

Please create a new project on Android Studio with CardViewTest name, or name it as you wish. Open Android Studio, click **File - New Project** fill in Application Name with **CardViewTest** name. In the Activity window please select Empty Activity.

After that, we will add dependencies library from CardView on the project being created. Please open the **build.gradle(Module: app)** file located on the already created project. Then add the code below.

```
compile 'com.android.support:appcompat-v7:21.0.3'  
compile 'com.android.support:cardview-v7:21.0.3'  
compile 'com.android.support:support-v4:21.0.3'
```

After adding the library, please open the **activity_main.xml** file located in the **res / layout** / created project directory. Then add the code as below.

```
<?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="oc.startup.ra.cardview2.MainActivity">

    <android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/card_view"
        android:layout_width="match_parent"
        android:layout_height="380dp"
        android:layout_margin="8dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:id="@+id/thumbnail"
                android:layout_width="match_parent"
                android:layout_height="250dp"
                android:layout_alignParentTop="true"
                android:scaleType="centerCrop"
                android:src="@drawable/wallpaper" />

            <TextView
                android:id="@+id/title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/thumbnail"
                android:maxLines="3"
                android:padding="8dp"
                android:text="@string/title"
                android:textColor="#222"
                android:textStyle="bold"
                android:textSize="22dp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/title"
                android:maxLines="3"
                android:padding="8dp"
                android:text="@string/description"
                android:textColor="#666"
                android:textSize="14dp" />

        </RelativeLayout>
    </android.support.v7.widget.CardView>

</RelativeLayout>
```

By adding the source code above, we can already see the results of the CardView display created. However, to complete the display to display good results please change the source code in ImageView and TextView. In ImageView, prepare a picture or photo to be used as wallpaper wrapped by CardView. To add a picture please go to the created project directory and save it in the **drawable** folder. In this example it is located in **CardViewTest \ app \ src \ main \ res \ drawable**. Give the image a name as desired, then change the source code below according to the name of the selected image.

```
android:src="@drawable/wallpaper"
```

To complete the TextView, please open the **res / values / string.xml** directory located on the created project. Then add the code like below.

```
<string name="title">Riswan Abidin</string>
<string name="description">Seorang yang sangat tertarik dengan perkembangan teknologi dan antusias terhadap komunitas dan startup</string>
```

Name = "title" and **name = "description"** are the same on the TextView source code found on the activity_main.xml as below.

```
<TextView
    android:id="@+id/title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/thumbnail"
    android:maxLines="3"
    android:padding="8dp"
    android:text="@string/title"
    android:textColor="#222"
    android:textStyle="bold"
    android:textSize="22dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/title"
    android:maxLines="3"
    android:padding="8dp"
    android:text="@string/description"
    android:textColor="#666"
    android:textSize="14dp" />
```

Below is an app view when running on an Android emulator.

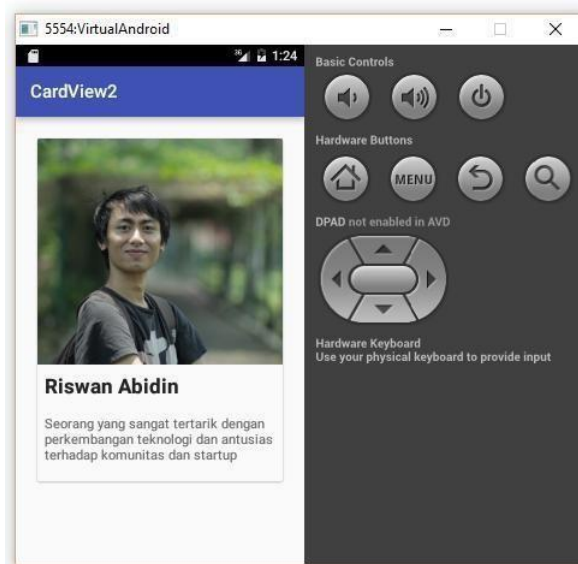


Figure 3.12 Result Display CardView

4. RECYCLERVIEW

Recyclerview is a development of listview android that has better performance and many other advantages. Recyclerview appeared since the emergence of android lollipop (Android 5.0). Recyclerview by default has no divider unlike listview.

1) Create a project

2) Dependencies

After Project is created, now open **Grandle Script > build.gradle (Module: app) >** Add Recyclerview dependency **com.android.support:recyclerview-v7:24.0.0 >** then Rebuild Project.

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:24.0.0'  
    compile 'com.android.support:recyclerview-v7:24.0.0'  
}
```

3) Create a view

Go to activity_main.xml in Res> layout> activity_main.xml to

```
<?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.support.v7.widget.RecyclerView  
        android:id="@+id/recyclerview"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent" />  
  
</RelativeLayout>
```

Create custom row_layout.xml in Res> Layout> row_layout.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#ffffff"
    android:orientation="horizontal"
    android:paddingLeft="10sp"
    android:paddingRight="10sp"
    android:paddingTop="5sp">

    <ImageView
        android:id="@+id/image_view"
        android:layout_width="45sp"
        android:layout_height="45sp"
        android:layout_gravity="center"
        android:src="@android:drawable/ic_lock_idle_low_battery" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="2sp"
        android:orientation="vertical"
        android:paddingLeft="10sp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:paddingBottom="10sp">

            <LinearLayout
                android:id="@+id/linearLayout"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/txtDate"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:singleLine="true"
                    android:text="Nama"
                    android:textStyle="bold" />

                <TextView
                    android:id="@+id/txtDay"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="status" />

            </LinearLayout>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_alignTop="@+id/linearLayout"
                android:text="Mobile" />
        </RelativeLayout>

        <View
            android:layout_width="match_parent"
            android:layout_height="0.5dp"
            android:background="@android:color/darker_gray"
            android:paddingLeft="20sp" />
    </LinearLayout>
</LinearLayout>

```


4) Create a Setter Getter Class

Create a Class Model or what we call the setter getter, eg name Contact.java.

```
package com.imamfarisi.whatsupcontact;

/**
 * Created by imam-pc on 04/07/2016.
 */
public class Contact {
    private String nama, status, tipePhone;
    private Integer photoPic;

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getTipePhone() {
        return tipePhone;
    }

    public void setTipePhone(String tipePhone) {
        this.tipePhone = tipePhone;
    }

    public Integer getPhotoPic() {
        return photoPic;
    }

    public void setPhotoPic(Integer photoPic) {
        this.photoPic = photoPic;
    }
}
```

5) Create a RecyclerView Adapter

Then we make a Special Adapter for Recyclerview, for example we named RecyclerviewAdapter.java.

```
package com.imamfarisi.whatsappcontact;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.support.v4.graphics.drawable.RoundedBitmapDrawable;
import android.support.v4.graphics.drawable.RoundedBitmapDrawableFactory;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.List;

/**
 * Created by imam-pc on 04/07/2016.
 */
public class RecyclerviewAdapter extends RecyclerView.Adapter<RecyclerviewAdapter.MyViewHolder> {

    private List<Contact> contactList;
    private Context ctx;

    public class MyViewHolder extends RecyclerView.ViewHolder {
        public TextView txtNama, txtStatus, txtTipePhone;
        public ImageView imgView;

        public MyViewHolder(View view) {
            super(view);
            txtNama = (TextView) view.findViewById(R.id.txtNama);
            txtStatus = (TextView) view.findViewById(R.id.txtStatus);
            txtTipePhone = (TextView) view.findViewById(R.id.txtPhoneType);
            imgView = (ImageView) view.findViewById(R.id.imgView);
        }
    }

    public RecyclerviewAdapter(List<Contact> contactList, Context ctx) {
        this.ctx = ctx;
        this.contactList = contactList;
    }
}
```

```

@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.row_layout, parent, false);

    return new MyViewHolder(itemView);
}

@Override
public void onBindViewHolder(MyViewHolder holder, int position) {
    Contact contact = contactList.get(position);
    holder.txtNama.setText(contact.getNama());
    holder.txtStatus.setText(contact.getStatus());
    holder.txtTipePhone.setText(contact.getTipePhone());

    Bitmap srcBitmap = BitmapFactory.decodeResource(ctx.getResources(), contact.getPhotoPic());
    holder.imgView.setImageBitmap(srcBitmap);
    circularImage(holder.imgView, srcBitmap);
}

@Override
public int getItemCount() {
    return contactList.size();
}

private void circularImage(ImageView imageView, Bitmap srcBitmap) {
    Paint paint = new Paint();

    int srcBitmapWidth = srcBitmap.getWidth();
    int srcBitmapHeight = srcBitmap.getHeight();

    int borderWidth = 30;

    // destination bitmap width
    int dstBitmapWidth = Math.min(srcBitmapWidth, srcBitmapHeight) + borderWidth * 2;
    //float radius = Math.min(srcBitmapWidth,srcBitmapHeight)/2;

    // Initializing a new bitmap to draw source bitmap, border and shadow
    Bitmap dstBitmap = Bitmap.createBitmap(dstBitmapWidth, dstBitmapWidth, Bitmap.Config.ARGB_8888);

    // Initialize a new canvas
    Canvas canvas = new Canvas(dstBitmap);

    // Draw a solid color to canvas
    canvas.drawColor(Color.WHITE);

    // Draw the source bitmap to destination bitmap by keeping border and shadow spaces
    canvas.drawBitmap(srcBitmap, (dstBitmapWidth - srcBitmapWidth) / 2, (dstBitmapWidth - srcBitmapHeight) / 2, null);

    // Use Paint to draw border
    paint.setStyle(Paint.Style.STROKE);
    paint.setStrokeWidth(borderWidth * 2);
    paint.setColor(Color.WHITE);

    // Draw the border in destination bitmap
    canvas.drawCircle(canvas.getWidth() / 2, canvas.getHeight() / 2, canvas.getWidth() / 2, paint);

    // Use Paint to draw shadow
    paint.setColor(Color.LTGRAY);

    // Draw the shadow on circular bitmap
    canvas.drawCircle(canvas.getWidth() / 2, canvas.getHeight() / 2, canvas.getWidth() / 2, paint);

    RoundedBitmapDrawable roundedBitmapDrawable = RoundedBitmapDrawableFactory.create(ctx.getResources(), dstBitmap);

    // Make the ImageView image to a circular image
    roundedBitmapDrawable.setCircular(true);

    roundedBitmapDrawable.setAntiAlias(true);

    // Set the ImageView image as drawable object
    imageView.setImageDrawable(roundedBitmapDrawable);
}
}

```

6) Create MainActivity.java

Then Last step is to register recyclerview to MainActivity.java.

```
package com.imamfarisi.whatsappcontact;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;

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

public class MainActivity extends AppCompatActivity {

    private List<Contact> contactList = new ArrayList<>();
    private RecyclerView recyclerView;
    private RecyclerViewAdapter mAdapter;

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

        recyclerView = (RecyclerView) findViewById(R.id.recyclerview);

        setData();
    }

    private void setData() {
        Contact dataSpiderman = new Contact("Spiderman", "Homecoming", "MOBILE", R.drawable.spiderman);
        Contact dataThor = new Contact("Thor", "Ragnarok", "MOBILE", R.drawable.thor);
        Contact dataIronMan = new Contact("Ironman", "Bukan Besi Biasa", "MOBILE", R.drawable.ironman);
        Contact dataCaptain = new Contact("Captain America", "Civil War", "MOBILE", R.drawable.captain);
        Contact dataFlash = new Contact("The Flash", "Speed is Everything", "MOBILE", R.drawable.flash);

        contactList.add(dataSpiderman);
        contactList.add(dataThor);
        contactList.add(dataIronMan);
        contactList.add(dataCaptain);
        contactList.add(dataFlash);

        mAdapter = new RecyclerViewAdapter(contactList, this);
        RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setAdapter(mAdapter);
    }
}
```

Once executed then the result will be like this

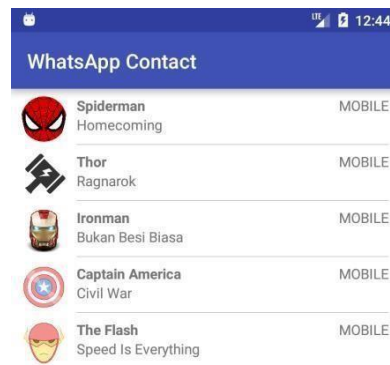


Figure 3.12 Result Display RecyclerView

5. NAVIGATION DRAWER

Navigation Drawer View is one of the navigation menu components or so-called sliding menu that serves to wrap and navigate a content inside Activity or Fragment, from Applications. Navigation drawer is now supported by the Material Design Library. Which belong to the Appcompat library (v21). Navigation Drawer View is often used in popular applications such as BBM. To use the user (user) just need to move your thumb or finger right then Navigation Drawer View will appear.

In this android learning tutorial, we will try to create and implement Navigation Drawer View with Android Material Design support.

1) Setup Gradle

Before getting started to by default make sure on the build gradle especially in the dependencies section you settings like this:

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. dependencies {
2.     compile fileTree(dir: 'libs', include: ['*.jar'])
3.     testCompile 'junit:junit:4.12'
4.     compile 'com.android.support:appcompat-v7:23.3.0'
5.     compile 'com.android.support:design:23.3.0'
6. }
```

2) Setup resource values

In the folder values (app / res / values) section you apply the instruction row (codingan) below on each file:

styles.xml

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <resources>
2.
3.     <!-- Base application theme. -->
4.     <style name="AppTheme" parent="Theme.AppCompat.NoActionBar">
5.         <item name="colorPrimary">@color/colorPrimary</item>
6.         <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
7.         <item name="colorAccent">@color/colorAccent</item>
8.         <!-- Customize your theme here. -->
9.     </style>
10.
11. </resources>
```

Karna kita nanti akan membuat toolbar, maka kita set atau pilih thema "Theme.AppCompat.NoActionBar"

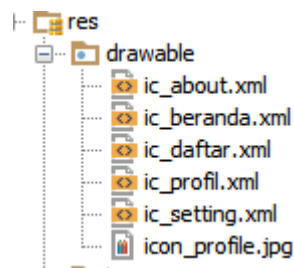
strings.xml

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <resources>
2.   <string name="app_name">"Contoh Navigation Drawer "</string>
3.   <string name="openDrawer">Drawer dibuka</string>
4.   <string name="closeDrawer">Drawer ditutup</string>
5.     <string name="navigation_view_item_1">Beranda</string>
6.     <string name="navigation_view_item_2">Profil</string>
7.     <string name="navigation_view_item_3">Daftar</string>
8.   <string name="navigasi_kategori_2">Sub Item Navigasi</string>
9.   <string name="navigation_view_item_4">Settings</string>
10.  <string name="navigation_view_item_5">About</string>
11.
12.
13. </resources>
```

This file is used to store resource or text data from menu navigation items.

3) Prepare the icon / drawing in drawable

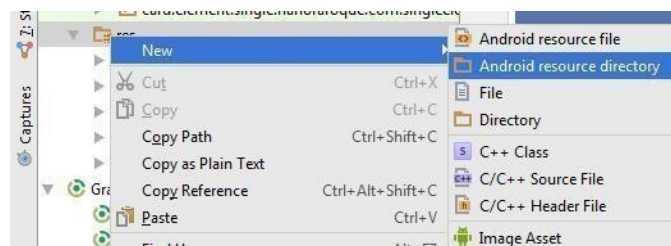
Prepare a .png or vector .xml image and save it in the folder (app / res / drawable).
Examples like this:



4) Creating a menu folder

By default (default) if we select Empty Activity, we do not include menu folder in resource, so we will make it manually by:

In the res folder in your Android Studio project structure, right click and select New> select Android Resource Directory as in the picture below:



Select the menu in the resource type and select the OK button.

item_navigasi.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <menu xmlns:android="http://schemas.android.com/apk/res/android">
3.     <group android:checkableBehavior="all">
4.         <item
5.             android:id="@+id/navigation1"
6.             android:checked="false"
7.             android:icon="@drawable/ic_beranda"
8.             android:title="@string/navigation_view_item_1" />
9.         <item
10.            android:id="@+id/navigation2"
11.            android:checked="false"
12.            android:icon="@drawable/ic_profil"
13.            android:title="@string/navigation_view_item_2" />
14.         <item
15.            android:id="@+id/navigation3"
16.            android:checked="false"
17.            android:icon="@drawable/ic_daftar"
18.            android:title="@string/navigation_view_item_3" />
19.     </group>
20.
21.
22.
23.
24.     <item
25.         android:id="@+id/navigasi_kategori_2"
26.         android:title="@string/navigasi_kategori_2">
27.             <menu>
28.                 <item
29.                     android:id="@+id/navigation4"
30.                     android:checked="false"
31.                     android:icon="@drawable/ic_setting"
32.                     android:title="@string/navigation_view_item_4" />
33.                 <item
34.                     android:id="@+id/navigation5"
35.                     android:checked="false"
36.                     android:icon="@drawable/ic_about"
37.                     android:title="@string/navigation_view_item_5" />
38.             </menu>
39.         </item>
40.
41.
42. </menu>
    
```

In the file we will define any navigation menu item, which we will display.

- **android: id = "@ + id / navigation1"** = id identification which will be called in the Java Activity file.
- **android: icon = "@ drawable / ic_beranda"** = call the vector image icon file contained in the drawable resource folder we have created.
- **android: title = "@ string / navigation_view_item_1"** = calling the text data file contained in the resource strings.xml folder.

5) Layout

In the layout folder, we will apply the instruction line (codingan) below in the activity_main.xml file, layout_header.xml (header navigation drawer), layout_toolbar.xml (toolbar for ActionBar).

activity_main.xml

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <android.support.v4.widget.DrawerLayout
2.     xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto"
4.     xmlns:tools="http://schemas.android.com/tools"
5.     android:id="@+id/drawer"
6.     android:layout_width="match_parent"
7.     android:layout_height="match_parent"
8.     android:fitsSystemWindows="true"
9.     tools:context=".MainActivity">
10.
11.     <LinearLayout
12.         android:layout_width="match_parent"
13.         android:layout_height="match_parent"
14.         android:measureWithLargestChild="false"
15.         android:orientation="vertical">
16.
17.         <include
18.             android:id="@+id/toolbar"
19.             layout="@layout/layout_toolbar" />
20.         <TextView
21.             android:layout_width="wrap_content"
22.             android:layout_height="wrap_content"
23.             android:textAppearance="?android:attr/textAppearanceLarge"
24.             android:text="Ini adalah Contoh isi konten dengan widget TextView yang
25.             dibungkus oleh Navigation Drawer"
26.             android:id="@+id/textView" android:layout_gravity="center_horizontal"/>
27.         </LinearLayout>
28.         <android.support.design.widget.NavigationView
29.             android:id="@+id/navigation_view"
30.             android:layout_width="wrap_content"
31.             android:layout_height="match_parent"
32.             android:layout_gravity="start"
33.             app:headerLayout="@layout/layout_header"
34.             app:menu="@menu/item_navigasi" />
35.     </android.support.v4.widget.DrawerLayout>
```

layout_header.xml

```

DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="190dp"
5.     android:background="#2ecc71"
6.     android:orientation="vertical">
7.     <ImageView
8.         android:layout_width="wrap_content"
9.         android:layout_height="wrap_content"
10.        android:layout_gravity="bottom"
11.        android:layout_marginLeft="100dp"
12.        android:padding="30dp"
13.        android:src="@drawable/icon_profile"
14.        android:layout_alignBottom="@+id/linearLayout"
15.        android:layout_alignParentRight="true"
16.        android:layout_alignParentEnd="true"/>
17.     <LinearLayout
18.         android:layout_width="wrap_content"
19.         android:layout_height="wrap_content"
20.         android:layout_gravity="bottom"
21.         android:orientation="vertical" android:layout_alignParentBottom="true"
22.         android:layout_alignParentLeft="true" android:layout_alignParentStart="true"
23.         android:id="@+id/linearLayout">
24.         <TextView
25.             android:id="@+id/name"
26.             android:layout_width="match_parent"
27.             android:layout_height="wrap_content"
28.             android:layout_marginBottom="5dp"
29.             android:layout_marginLeft="16dp"
30.             android:text="Okedroid.com"
31.             android:textAppearance="@style/TextAppearance.AppCompat.Body1"
32.             android:textStyle="bold" />
33.         <TextView
34.             android:layout_width="match_parent"
35.             android:layout_height="wrap_content"
36.             android:layout_marginBottom="16dp"
37.             android:layout_marginLeft="16dp"
38.             android:text="Blog Android and Teknologi"
39.             android:textColor="#fff" />
40.     </LinearLayout>
41. </RelativeLayout>
    
```

layout_toolbar.xml

```

DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <?xml version="1.0" encoding="utf-8"?>
2. <android.support.v7.widget.Toolbar
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:layout_width="match_parent"
5.     android:layout_height="wrap_content"
6.     android:background="@color/colorPrimary"
7.     android:theme="@style/ThemeOverlay.AppCompat">
8. </android.support.v7.widget.Toolbar>
    
```


6) Activity

In the MainActivity.java file you can apply the instruction line below:

MainActivity.java

```

DATA HOSTED WITH ▼ BY PASTESIH.COM - DOWNLOAD RAW - SEE ORIGINAL
1. package com.okedroid.contohnavigationdrawer;
2.
3. import android.os.Bundle;
4. import android.support.design.widget.NavigationView;
5. import android.support.v4.widget.DrawerLayout;
6. import android.support.v7.app.ActionBarDrawerToggle;
7. import android.support.v7.app.AppCompatActivity;
8. import android.support.v7.widget.Toolbar;
9. import android.view.MenuItem;
10. import android.view.View;
11. import android.widget.Toast;
12. public class MainActivity extends AppCompatActivity {
13.     //Mendefinisikan variabel
14.     private Toolbar toolbar;
15.     private NavigationView navigationView;
16.     private DrawerLayout drawerLayout;
17.     @Override
18.     protected void onCreate(Bundle savedInstanceState) {
19.         super.onCreate(savedInstanceState);
20.         setContentView(R.layout.activity_main);
21.         // Menginisiasi Toolbar dan mensetting sebagai actionbar
22.         toolbar = (Toolbar) findViewById(R.id.toolbar);
23.         setSupportActionBar(toolbar);
24.         // Menginisiasi NavigationView
25.         navigationView = (NavigationView) findViewById(R.id.navigation_view);
26.         //Mengatur Navigasi View Item yang akan dipanggil untuk menangani item klik
        menu navigasi
27.         navigationView.setNavigationItemSelectedListener(new
        NavigationView.OnNavigationItemSelectedListener() {
28.             // This method will trigger on item click of navigation menu
29.             @Override
30.             public boolean onNavigationItemSelectedListener(MenuItem menuItem) {
31.                 //Memeriksa apakah item tersebut dalam keadaan dicek atau tidak,
32.                 if(menuItem.isChecked()) menuItem.setChecked(false);
33.                 else menuItem.setChecked(true);
34.                 //Menutup drawer item klik
35.                 drawerLayout.closeDrawers();
36.                 //Memeriksa untuk melihat item yang akan diklik dan melakukan aksi
37.                 switch (menuItem.getItemId()){
38.                     // pilihan menu item navigasi akan menampilkan pesan toast klik
                    kalian bisa menggantinya
39.                     //dengan intent activity
40.                     case R.id.navigation1:
41.                         Toast.makeText(getApplicationContext(), "Beranda Telah
                    Dipilih", Toast.LENGTH_SHORT).show();
42.                         return true;
43.                     case R.id.navigation2:
44.                         Toast.makeText(getApplicationContext(), "Profil Telah
                    Dipilih", Toast.LENGTH_SHORT).show();
45.                         return true;
                }
            }
        });
    }
}

```

```

46.         case R.id.navigation3:
47.             Toast.makeText(getApplicationContext(),"Daftar Telah
Dipilih",Toast.LENGTH_SHORT).show();
48.             return true;
49.         case R.id.navigation4:
50.             Toast.makeText(getApplicationContext(),"Setting telah
dipilih",Toast.LENGTH_SHORT).show();
51.             return true;
52.         case R.id.navigation5:
53.             Toast.makeText(getApplicationContext(),"About telah
dipilih",Toast.LENGTH_SHORT).show();
54.             return true;
55.         default:
56.             Toast.makeText(getApplicationContext(),"Kesalahan Terjadi
",Toast.LENGTH_SHORT).show();
57.             return true;
58.     }
59. }
60. });
61. // Menginisiasi Drawer Layout dan ActionBarToggle
62. drawerLayout = (DrawerLayout) findViewById(R.id.drawer);
63. ActionBarDrawerToggle actionBarDrawerToggle = new
ActionBarDrawerToggle(this,drawerLayout,toolbar,R.string.openDrawer,
R.string.closeDrawer){
64.     @Override
65.     public void onDrawerClosed(View drawerView) {
66.         // Kode di sini akan merespons setelah drawer menutup disini kita
biarkan kosong
67.         super.onDrawerClosed(drawerView);
68.     }
69.     @Override
70.     public void onDrawerOpened(View drawerView) {
71.         // Kode di sini akan merespons setelah drawer terbuka disini kita
biarkan kosong
72.         super.onDrawerOpened(drawerView);
73.     }
74. };
75. //mensetting actionBarToggle untuk drawer Layout
76. drawerLayout.setDrawerListener(actionBarDrawerToggle);
77. //menanggil synstate
78. actionBarDrawerToggle.syncState();
79. }
80.
81.
82. }

```

7) Run project

After finished all we try to run the application via Android Studio. The result will look more or less like this.

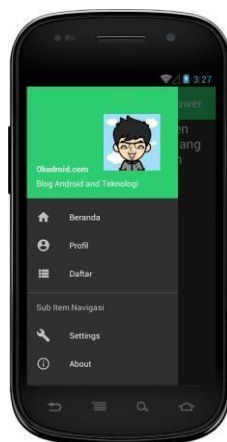


Figure 3.13Result Display Navigation Drawer

6. TAB MENU

The tab menu is often used to display menus in applications. Many applications use tabbed menus such as the built-in android media player.

- 1) Create a new project named MenuTab
- 2) Then open the folder res => layout => activity_main.xml. Here are 3 components: TabHost, TabWidget and FrameLayout. TabWidget is used to display tab menus that we create. And FrameLayout is used to display the contents of the tab menu. Here's the full code.

```
<TabHost xmlns:android="http://schemas.android.com/apk/res
    android:id="@android:id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom" />

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_gravity="bottom" />

    </LinearLayout>
</TabHost>
```

- 3) Next go into the src -> MainActivity.java folder to create tab menus. MainActivity under using TabActivity extends which means that the class contained in TabActivity can be downgraded to MainActivity.

```
package com.example.menutab;

import android.os.Bundle;
import android.app.TabActivity;
import android.content.Intent;
import android.widget.TabHost;

@SuppressWarnings("deprecation")
public class MainActivity extends TabActivity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TabHost tabhost = getTabHost();
        TabHost.TabSpec spec;
        Intent intent;

        intent = new Intent().setClass(this, BerandaActivi
        spec = tabhost.newTabSpec("beranda").setIndicator(
        tabhost.addTab(spec); //untuk membuat tabbaru disin

        intent = new Intent().setClass(this, BeritaActivit
        spec = tabhost.newTabSpec("berita").setIndicator("
        tabhost.addTab(spec);

        intent = new Intent().setClass(this, TemanActivity
        spec = tabhost.newTabSpec("teman").setIndicator("T
        tabhost.addTab(spec);

    }
}
```

- 4) Then we set up the layout to display in the tab menu.
- 5) Source home.xml. This layout is used to display the home menu.

```
<?xml version="1.0" encoding="utf-8"?>
<DigitalClock xmlns:android="http://schemas.android.com/apk
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:gravity="center_vertical|center"
  android:textSize="50sp" >

</DigitalClock>
```

- 6) Source news.xml. This layout is used to display news menu.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/ap
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  >
  <ListView
    android:id="@android:id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
  />
</LinearLayout>
```

- 7) Source teman.xml. This layout is used to display the friend menu.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/ap
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  >
  <ListView
    android:id="@android:id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
  />
</LinearLayout>
```

- 8) Next create a new class to call the layout "we have created.
- 9) BerandaActivity.java source code.

```
package com.example.menutab;

import android.app.Activity;
import android.os.Bundle;

public class BerandaActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.beranda);
    }
}
```

10) BeritaActivity.java source code.

```
package com.example.menutab;

import android.app.ListActivity;
import android.os.Bundle;
import android.widget.AdapterView;

public class BeritaActivity extends ListActivity {
    String [] berita ={"Jadwal Piala Dunia 2014", "Capres
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.berita);

        setListAdapter(new ArrayAdapter<String>(this, andr
    }
}
```

11) TemanActivity.java source code.

```
package com.example.menutab;

import android.app.ListActivity;
import android.os.Bundle;
import android.widget.AdapterView;

public class TemanActivity extends ListActivity {
    String [] teman ={"Andra", "Dina", "Edo", "Julia"};
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.teman);

        setListAdapter(new ArrayAdapter<String>(this, andr
    }
}
```

12) Do not forget to register the activity that we created to Androidmanifest.xml code as follows.

```
<activity android:name="BerandaActivity"></activity>
<activity android:name="BeritaActivity"></activity>
<activity android:name="TemanActivity"></activity>
```

Here are the results:



Figure 3.14 Result Display Tab Menu

7. FRAGMENT

Fragment is one component of the interface (user interface) which is a part of Activity, can also be called by the name of Sub-Activity. One Activity can manage multiple fragments. To display results on a user screen. In One Activity too, a fragment can be replaced, added and removed, as needed. Fragment is affected from Activity's lifecycle (lifecycle), because Fragment is part of Activity. Here are the steps to create a fragment:

- 1) **Create a new project**
- 2) **Create a new activity**

Here will need 2 pieces of java files and 2 pieces of file layout, for it created a new Activity first. For example I will create an activity file named Fragment1.java, Fragment2.java and file layout fragment_satu.xml, fragment_dua.xml.

- 3) **Applying activity source code**

Layout

activity_main.xml

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     android:layout_width="fill_parent"
3.     android:layout_height="fill_parent" >
4.
5.     <fragment
6.         android:id="@+id/fragment2"
7.         android:name="com.okedroid.aplikasisaya.Fragment2"
8.         android:layout_width="0px"
9.         android:layout_height="match_parent"
10.        android:layout_weight="1"
11.    />
12.
13.    <fragment
14.        android:id="@+id/fragment1"
15.        android:name="com.okedroid.aplikasisaya.Fragment1"
16.        android:layout_width="0px"
17.        android:layout_height="match_parent"
18.        android:layout_weight="1"
19.    />
20.
21. </LinearLayout>
```

fragment_satu.xml

```
DATA HOSTED WITH ▼ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical"
6.     android:background="#3498db"
7. >
8.
9.     <TextView
10.        android:id="@+id/textView1"
11.        android:layout_width="wrap_content"
12.        android:layout_height="wrap_content"
13.        android:text="Fragment Pertama"
14.        android:textAppearance="?android:attr/textAppearanceLarge" />
15.
16. </LinearLayout>
```

fragment_dua.xml

```
DATA HOSTED WITH ▼ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical"
6.     android:background="#2980b9"
7. >
8.
9.     <TextView
10.        android:id="@+id/textView1"
11.        android:layout_width="wrap_content"
12.        android:layout_height="wrap_content"
13.        android:text="Fragment Kedua"
14.        android:textAppearance="?android:attr/textAppearanceLarge" />
15.
16. </LinearLayout>
```

Java

MainActivity.java

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL

1. package com.okedroid.aplikasisaya;
2.
3. import android.os.Bundle;
4. import android.support.v7.app.AppCompatActivity;
5.
6. public class MainActivity extends AppCompatActivity {
7.
8.
9.
10.     // Button variables
11.
12.
13.     @Override
14.     protected void onCreate(Bundle savedInstanceState) {
15.         super.onCreate(savedInstanceState);
16.         setContentView(R.layout.activity_main);
17.
18.     }
19.
20.
21. }
```

Fragment1.java

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL

1. package com.okedroid.aplikasisaya;
2.
3.
4. import android.os.Bundle;
5. import android.app.Fragment;
6. import android.view.LayoutInflater;
7. import android.view.View;
8. import android.view.ViewGroup;
9.
10. public class Fragment1 extends Fragment {
11.     @Override
12.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
13.                             Bundle savedInstanceState) {
14.         // TODO Auto-generated method stub
15.         return inflater.inflate(R.layout.fragment_satu, container, false);
16.     }
17.
18. }
```


Fragment2.java

```
DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL  
1. package com.okedroid.aplikasisaya;  
2.  
3.  
4. import android.os.Bundle;  
5. import android.app.Fragment;  
6. import android.view.LayoutInflater;  
7. import android.view.View;  
8. import android.view.ViewGroup;  
9.  
10. public class Fragment2 extends Fragment {  
11.     @Override  
12.     public View onCreateView(LayoutInflater inflater, ViewGroup container,  
13.                             Bundle savedInstanceState) {  
14.         // TODO Auto-generated method stub  
15.         return inflater.inflate(R.layout.fragment_dua,container, false);  
16.     }  
17.  
18. }
```

4) Run the application

After successfully applied we try to run the application via Android Studio. The result would be something like this:

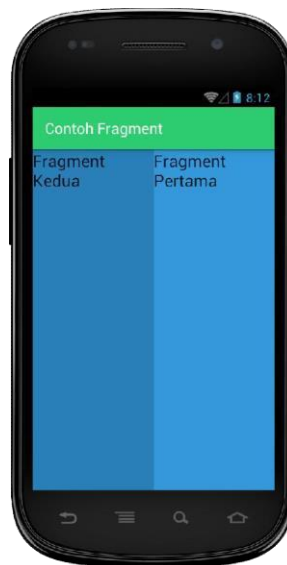


Figure 3.15 Result Display Fragment

MODUL 4 MAPS

Google Maps is a google navigation map that helps the user search, view and find the location of the place where you want to go and where you want to go. Google Maps is not only available on the Web, but it is also available in mobile applications especially Android. Well Google provides an API used for Android developers, developing their Navigation Maps app called Google Maps API.

1. CURRENT LOCATION

This section uses one of the features of the Google Maps API, which displays the location area around the user, which is marked with a marker on the Google Maps screen. Here are the steps to display the current location.

1) Manifest

First on your Android Studio project in the manifest folder> AndroidManifest.xml. Then add the following user permissions to the *AndroidManifest.xml* file.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

```
DATA HOSTED WITH ▼ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.     package="com.okedroid.googlemaps">
4.
5.     <!--
6.         The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
7.         Google Maps Android API v2, but you must specify either coarse or fine
8.         location permissions for the 'MyLocation' functionality.
9.     -->
10.    <uses-permission android:name="android.permission.INTERNET" />
11.    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
12.    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
13.
14.    <application
15.        android:allowBackup="true"
16.        android:icon="@mipmap/ic_launcher"
17.        android:label="@string/app_name"
18.        android:supportRtl="true"
19.        android:theme="@style/AppTheme">
20.
21.        <!--
22.            The API key for Google Maps-based APIs is defined as a string resource.
23.            (See the file "res/values/google_maps_api.xml").
24.            Note that the API key is linked to the encryption key used to sign the
25.            APK.
26.            You need a different API key for each encryption key, including the
27.            release key that is used to
28.            sign the APK for publishing.
29.            You can define the keys for the debug and release targets in src/debug/
30.            and src/release/.
31.        -->
32.        <meta-data
33.            android:name="com.google.android.geo.API_KEY"
34.            android:value="@string/google_maps_key"/>
35.
36.        <activity
37.            android:name=".MapsActivity"
38.            android:label="@string/title_activity_maps">
39.            <intent-filter>
40.                <action android:name="android.intent.action.MAIN">
41.
42.                <category android:name="android.intent.category.LAUNCHER"/>
43.            </intent-filter>
44.        </activity>
45.    </application>
```

2) Layout

After that in the file layout *activity_maps.xml*. Make sure the instruction line is like this:

activity_maps.xml

```

DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. <?xml version="1.0" encoding="utf-8"?><!--
2.     Copyright (C) 2012 The Android Open Source Project
3.     Licensed under the Apache License, Version 2.0 (the "License");
4.     you may not use this file except in compliance with the License.
5.     You may obtain a copy of the License at
6.         http://www.apache.org/licenses/LICENSE-2.0
7.     Unless required by applicable law or agreed to in writing, software
8.     distributed under the License is distributed on an "AS IS" BASIS,
9.     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
10.    See the License for the specific language governing permissions and
11.    limitations under the License.
12. -->
13. <!-- This can go anywhere in your layout (see other demos for some examples). -->
14. <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
15.             android:layout_width="match_parent"
16.             android:layout_height="match_parent"
17.             android:id="@+id/layout">
18.
19.     <fragment
20.         android:id="@+id/map"
21.         class="com.google.android.gms.maps.SupportMapFragment"
22.         xmlns:android="http://schemas.android.com/apk/res/android"
23.         android:layout_width="match_parent"
24.         android:layout_height="match_parent" />
25. </FrameLayout>
    
```

3) Activity

In the MainActivity.java java file, copy the instruction line below:

MainActivity.java

```

DATA HOSTED WITH ♥ BY PASTEBIN.COM - DOWNLOAD RAW - SEE ORIGINAL
1. package com.okedroid.googlemaps;
2.
3. import android.Manifest;
4. import android.content.Context;
5. import android.content.pm.PackageManager;
6. import android.location.Criteria;
7. import android.location.Location;
8. import android.location.LocationListener;
9. import android.location.LocationManager;
10. import android.location.LocationProvider;
11. import android.support.v4.app.FragmentActivity;
12. import android.os.Bundle;
13. import android.util.Log;
14. import com.google.android.gms.maps.CameraUpdateFactory;
15. import com.google.android.gms.maps.GoogleMap;
16. import com.google.android.gms.maps.OnMapReadyCallback;
17. import com.google.android.gms.maps.SupportMapFragment;
18. import com.google.android.gms.maps.model.BitmapDescriptorFactory;
19. import com.google.android.gms.maps.model.CameraPosition;
20. import com.google.android.gms.maps.model.CircleOptions;
21. import com.google.android.gms.maps.model.LatLng;
22. import com.google.android.gms.maps.model.Marker;
23. import com.google.android.gms.maps.model.MarkerOptions;
    
```

```

25. public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
    LocationListener {
26.
27.     private GoogleMap mMap;
28.     private LocationManager locationManager = null;
29.     private String provider = null;
30.     private Marker mCurrentPosition = null;
31.
32.     @Override
33.     protected void onCreate(Bundle savedInstanceState) {
34.         super.onCreate(savedInstanceState);
35.         setContentView(R.layout.activity_maps);
36.         // Obtain the SupportMapFragment and get notified when the map is ready to
    be used.
37.         SupportMapFragment mapFragment = (SupportMapFragment)
    getSupportFragmentManager()
38.             .findFragmentById(R.id.map);
39.         mapFragment.getMapAsync(this);
40.     }
41.
42.     @Override
43.     public void onMapReady(GoogleMap googleMap) {
44.         mMap = googleMap;
45.         mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
46.
47.         if (isProviderAvailable() && (provider != null)) {
48.             locateCurrentPosition();
49.         }
50.     }
51.
52.     private void locateCurrentPosition() {
53.
54.         int status =
55.             getPackageManager().checkPermission(Manifest.permission.ACCESS_COARSE_LOCATION,
56.                 getPackageName());
57.
58.         if (status == PackageManager.PERMISSION_GRANTED) {
59.             Location location = locationManager.getLastKnownLocation(provider);
60.             updateWithNewLocation(location);
61.             // locationManager.addGpsStatusListener(this);
62.             long minTime = 5000; // ms
63.             float minDist = 5.0f; // meter
64.             locationManager.requestLocationUpdates(provider, minTime, minDist,
65.                 this);
66.         }
    }

```

```

69. private boolean isProviderAvailable() {
70.     mLocationManager = (LocationManager) getSystemService(
71.         Context.LOCATION_SERVICE);
72.     Criteria criteria = new Criteria();
73.     criteria.setAccuracy(Criteria.ACCURACY_COARSE);
74.     criteria.setAltitudeRequired(false);
75.     criteria.setBearingRequired(false);
76.     criteria.setCostAllowed(true);
77.     criteria.setPowerRequirement(Criteria.POWER_LOW);
78.
79.     provider = mLocationManager.getBestProvider(criteria, true);
80.     if (mLocationManager
81.         .isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
82.         provider = LocationManager.NETWORK_PROVIDER;
83.
84.         return true;
85.     }
86.
87.     if (mLocationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
88.         provider = LocationManager.GPS_PROVIDER;
89.         return true;
90.     }
91.
92.     if (provider != null) {
93.         return true;
94.     }
95.     return false;
96. }
97.
98. private void updateWithNewLocation(Location location) {
99.
100.     if (location != null && provider != null) {
101.         double lng = location.getLongitude();
102.         double lat = location.getLatitude();
103.
104.         addBoundaryToCurrentPosition(lat, lng);
105.
106.         CameraPosition camPosition = new CameraPosition.Builder()
107.             .target(new LatLng(lat, lng)).zoom(10f).build();
108.
109.         if (mMap != null)
110.             mMap.animateCamera(CameraUpdateFactory
111.                 .newCameraPosition(camPosition));
112.     } else {
113.         Log.d("Location error", "Something went wrong");
114.     }
115. }
116. }
    
```

```

119. private void addBoundaryToCurrentPosition(double lat, double lang) {
120.
121.     MarkerOptions mMarkerOptions = new MarkerOptions();
122.     mMarkerOptions.position(new LatLng(lat, lang));
123.     mMarkerOptions.icon(BitmapDescriptorFactory
124.         .fromResource(R.drawable.marker));
125.     mMarkerOptions.anchor(0.5f, 0.5f);
126.
127.     CircleOptions mOptions = new CircleOptions()
128.         .center(new LatLng(lat, lang)).radius(10000)
129.         .strokeColor(0x110000FF).strokeWidth(1).fillColor(0x110000FF);
130.     mMap.addCircle(mOptions);
131.     if (mCurrentPosition != null)
132.         mCurrentPosition.remove();
133.     mCurrentPosition = mMap.addMarker(mMarkerOptions);
134. }
135.
136.
137. @Override
138. public void onLocationChanged(Location location) {
139.
140.     updateWithNewLocation(location);
141. }
142.
143. @Override
144. public void onProviderDisabled(String provider) {
145.
146.     updateWithNewLocation(null);
147. }
148.
149. @Override
150. public void onProviderEnabled(String provider) {
151.
152. }
153.
154. @Override
155. public void onStatusChanged(String provider, int status, Bundle extras) {
156.     switch (status) {
157.         case LocationProvider.OUT_OF_SERVICE:
158.             break;
159.         case LocationProvider.TEMPORARILY_UNAVAILABLE:
160.             break;
161.         case LocationProvider.AVAILABLE:
162.             break;
163.     }
164. }
165. }
    
```

In the `.fromResource (R.drawable.marker)` section); if there is error then make sure you have set up marker then save in resource location> drawable> marker.png

4) Run the application

After all is done, let's try to test the Application. By running the Application in Android Studio. The result will look something like this:

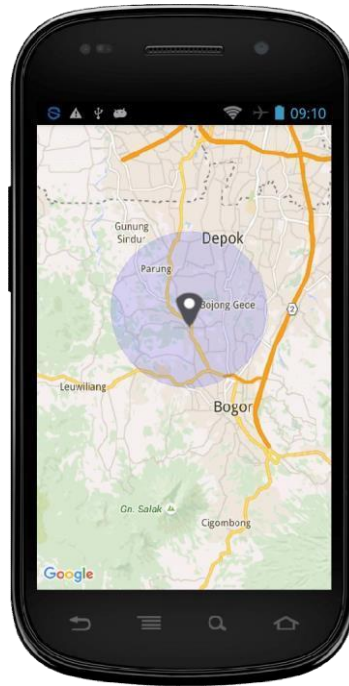


Figure 4.1 Result Display Maps

2. PLACE PICKER

First of all, make sure Google Places API for Android is already enabled on your Developer Console project. The way to go to this page and then go to API> Google Places for Android and then click "Enable API". After that create a new project on Android Studio.

After that go into your build.gradle file and add dependencies as follows:

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support.appcompat-v7:25.1.1'
    compile 'com.google.android.gms:play-services-places:10.0.1'
}
```

After that, we start by making the layout first. This main layout is called *activity_main.xml* and contains only one key and one TextView.

activity_main.xml

```
<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"
    tools:context="id.web.twoh.placesapitutorial.MainActivity">

    <Button
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/bt_ppicker"
        android:text="Launch Place Picker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:layout_below="@id/bt_ppicker"
        android:id="@+id/tv_place_id"
        android:text="Place name..."
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Then we will create the main code in the main activity, open the MainActivity.java class, or other classes that become main activity on your Android Studio project. Then copy-paste the following code:

MainActivity.java

```

package id.web.twoh.placesapitutorial;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.GooglePlayServicesNotAvailableException;
import com.google.android.gms.common.GooglePlayServicesRepairableException;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.ui.PlacePicker;

public class MainActivity extends AppCompatActivity {

    private Button btPlacesAPI;
    private TextView tvPlaceAPI;
    // konstanta untuk mendeteksi hasil balikan dari place picker
    private int PLACE_PICKER_REQUEST = 1;

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

        tvPlaceAPI = (TextView) findViewById(R.id.tv_place_id);

        btPlacesAPI = (Button) findViewById(R.id.bt_ppicker);
        btPlacesAPI.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // membuat Intent untuk Place Picker
                PlacePicker.IntentBuilder builder = new PlacePicker.IntentBuilder();
                try {
                    //menjalankan place picker
                    startActivityForResult(builder.build(MainActivity.this), PLACE_PICKER_

                    // check apabila <a title="Solusi Tidak Bisa Download Google Play Services
                } catch (GooglePlayServicesRepairableException e) {
                    e.printStackTrace();
                } catch (GooglePlayServicesNotAvailableException e) {
                    e.printStackTrace();
                }
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        // menangkap hasil balikan dari Place Picker, dan menampilkannya pada TextView
        if (requestCode == PLACE_PICKER_REQUEST) {
            if (resultCode == RESULT_OK) {
                Place place = PlacePicker.getPlace(data, this);
                String toastMsg = String.format(
                    "Place: %s \n" +
                    "Alamat: %s \n" +
                    "Latlng %s \n", place.getName(), place.getAddress(), place.getLatL

                tvPlaceAPI.setText(toastMsg);
            }
        }
    }
}

```

Then the last one is to add permissions-permissions required on Android Manifest. The required permissions are more or less the same as when we want to display Maps on Android, such as API Key, GMS Version, Internet permissions, location and so on. How, on Android Studio, open AndroidManifest.xml and add the permissions or the missing sections in your Android Manifest according to the following code:

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

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVI
    <!--
    The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
    Google Maps Android API v2, but are recommended.
    -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="@string/google_maps_key" />
        <activity
            android:name=".MainActivity"
            android:label="TWOHsPlacesAPI" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

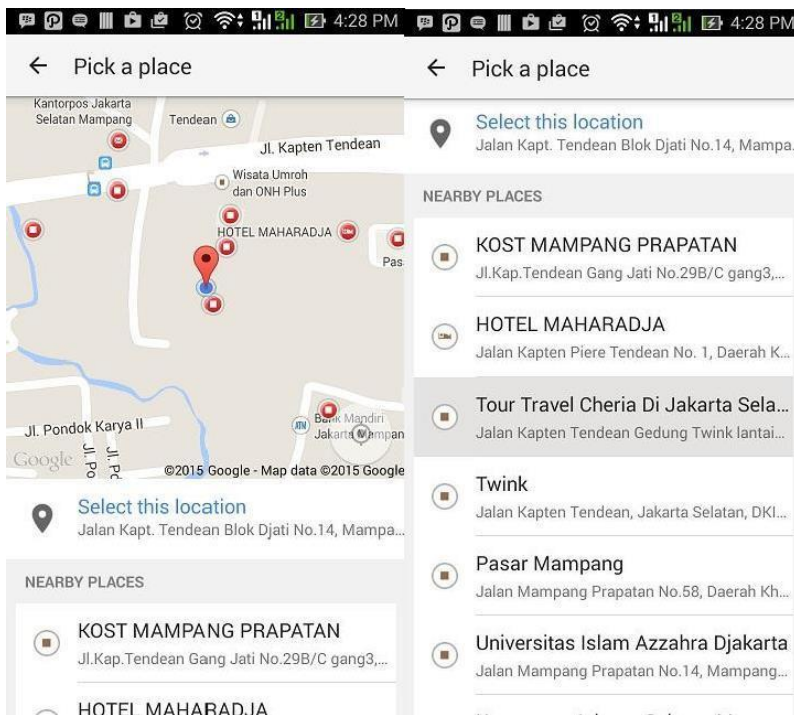


Figure 4.2 Result Display Place Picker

3. SEARCH ADDRESS



Figure 4.3 Result Display Search Address

Open your activity_maps.xml file and a LinearLayout as root layout and add EditText and Button inside LinearLayout. EditText is used to type location name to search in map, Button is for search location after typing something in address bar. Give an id to edit text and button and add onClick attribute in button. Following is the complete content of maps XML layout file.

res/layout/activity_maps.xml

```

1  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      android:layout_width="match_parent"
3      android:layout_height="wrap_content"
4      android:orientation="vertical">
5
6      <LinearLayout
7          android:layout_width="match_parent"
8          android:layout_height="wrap_content"
9          android:layout_margin="5dp"
10
11         android:orientation="horizontal">
12
13         <EditText
14             android:id="@+id/editText"
15             android:layout_width="wrap_content"
16             android:layout_height="wrap_content"
17             android:layout_weight="4"
18             android:hint="Search Location Here" />
19
20         <Button
21             android:id="@+id/search_button"
22             android:layout_width="wrap_content"
23             android:layout_height="wrap_content"
24             android:layout_weight="0.5"
25             android:onClick="onMapSearch"
26             android:text="Search" />
27
28     </LinearLayout>
29
30     <fragment xmlns:android="http://schemas.android.com/apk/res/android"
31         xmlns:tools="http://schemas.android.com/tools"
32         android:id="@+id/map"
33         android:name="com.google.android.gms.maps.SupportMapFragment"
34         android:layout_width="match_parent"
35         android:layout_height="match_parent"
36         tools:context="com.viralandroid.googlemapsandroidapi.MainActivity" />
37
38 </LinearLayout>

```

Now open your java activity file MapsActivity.java and add following lines of code. The complete java code of MapsActivity.java file will look like below.

MainActivity.java

```
package com.viralandroid.googlemapsandroidapi;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.view.View;
import android.widget.EditText;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
import java.util.List;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    public void onMapSearch(View view) {
        EditText locationSearch = (EditText) findViewById(R.id.editText);
        String location = locationSearch.getText().toString();
        List<Address> addressList = null;

        if (location != null || !location.equals("")) {
            Geocoder geocoder = new Geocoder(this);
            try {
                addressList = geocoder.getFromLocationName(location, 1);
            } catch (IOException e) {
                e.printStackTrace();
            }
            Address address = addressList.get(0);
            LatLng latLng = new LatLng(address.getLatitude(), address.getLongitude());
            mMap.addMarker(new MarkerOptions().position(latLng).title("Marker"));
            mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(27.746974, 85.361582);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Kathmandu, Nepal"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        mMap.setMyLocationEnabled(true);
    }
}
```

MODUL 5 DATABASE

Purpose:

Understand how the integration android and database

Material:

Create the database SQLite and do crud process (Insert, Update, delete, and Read)

Database. The use of databases in android is very important in the development android applications. The database used in this material is a MySQL database CRUD, CRUD is insert, update, delete, perform.

➤ The Advantages Of SQLite:

- Do not require third parties to access the data
- Because these databases are portable, then the application can directly coupled with the applications that are often referred to as an embed
- If you want to copy the database, simply copy the file only
- is suitable in applications which are not connected to the internet applications, both desktop or mobile applications

➤ SQLite Deficiencies:

- Because these applications directly connected on the application without using intermediaries, so that SQLite is used to store data that is a little data or temporary, for example game the only store information score the winner.
- SQLite does not use the User Management, meaning that if get the file database, could be opened without using a username or password
- not all query command can be done on this SQLite
- Example a programmer wants to make a game application the same but on different platforms, such as android, iphone and blackBerry. Then the programmer had to create a database on the each of these platforms

➤ Advantages Of MySql:

- Database has better security, because access should use the username and password
- Has a complete query command
- If you create many applications with different platforms, simply access only one database
- can store data that overwhelmingly and accessed quickly (depending on the query that is used)

➤ Lack Of Mysql:

- Installation more complicated than SQLite
- must have a Server that connects databases with applications
- If you use Android, android should then connect to the server

(usually using the internet). Due to access of
Android to MySql through PHP

1. Getting started

To make it easier to see the results of further work, please install

SQLite Manager on Android Studio:

1. [Download SQLite Manager_1.0.0.jar](#)

2. put the SQLite Manager in dopins folder (the location of the
/Android/Sdk/tools/lib/monitor-x86_64/plugins/ sqlitemanager_1.0.0.jar)

3. on Android in the Android Studio, Android Device Monitor open Windows -> ShowView -> Other ->
File Explorer

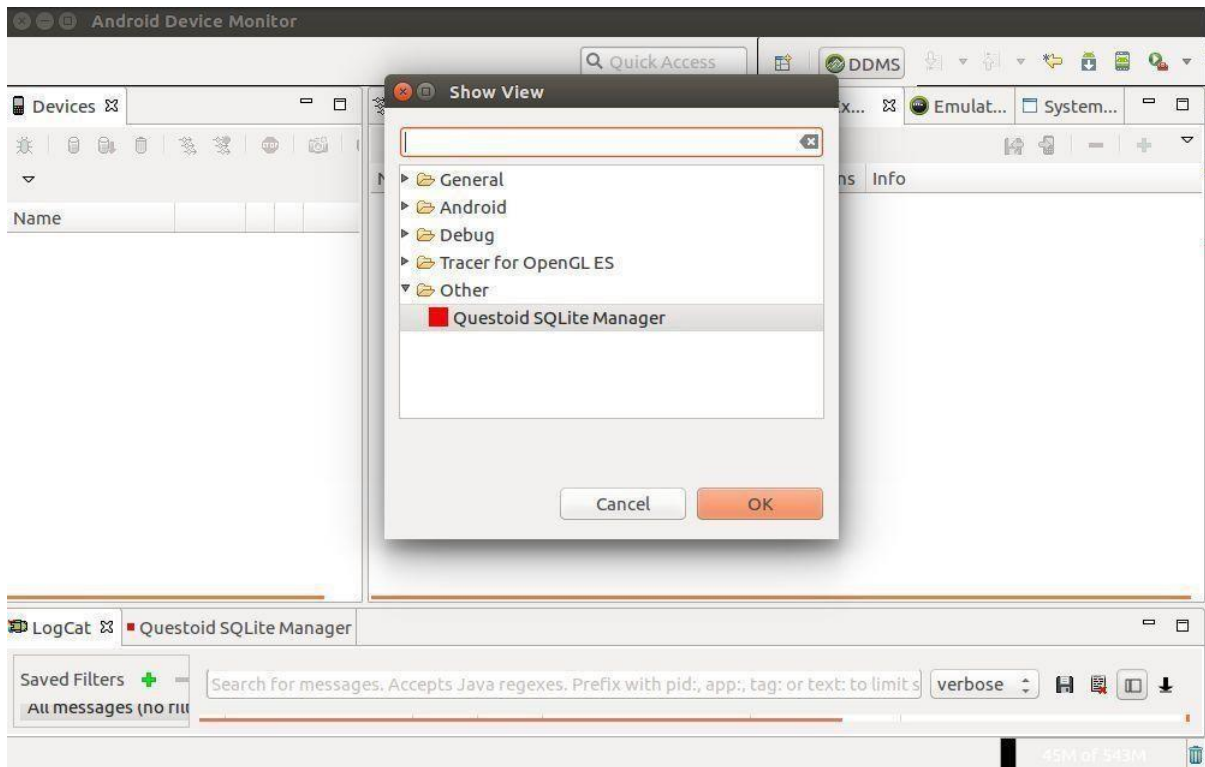


Figure 5.1 Getting started

If emulator has not been executed, the File Explorer will be empty, make sure the emulator already running if you want to see the display as above.

2. creating a Project CRUD_SQLITE

1. open the Android's Studio, and then file->New->Other->Android->Android Application Project
2. Fill in each Field (for the other field may be made According to the wishes of each):

Application Name : **CRUD_SQLITE**

Package Name : **nama_nim.com**

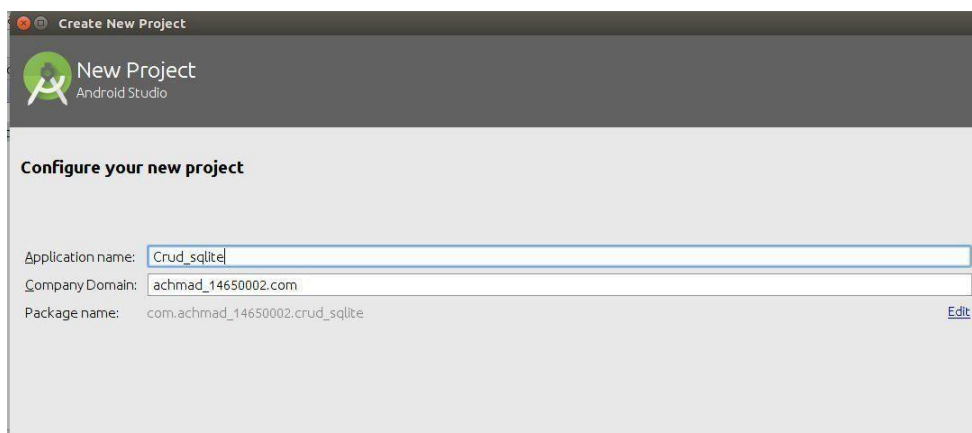


Figure 5.2 Create New Project

3. Create a database_mahasiswa.db, tabel_biodata and the trials enteringData

To perform this test, the data used are not using the userinterface, so that the data entered is still using the data entered manual.

1. create a file with the name **SQLiteHelper.java**

2. in the **SQLiteHelper.java**, enter the following source code

```
import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class SQLiteHelper extends SQLiteOpenHelper{
    private static final String nama_database = "database_mahasiswa.db";
    private static final int versi_database = 1;
    private static final String query_buat_tabel_biodata_penain = "CREATE TABLE IF NOT EXISTS tabel_biodata(id_biodata INTEGER PRIMARY KEY AUTOINCREMENT, nama TEXT, alamat TEXT)";
    private static final String query_hapus_tabel_biodata_penain = "DROP TABLE IF EXISTS query_buat_tabel_biodata_penain";

    public SQLiteHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, nama_database, null, versi_database);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL(query_buat_tabel_biodata_penain);
        System.out.println("tabel_biodata sudah dibuat");
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int versi_lama, int versi_baru) {
        sqLiteDatabase.execSQL(query_hapus_tabel_biodata_penain);
        onCreate(sqLiteDatabase);
    }

    public void tambah_biodata(String nama, String alamat) {
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("nama", nama);
        values.put("alamat", alamat);
        database.insert("tabel_biodata", null, values);
        database.close();
    }
}
```

If executed, the result can be found in the File

Explorer->data->data->nama_nim.com->databases->database_pemain.db

4. Method of displaying Data

To display the data, add the method tampil_semua_biodata () on a file

```
public ArrayList<HashMap<String, String>> tampil_semua_biodata() {
    // deklarasikan sebuah arraylist yang bisa menampung hashmap
    ArrayList<HashMap<String, String>> arrayListBiodata = new ArrayList<HashMap<String, String>>();
    SQLiteDatabase database = this.getWritableDatabase();
    Cursor cursor = database.rawQuery("SELECT * FROM tabel_biodata", null);
    // kursor langsung diarkan ke posisi paling awal data pada tabel biodata
    if (cursor.moveToFirst()) {
        do {
            // deklarasikan sebuah hashmap, yang bisa menamp
            HashMap<String, String> hashMapBiodata = new HashMap<String, String>();
            // masukkan masing-masing field dari tabel biodata ke dalam hashMapBiodata
            //pastikan id_biodata, nama dan alamat sama persis dengan field yang ada pada tabel biodata
            hashMapBiodata.put("id_biodata", cursor.getString(0));
            hashMapBiodata.put("nama", cursor.getString(1));
            hashMapBiodata.put("alamat", cursor.getString(2));
            // masukkan hashMapBiodata ke dalam arrayListBiodata
            arrayListBiodata.add(hashMapBiodata);
        } while (cursor.moveToNext());
    }
    return arrayListBiodata;
}
```


5. Method Update Data

Add the method `update_biodata ()` on the `SQLiteHelper.java`

```
public int update_biodata(int id, String nama, String alamat) {  
    SQLiteDatabase database = this.getWritableDatabase();  
    ContentValues recordBiodata = new ContentValues();  
    recordBiodata.put("nama", nama);  
    recordBiodata.put("alamat", alamat);  
    return database.update("tabel_biodata", recordBiodata, "id_biodata=" + id, null);  
}
```

6. Method Hapus Data

Add the method `hapus_biodata pada()` on the `SQLiteHelper.java`

```
public void hapus_biodata(int id) {  
    SQLiteDatabase database = this.getWritableDatabase();  
    database.execSQL("DELETE FROM tabel_biodata WHERE id_biodata=" + id + "");  
    database.close();  
}
```

7. Method Fetch data based on ID

At this point the user will send the id and sqlite will send back the value of the to the sender only data based on the specified id

```
public HashMap<String, String> tampil_biodata_berdasarkan_id(int id) {  
    SQLiteDatabase database = this.getReadableDatabase();  
    HashMap<String, String> hashMapBiodata = new HashMap<String, String>();  
    Cursor cursor = database.rawQuery("SELECT * FROM tabel_biodata WHERE id_biodata=" + id + "", null);  
  
    if (cursor.moveToFirst()) {  
        do {  
            hashMapBiodata.put("id_biodata", cursor.getString(0));  
            hashMapBiodata.put("nama", cursor.getString(1));  
            hashMapBiodata.put("alamat", cursor.getString(2));  
        } while (cursor.moveToNext());  
    }  
  
    return hashMapBiodata;  
}
```

8 . CRUD with User Interface main.xml

On the previous stage has not used the look, so for testing still using the final data, order data is statistic processed, the user used user interface.



Figure 5.3 Result Display Crud

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/buttonTambahBiodata"
        android:layout_width="186dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Tambah Biodata" />

    <HorizontalScrollView
        android:id="@+id/horizontalScrollView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <ScrollView
            android:id="@+id/verticalScrollView"
            android:layout_width="wrap_content"
            android:layout_height="match_parent" >

            <TableLayout
                android:id="@+id/tableBiodata"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" >
            </TableLayout>
        </ScrollView>
    </HorizontalScrollView>
</LinearLayout>
```

9. MainActivity.java

MainActivity.java like this

```

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import java.util.ArrayList;
import java.util.HashMap;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Color;
import android.os.Bundle;
import android.support.v4.view.ViewPager.LayoutParams;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements OnClickListener {

    SQLiteHelper sqliteHelper = new SQLiteHelper(this);
    TableLayout tabelBiodata;
    Button buttonTambahBiodata;
    ArrayList<Button> buttonEdit = new ArrayList<>();
    ArrayList<Button> buttonDelete = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tabelBiodata = (TableLayout) findViewById(R.id.tableBiodata);
        buttonTambahBiodata = (Button) findViewById(R.id.buttonTambahBiodata);
        buttonTambahBiodata.setOnClickListener(this);

        TableRow barisTabel = new TableRow(this);
        barisTabel.setBackgroundColor(Color.RED);

        TextView viewHeaderId = new TextView(this);
        TextView viewHeaderNama = new TextView(this);
        TextView viewHeaderAlamat = new TextView(this);
        TextView viewHeaderAction = new TextView(this);

        viewHeaderId.setText("ID");
        viewHeaderNama.setText("Nama");
        viewHeaderAlamat.setText("Alamat");
        viewHeaderAction.setText("Action");

        viewHeaderId.setPadding(5, 1, 5, 1);
        viewHeaderNama.setPadding(5, 1, 5, 1);
        viewHeaderAlamat.setPadding(5, 1, 5, 1);
        viewHeaderAction.setPadding(5, 1, 5, 1);

        barisTabel.addView(viewHeaderId);
        barisTabel.addView(viewHeaderNama);
        barisTabel.addView(viewHeaderAlamat);
        barisTabel.addView(viewHeaderAction);
    }
}

```

```

tabelBiodata.addView(barisLabel, new TableLayout.LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));

ArrayList<HashMap<String, String>, String> array iStBiodata; = sqlLiteHelper.icmpil sexua Diodaia(,

if (arrayListBiodata.size() > 0) {
    for (int i = 0; i < arrayListBiodata.size(); i++) {
        // Mendapatkan masing-masing data dari arraylist biodata
        HashMap<String, String> hashMapRecordBiodata = arrayListBiodata.get(i);

        String nama = hashMapRecordBiodata.get("nama");
        String alamat = hashMapRecordBiodata.get("alamat");
        String id = hashMapRecordBiodata.get("id_biodata");

        System.out.println("Mencetak Data Biodata");
        System.out.println("Nama: " + nama);
        System.out.println("Alamat: " + alamat);
        System.out.println("ID: " + id);

        barisLabel = new TableRow(this);
        barisLabel.setBackgroundColor(Color.LTGRAY);

        TextView viewId = useTextView(this);
        TextView viewNama = useTextView(this);
        TextView viewAlamat = useTextView(this);

        barisLabel.addView(viewId);
        barisLabel.addView(viewNama);
        barisLabel.addView(viewAlamat);

        TextView viewNama = new TextView(this);
        viewNama.setText(nama);
        viewNama.setPadding(5, 10, 5, 10);
        barisLabel.addView(viewNama);

        TextView viewAlamat = new TextView(this);
        viewAlamat.setText(alamat);
        viewAlamat.setPadding(5, 10, 5, 10);
        barisLabel.addView(viewAlamat);

        Button edit = new Button(this);
        edit.setText("Edit");
        edit.setOnClickListener() {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("Edit Biodata");
            builder.setMessage("Mau edit biodata?");
            builder.setPositiveButton("Ya") {
                dialog, whichButton -> {
                    // Mendapatkan data yang akan di edit
                    String id = viewId.getText().toString();
                    String nama = viewNama.getText().toString();
                    String alamat = viewAlamat.getText().toString();

                    // Mengupdate data biodata
                    sqlLiteHelper.updateBiodata(id, nama, alamat);

                    // Menampilkan pesan sukses
                    Toast.makeText(this, "Biodata berhasil di update", Toast.LENGTH_SHORT).show();
                }
            };
            builder.setNegativeButton("Tidak") {
                dialog, whichButton -> {
                    // Menampilkan pesan sukses
                    Toast.makeText(this, "Biodata tidak di update", Toast.LENGTH_SHORT).show();
                }
            };
            builder.create().show();
        };
        barisLabel.addView(edit);
    }
}

```

```

        tabelBiodata.addView(barisTabel, new TableLayout.LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
    }
}

@Override
public void onClick(View view) {

    if (view.getId() == R.id.buttonTambahBiodata) {
        // Toast.makeText(MainActivity.this, "Button Tambah Data",
        // Toast.LENGTH_SHORT).show();
        tambahBiodata();
    } else {
        /*
        * Melakukan pengecekan pada data array, agar sesuai dengan index
        * masing-masing button
        */
        for (int i = 0; i < buttonEdit.size(); i++) {

            /* jika yang diklik adalah button edit */
            if (view.getId() == buttonEdit.get(i).getId()
                && view.getTag().toString().trim().equals("Edit")) {
                // Toast.makeText(MainActivity.this, "Edit : " +
                // buttonEdit.get(i).getId(), Toast.LENGTH_SHORT).show();
                int id = buttonEdit.get(i).getId();
                getDataByID(id);

            } /* jika yang diklik adalah button delete */
            else if (view.getId() == buttonDelete.get(i).getId()
                && view.getTag().toString().trim().equals("Delete")) {
                // Toast.makeText(MainActivity.this, "Delete : " +
                // buttonDelete.get(i).getId(), Toast.LENGTH_SHORT).show();
                int id = buttonDelete.get(i).getId();
                deleteBiodata(id);

            }
        }
    }
}

public void deleteBiodata(int id) {
    sqliteHelper.hapus_biodata(id);
}
}

```

```

        /* restart activity */
        finish();
        startActivity(getIntent());
    }
    public void getDataByID(int id) {
        String namaEdit = null, alamatEdit = null;
        HashMap<String, String> hashMapBiodata = sqLiteHelper.tampil_biodata_berdasarkan_id(id);

        for (int i = 0; i < hashMapBiodata.size(); i++) {
            namaEdit = hashMapBiodata.get("nama");
            alamatEdit = hashMapBiodata.get("alamat");
        }

        LinearLayout layoutInput = new LinearLayout(this);
        layoutInput.setOrientation(LinearLayout.VERTICAL);

        // buat id tersembunyi di alertbuilder
        final TextView viewId = new TextView(this);
        viewId.setText(String.valueOf(id));
        viewId.setTextColor(Color.TRANSPARENT);
        layoutInput.addView(viewId);

        final EditText editNama = new EditText(this);
        editNama.setText(namaEdit);
        layoutInput.addView(editNama);

        final EditText editAlamat = new EditText(this);
        editAlamat.setText(alamatEdit);
        layoutInput.addView(editAlamat);

        AlertDialog.Builder builderEditBiodata = new AlertDialog.Builder(this);
        builderEditBiodata.setTitle("Update Biodata");
        builderEditBiodata.setView(layoutInput);
        builderEditBiodata.setPositiveButton("Update", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String nama = editNama.getText().toString();
                String alamat = editAlamat.getText().toString();

                System.out.println("Nama : " + nama + " Alamat : "
                    + alamat);

                sqLiteHelper.update_biodata(Integer.parseInt(viewId.getText().toString()), editNama.getText().toString(), editAlamat.getText().toString());

                /* restart activity */
                finish();
                startActivity(getIntent());
            }
        });

        builderEditBiodata.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });
        builderEditBiodata.show();
    }
    public void tambahBiodata() {
        /* layout akan ditampilkan pada AlertDialog */
        LinearLayout layoutInput = new LinearLayout(this);
        layoutInput.setOrientation(LinearLayout.VERTICAL);

        final EditText editNama = new EditText(this);
        editNama.setHint("Nama");
        layoutInput.addView(editNama);

        final EditText editAlamat = new EditText(this);
        editAlamat.setHint("Alamat");
        layoutInput.addView(editAlamat);

        AlertDialog.Builder builderInsertBiodata = new AlertDialog.Builder(this);
        builderInsertBiodata.setTitle("Insert Biodata");
        builderInsertBiodata.setView(layoutInput);
        builderInsertBiodata.setPositiveButton("Insert", new DialogInterface.OnClickListener() {
    }

```

```
@Override
public void onClick(DialogInterface dialog, int which) {
    String nama = editNama.getText().toString();
    String alamat = editAlamat.getText().toString();

    System.out.println("Nama : " + nama + " Alamat : "
        + alamat);

    sqliteHelper.tambah_biodata(nama, alamat);
    /* restart activity */
    finish();
    startActivity(getIntent());
}

});

builderInsertBiodata.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
builderInsertBiodata.show();
}
}
```

Output Program :



Figure 5.4 Output Display Crud

Practical

1. try the entire code above to understand every line of code and run it on emulator

Practical Tasks

1. Create a new project with the database and the table adjusts to the title of the final project

Practical

1. try the entire code above to understand every line of code and run it on emulator

Optional Source Code

```

android.widget.LinearLayout;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity implements OnClickListener
{
    SQLiteHelper sqliteHelper = new SQLiteHelper(this);
    TableLayout tabelBiodata;
    Button buttonTambahBiodata;
    ArrayList<Button>buttonEdit = newArrayList<Button>();
    ArrayList<Button>buttonDelete = new ArrayList<Button>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main );
        tabelBiodata = (TableLayout) findViewById(R.id.tableBiodata );
        buttonTambahBiodata = (Button) findViewById(R.id.buttonTambahBiodata );
        buttonTambahBiodata.setOnClickListener(this);
        TableRow barisTabel = new TableRow(this);
        barisTabel.setBackgroundColor(Color.RED );
        TextView viewHeaderId = new TextView(this);
        TextView viewHeaderNama = new TextView(this);
        TextView viewHeaderAlamat = new TextView(this);
        TextView viewHeaderAction = new TextView(this);
        viewHeaderId.setText("ID");
        viewHeaderNama.setText("Nama");
        viewHeaderAlamat.setText("Alamat");
        viewHeaderAction.setText("Action");
        viewHeaderId.setPadding(5, 1, 5, 1);
        viewHeaderNama.setPadding(5, 1, 5, 1);
        viewHeaderAlamat.setPadding(5, 1, 5, 1);
        viewHeaderAction.setPadding(5, 1, 5, 1);
        barisTabel.addView(viewHeaderId);
        barisTabel.addView(viewHeaderNama);
        barisTabel.addView(viewHeaderAlamat);
        barisTabel.addView(viewHeaderAction);
        tabelBiodata.addView(barisTabel, new
        TableLayout.LayoutParams(LayoutParams.WRAP_CONTENT ,
        LayoutParams.WRAP_CONTENT ));
        ArrayList<HashMap<String, String>> arrayListBiodata =
        sqliteHelper.tampil_semua_biodata();
        if (arrayListBiodata.size() >0) {
            for (int i = 0; i < arrayListBiodata.size(); i++) {
                // ambil masing-masing hasmap dari arrayListBiodata
                HashMap<String, String> hashMapRecordBiodata =
                arrayListBiodata.get(i);
                // JSONObject jsonChildNode = arrayBiodata.getJSONObject(i);
                String name = hashMapRecordBiodata.get("nama");
                String alamat = hashMapRecordBiodata.get("alamat");
                String id = hashMapRecordBiodata.get("id_biodata");
                System.out.println("Nama :" + name);
                System.out.println("Alamat :" + alamat);
                System.out.println("ID :" + id);
                barisTabel = new TableRow(this);
                if (i % 2 == 0) {
                    barisTabel.setBackgroundColor(Color.LTGRAY );
                }
                TextView viewId = new TextView(this);
                viewId.setText(id);
                viewId.setPadding(5, 1, 5, 1);
                barisTabel.addView(viewId);
                TextView viewNama = new TextView(this);
                viewNama.setText(name);
                viewNama.setPadding(5, 1, 5, 1);
                barisTabel.addView(viewNama);
                TextView viewAlamat = new TextView(this);
    
```



```

viewAlamat.setText(alamat);
viewAlamat.setPadding(5, 1, 5, 1);
barisTabel.addView(viewAlamat);
buttonEdit.add(i, new Button(this));
buttonEdit.get(i).setId(Integer.parseInt(id));
buttonEdit.get(i).setTag("Edit");
buttonEdit.get(i).setText("Edit");
buttonEdit.get(i).setOnClickListener(this);
barisTabel.addView(buttonEdit.get(i));
buttonDelete.add(i, new Button(this));
buttonDelete.get(i).setId(Integer.parseInt(id));
buttonDelete.get(i).setTag("Delete");
buttonDelete.get(i).setText("Delete");
buttonDelete.get(i).setOnClickListener(this);
barisTabel.addView(buttonDelete.get(i));
tabelBiodata.addView(barisTabel, new TableLayout.LayoutParams(
    LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
}
}
}
@Override
public void onClick(View view) {
    if (view.getId() == R.id.buttonTambahBiodata) {
        // Toast.makeText(MainActivity.this, "Button Tambah Data",
        // Toast.LENGTH_SHORT).show();
        tambahBiodata();
    } else {
        /*
        * Melakukan pengecekan pada data array, agar sesuai dengan index
        * masing-masing button
        */
        for (int i = 0; i < buttonEdit.size(); i++) {
            /* jika yang diklik adalah button edit */
            if (view.getId() == buttonEdit.get(i).getId()
                && view.getTag().toString().trim().equals("Edit")) {
                // Toast.makeText(MainActivity.this, "Edit : " +
                // buttonEdit.get(i).getId(), Toast.LENGTH_SHORT).show();
                int id = buttonEdit.get(i).getId();
                getDataByID(id);
            } /* jika yang diklik adalah button delete */
            else if (view.getId() == buttonDelete.get(i).getId()
                && view.getTag().toString().trim().equals("Delete")) {
                // Toast.makeText(MainActivity.this, "Delete : " +
                // buttonDelete.get(i).getId(), Toast.LENGTH_SHORT).show();
                int id = buttonDelete.get(i).getId();
                deleteBiodata(id);
            }
        }
    }
}

public void deleteBiodata(int id) {
    sqliteHelper.hapus_biodata(id);
    /* restart activity */
    finish();
    startActivity(getIntent());
}

public void getDataByID(int id) {
    String namaEdit = null, alamatEdit = null;
    HashMap<String, String> hashMapBiodata =
        sqliteHelper.tampil_biodata_berdasarkan_id(id);
    for (int i = 0; i < hashMapBiodata.size(); i++) {
        namaEdit = hashMapBiodata.get("nama");
        alamatEdit = hashMapBiodata.get("alamat");
    }
    LinearLayout layoutInput = new LinearLayout(this);
    layoutInput.setOrientation(LinearLayout.VERTICAL);
    // buat id tersembunyi di alertbuilder
    final TextView viewId = new TextView(this);

```

```

viewId.setText(String.valueOf (id));
viewId.setTextColor(Color.TRANSPARENT );
layoutInput.addView(viewId);
final EditText editNama = new EditText(this);
editNama.setText(namaEdit);
layoutInput.addView(editNama);
final EditText editAlamat = new EditText(this);
editAlamat.setText(alamatEdit);
layoutInput.addView(editAlamat);
AlertDialog.Builder builderEditBiodata = new AlertDialog.Builder(this);
builderEditBiodata.setTitle("Update Biodata");
builderEditBiodata.setView(layoutInput);
builderEditBiodata.setPositiveButton("Update",new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        String nama = editNama.getText().toString();
        String alamat = editAlamat.getText().toString();
        System.out.println("Nama : " + nama + " Alamat : "
+ alamat);
        sqliteHelper.update_biodata(Integer.parseInt (viewId.getText().toString()),
editNama.getText().toString(), editAlamat.getText().toString());
        /* restart acrtivity */
        finish();
        startActivity(getIntent());
    }
});
builderEditBiodata.setNegativeButton("Cancel",new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
builderEditBiodata.show();
}
public void tambahBiodata() {
    /* layout akan ditampilkan pada AlertDialog */
    LinearLayout layoutInput = new LinearLayout(this);
    layoutInput.setOrientation(LinearLayout.VERTICAL );
    final EditText editNama = new EditText(this);
    editNama.setHint("Nama");
    layoutInput.addView(editNama);
    final EditText editAlamat = new EditText(this);
    editAlamat.setHint("Alamat");
    layoutInput.addView(editAlamat);
    AlertDialog.Builder builderInsertBiodata = new
AlertDialog.Builder(this);
    builderInsertBiodata.setTitle("Insert Biodata");
    builderInsertBiodata.setView(layoutInput);
    builderInsertBiodata.setPositiveButton("Insert",new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            String nama = editNama.getText().toString();
            String alamat = editAlamat.getText().toString();
            System.out.println("Nama : " + nama + " Alamat : "
+ alamat);
            sqliteHelper.tambah_biodata(nama, alamat);
            /* restart acrtivity */
            finish();
            startActivity(getIntent());
        }
    });
    builderInsertBiodata.setNegativeButton("Cancel",new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {

```

```

dialog.cancel();
}
});
builderInsertBiodata.show();
}
}
Optional jika bingung membaca kode mainactivity silahkan copas kode
dibawah]
SQLiteHelper.java
package com.achmad_14650002.crud_sqlite;
/**
 * Created by coldwarrior on 29/09/16.
 */
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;
import java.util.HashMap;
public class SQLiteHelper extends SQLiteOpenHelper{
private static final String nama_database = "database_mahasiswa.db";
private static final int versi_database = 1;
private static final String query_buat_tabel_biodata_pemain = "CREATE TABLE
IF NOT EXISTS tabel_biodata(id_biodata INTEGER PRIMARY KEY AUTOINCREMENT, nama
TEXT,alamat TEXT)";
private static final String query_hapus_tabel_biodata_pemain = "DROP TABLE
IF EXISTS query_buat_tabel_biodata_pemain";
public SQLiteHelper(Context context) {
super(context, nama_database , null, versi_database );
}
@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
sqLiteDatabase.execSQL(query_buat_tabel_biodata_pemain );
System.out .println("tabel_biodata sudah dibuat");
}
@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int versi_lama, int
versi_baru) {
sqLiteDatabase.execSQL(query_hapus_tabel_biodata_pemain );
onCreate(sqLiteDatabase);
}
public void tambah_biodata(String nama, String alamat) {
SQLiteDatabase database = this.getWritableDatabase();
ContentValues values = new ContentValues();
values.put("nama", nama);
values.put("alamat", alamat);
database.insert("tabel_biodata", null, values);
database.close();
}
public ArrayList<HashMap<String, String>> tampil_semua_biodata() {
// deklarasikan sebuah arraylist yang bisa menampung hashmap
ArrayList<HashMap<String, String>> arrayListBiodata = new
ArrayList<HashMap<String, String>>();
SQLiteDatabase database = this.getWritableDatabase();
Cursor cursor = database.rawQuery("SELECT * FROM tabel_biodata", null);
// kursor langsung diarkan ke posisi paling awal data pada tabel_biodata
if (cursor.moveToFirst()) {
do {
// deklarasikan sebuah hashmap, yang bisa menamp
HashMap<String, String> hashMapBiodata = new HashMap<String,
String>();
// masukkan masing-masing field dari tabel_biodata ke dalam
hashMapBiodata
//pastikan id_biodata, nama dan alamat sama persis dengan field
yang ada pada tabel_biodata
hashMapBiodata.put("id_biodata", cursor.getString(0));
hashMapBiodata.put("nama", cursor.getString(1));

```

```
hashMapBiodata.put("alamat", cursor.getString(2));  
// masukkan hashMapBiodata ke dalam arrayListBiodata  
arrayListBiodata.add(hashMapBiodata);  
} while (cursor.moveToNext());  
}  
return arrayListBiodata;  
}  
public int update_biodata(int id, String nama, String alamat) {  
    SQLiteDatabase database = this.getWritableDatabase();  
    ContentValues recordBiodata = new ContentValues();  
    recordBiodata.put("nama", nama);  
    recordBiodata.put("alamat", alamat);  
    return database.update("tabel_biodata", recordBiodata, "id_biodata=" +  
id, null);  
}  
public void hapus biodata(int id) {  
    SQLiteDatabase database = this.getWritableDatabase();  
    database.execSQL("DELETE FROM tabel_biodata WHERE id_biodata=" + id +  
    "");  
    database.close();  
}  
public HashMap<String, String> tampil_biodata_berdasarkan_id(int id) {  
    SQLiteDatabase database = this.getReadableDatabase();  
    HashMap<String, String> hashMapBiodata = new HashMap<String, String>();  
    Cursor cursor = database.rawQuery("SELECT * FROM tabel_biodata WHERE  
id_biodata=" + id + "", null);  
    if (cursor.moveToFirst()) {  
        do {  
            hashMapBiodata.put("id_biodata", cursor.getString(0));  
            hashMapBiodata.put("nama", cursor.getString(1));  
            hashMapBiodata.put("alamat", cursor.getString(2));  
        } while (cursor.moveToNext());  
    }  
    return hashMapBiodata;  
}  
}
```

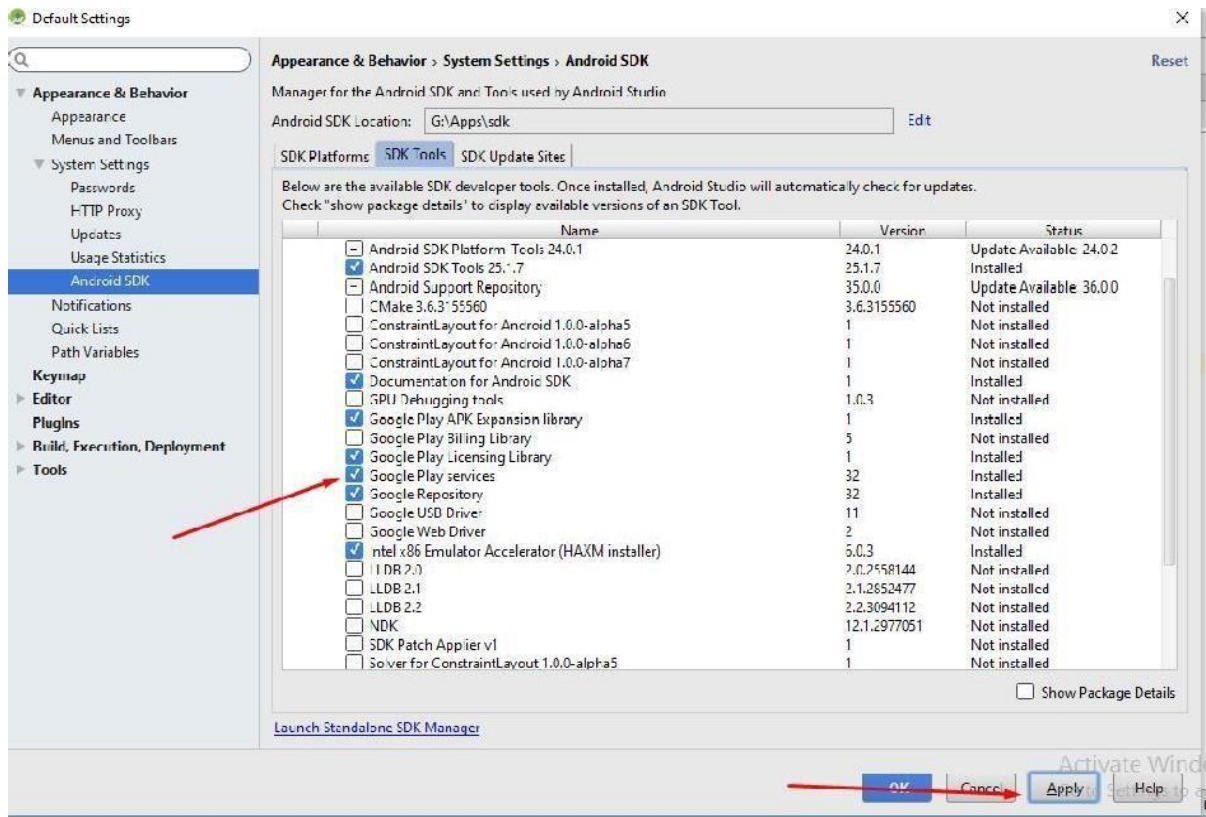
MODUL 6

API

A. Install Google Play Service SDK

The first way is by installing Google Android SDK Service Play in the Studio.

1. Go to the Menu Manager SDK
2. Select Appearance & Behavior < System Setting < Android SDK
3. on the select SDK Android SDK Tools

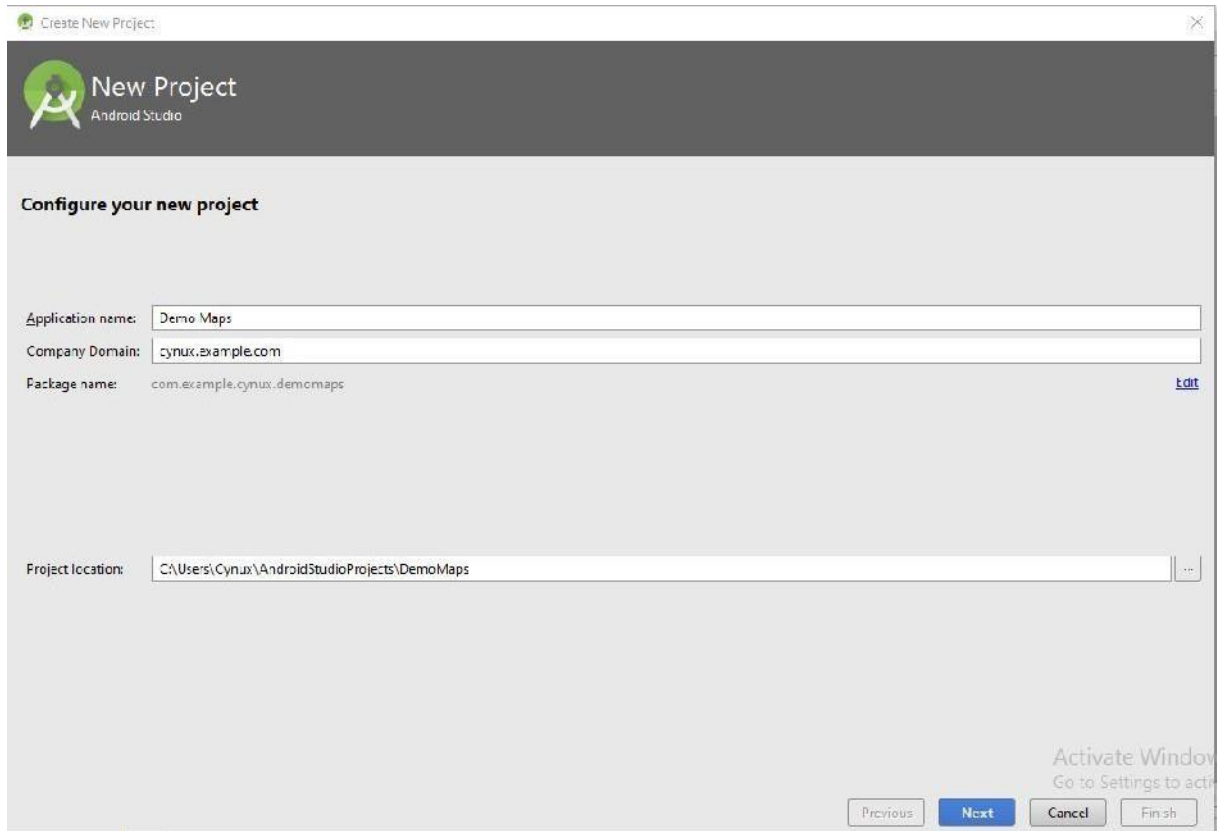


4. Search Google Play Services and checked
5. Click Apply then we will go to the Process Download Google Play Services

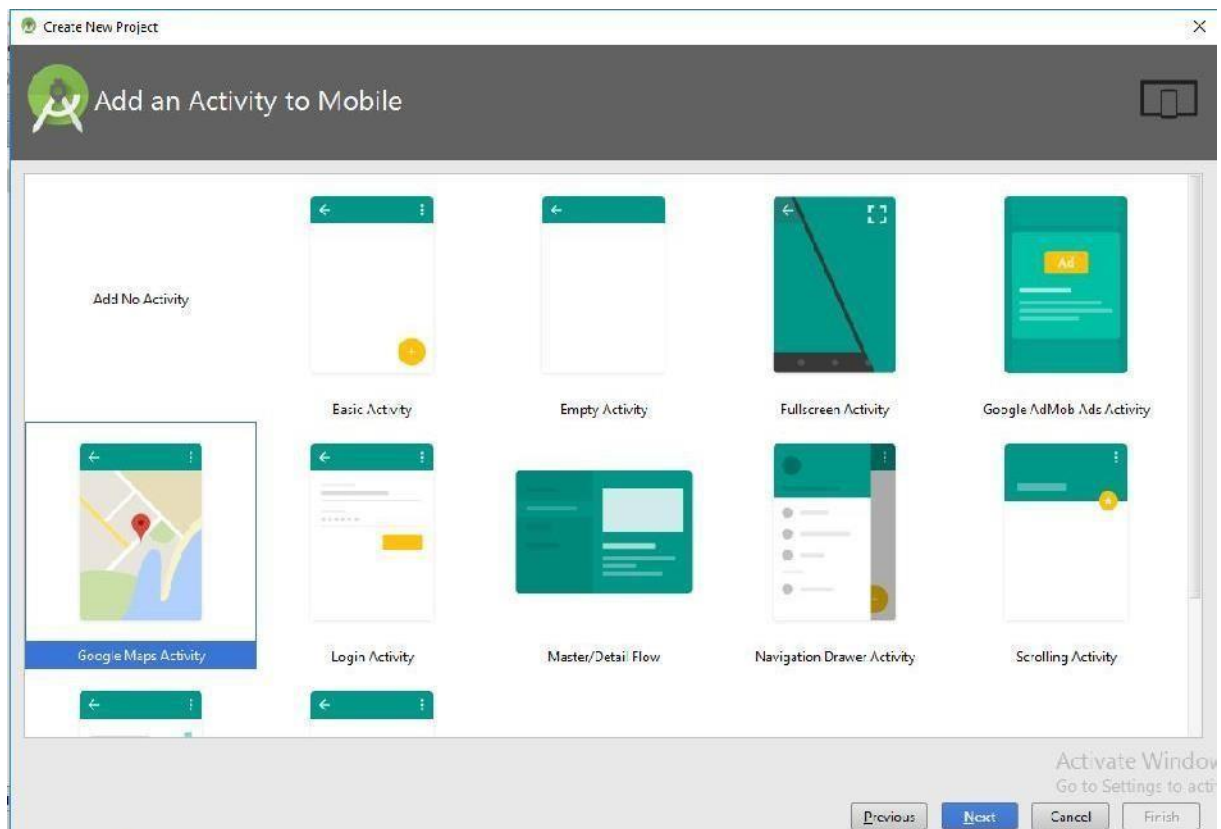
B. Create A Project Of Google Maps

The next step we will create a new application project

1. New Project, in the Application Name give name the fit you want. Here I will give you the name of "Demo Maps". Then Select Next

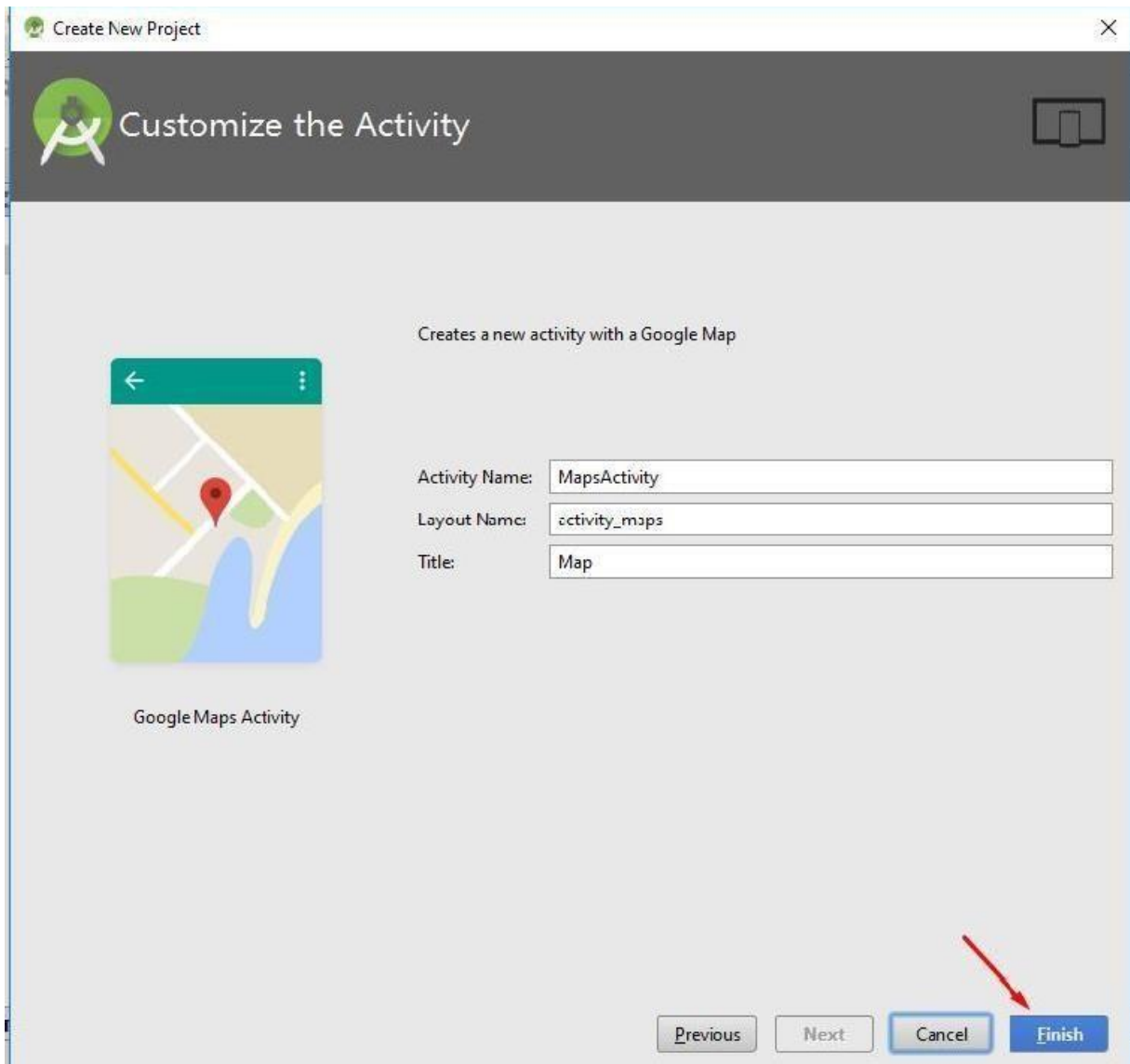


2. Next we will be faced with a screen to select Minimal Android SDK 4.2 Jelly Bean in order to can the device used to JB
3. after click Next we will choose activitynya. Usually we use an empty activity, so this time we'll use Google Maps Activity and click N

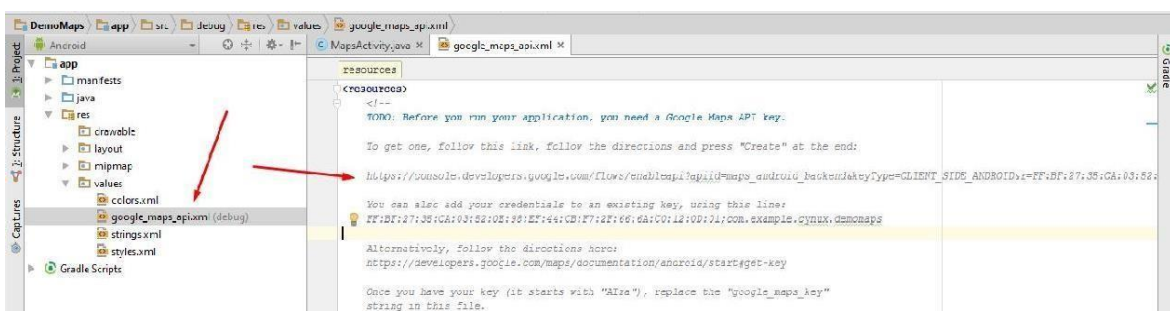


ext

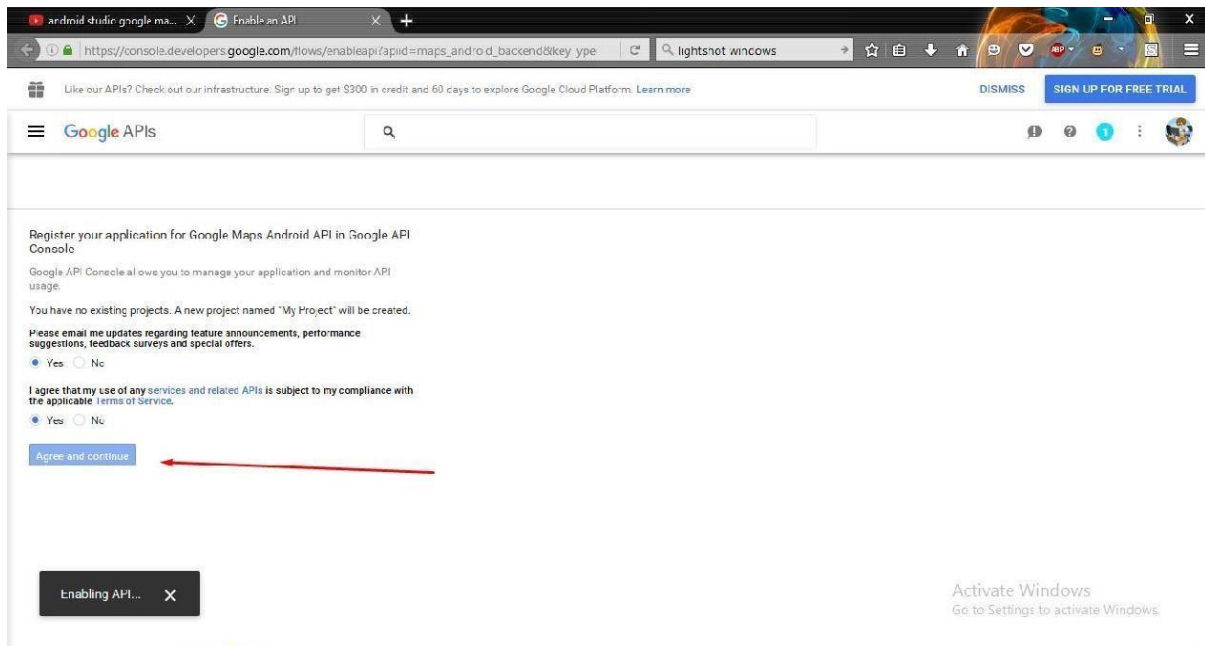
4. Click Next again, and please fill the content of the Activity name and Layout Name. I will keep using the defaults only. And the last step click Finish



5. An then opens the window of our project. In the tree directory, select the App < res < values < chose google_maps_api.xml. Next, Copy the files on the credentials provided in the google_maps_api.xml paste in your browser.

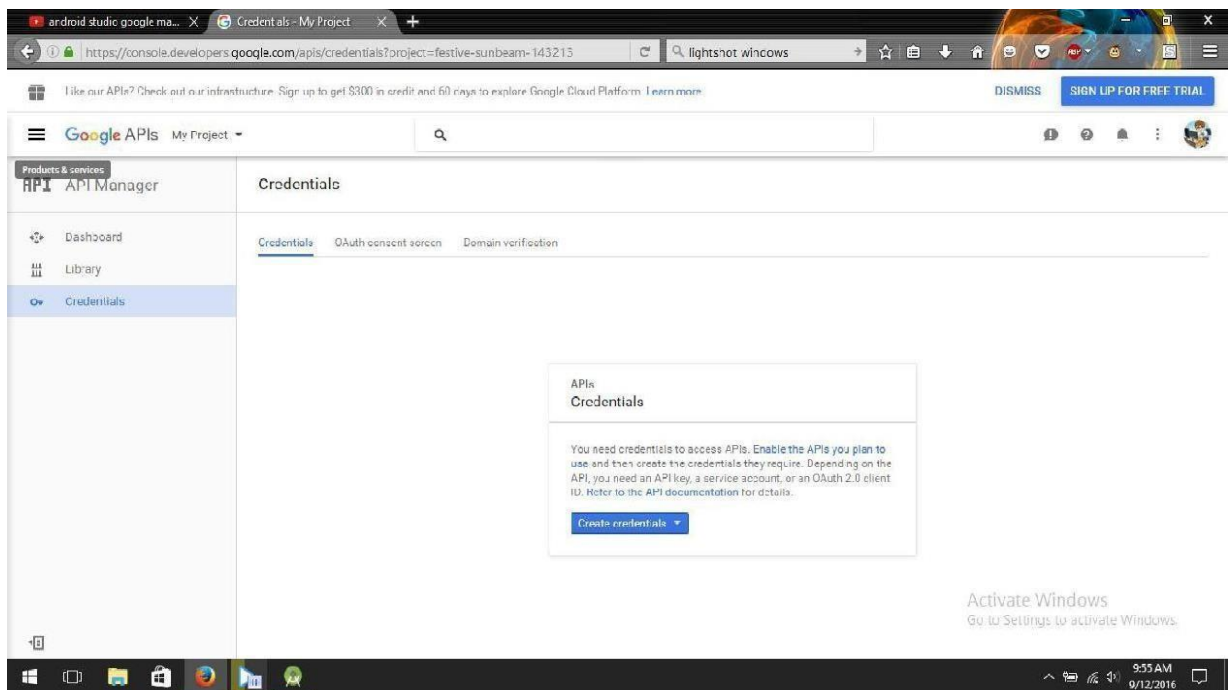


6. Next will appear as below. Simply Accept and Continue

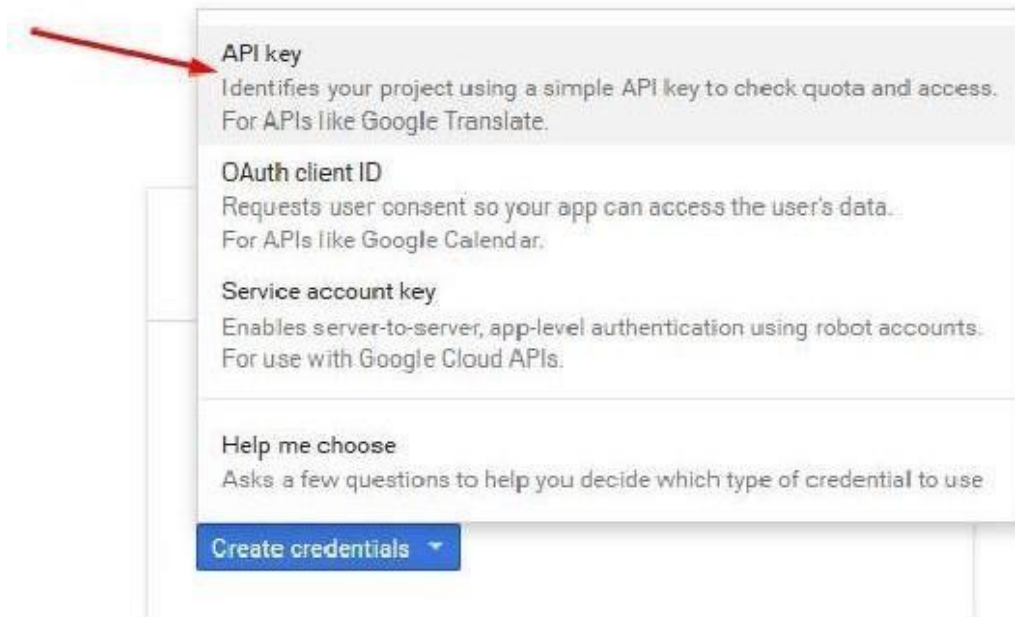


7.

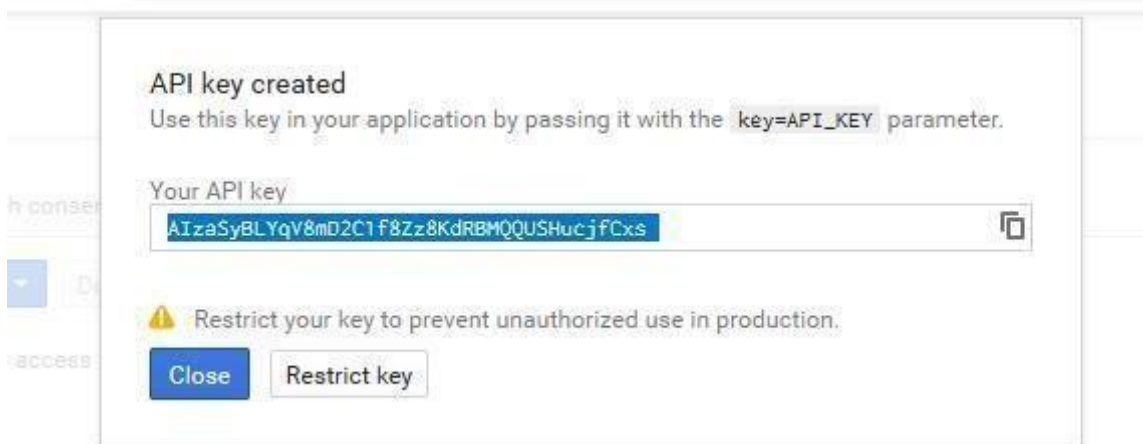
8. Select menu at the bottom left hand corner, next select credentials and then make Create Credentials



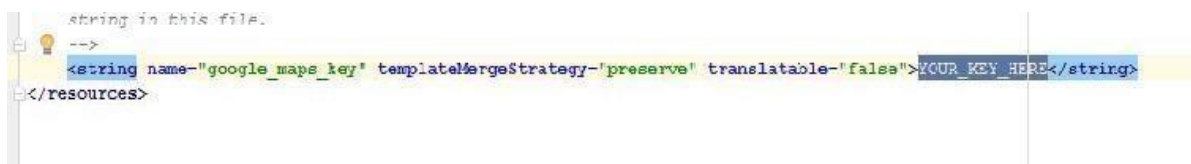
9. Then will display options like the image below, click the API Key



10. Once we will get the API Key that we will be pastekan our Studio in Android



11. Open again to our Android Studio. Open the google_maps_api.xml file, the pastekan file into the "Your Key Here"



After that, we can Run our App. Can using Emulator Android Studio or Genymotion or using our own mobile phones.

MODUL 7

Database SQLite

SQLite is a software RDBMS (Relational Database Management System) that supports natively (original) for Android devices. SQLite is a database management system, which has properties of ACID-compliant, which is programmed with C language, and have the size or the size of the memory that is relatively small. Because Sqlite database engine includes embedded (embedded), so Sqlite command that you can use only the standard commands only. And Sqlite only supports data types like INTEGER, DATETIME, NUMERIC, TEXT, and other things.

Line of source code in Sqlite also are public domain developed by d. Richard Hipp. That means you can use Sqlite freely for any purpose, whether commercial or private in nature. SQLite supports all operating system platforms, such as: Windows, Linux, Android, and iOS. Especially on the Android OS, Android apps on Sqlite will converge in a system called Android Runtime. In SQLite built in available in the android library. So can use it, directly, without the need to wear a piece of software or importing other libraries, while developing Android applications. Can also create a Sqlite database, use the SQL user interface.

On learning android this time, will try to create an application entry for Yourself by using the Android Sqlite database. Here will also try to apply the basic functions of a database that is a CRUD (Create, Read, Update and Delete).

1. Implement The Code

- Table Structure

The structure of the table that will be created will be approximately like this:

Field	Type Data	Key
no	Integer	Primary Key
nama	Text Null	
tgl	Text Null	
jk	Text Null	
alamat	Text Null	

- SQLiteOpenHelper

SQLiteOpenHelper is a subclass, which is used to specify the database name and the version of the database that is being used. Able to apply the method in this class such as: onCreate (SQLiteDatabase), onUpgrade (SQLiteDatabase) and onOpen (SQLiteDatabase).

1. Here is created the file class Activity, named URDataHelper.java. Here will be applied to a method or function, such as: SQLiteOpenHelper from onCreate (). who will run the database, if the database is not yet available or does not exist.

typing the line below the source code to the class file Activity named URDataHelper.java

```
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

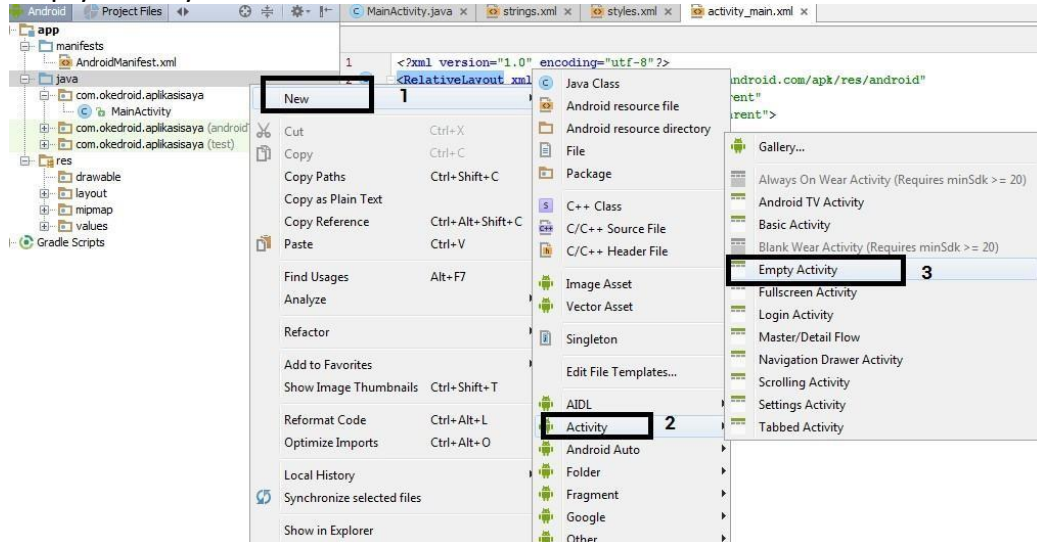
public class UpdateBiodataextends AppCompatActivity {
    protected Cursor cursor;
    DataHelperdbHelper;
    Button ton1, ton2;
    EditTexttext1, text2, text3, text4, text5;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_biodata);

        dbHelper= new DataHelper(this);
        text1 = (EditText) findViewById(R.id.editText1);
        text2 = (EditText) findViewById(R.id.editText2);
        text3 = (EditText) findViewById(R.id.editText3);
        text4 = (EditText) findViewById(R.id.editText4);
        text5 = (EditText) findViewById(R.id.editText5);
        SQLiteDatabasedb = dbHelper.getReadableDatabase();
        cursor = db.rawQuery("SELECT * FROM biodata WHERE nama = '" +
            getIntent().getStringExtra("nama") + "'",null);
        cursor.moveToFirst();
        if (cursor.getCount(>0)
            {
                cursor.moveToPosition(0);
                text1.setText(cursor.getString(0).toString());
                text2.setText(cursor.getString(1).toString());
                text3.setText(cursor.getString(2).toString());
                text4.setText(cursor.getString(3).toString());
                text5.setText(cursor.getString(4).toString());
            }
        ton1 = (Button) findViewById(R.id.button1);
        ton2 = (Button) findViewById(R.id.button2);
        // daftarkan even onClickpadabtnSimpan
        ton1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                SQLiteDatabasedb = dbHelper.getWritableDatabase();
                db.execSQL("update biodata set nama='"+
                    text2.getText().toString() + "', tgl='"+
                    text3.getText().toString()+"', jk='"+
                    text4.getText().toString() + "', alamat='"+
                    text5.getText().toString() + "' where no='"+
                    text1.getText().toString()+"");
                Toast.makeText(getApplicationContext(), "Berhasil", Toast.LENGTH_LONG).show();
                MainActivity.ma.RefreshList();
                finish();
            }
        });
        ton2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }
    @Override
    public booleanonCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

2. After that, we'll create a New Activity

Here will be made 3 Activity is 3 file java ,3 file layout, for it will be created a new Activity in advance:

On the folder java or res, can right click Select New, select the Activity, then an Empty Activity.



New Android activity on the contents of the name of the activity or the java file and name the file layout (it's up to you guys). As an example I will create a file with the name of class activity:

Biodata.java,
ViewBiodata.java,
UpdateBiodata.java

file layout

activity_biodata.xml
activity_View_biodata.xml.
activity_update_biodata.xml.

Then you can choose the Finish button.

3. After that when the source code below in the respective file.

MainActivity.java

```
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {
    String[] daftar;
    ListView listView01;
    Menu menu;
    protected Cursor cursor;
```

```

DataHelperdbcenter;
public static MainActivity ma;

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

        Button btn=(Button)findViewById(R.id.button2);

btn.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
Intent inte = new Intent(MainActivity.this, BuatBiodata.class);
startActivity(inte);
        }
});

ma = this;
dbcenter= new DataHelper(this);
RefreshList();
}

public void RefreshList(){
SQLiteDatabase db = dbcenter.getReadableDatabase();
cursor = db.rawQuery("SELECT * FROM biodata",null);
daftar= new String[cursor.getCount()];
cursor.moveToFirst();
for (intcc=0; cc <cursor.getCount(); cc++){
cursor.moveToPosition(cc);
daftar[cc] = cursor.getString(1).toString();
}
ListView01 = (ListView)findViewById(R.id.listView1);
ListView01.setAdapter(new ArrayAdapter(this, android.R.layout.simple_list_item_1,
daftar));
ListView01.setSelected(true);
ListView01.setOnItemClickListener(new OnItemClickListener() {

public void onItemClick(AdapterView arg0, View arg1, intarg2, long arg3) {
final String selection = daftar[arg2]; //.getItemAtPosition(arg2).toString();
final CharSequence[] dialogitem = {"View Biodata", "Update Biodata", "Delete
Biodata"};
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
builder.setTitle("Option");
builder.setItems(dialogitem, new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, intitem) {
switch(item){
case 0 :
    
```

```

        Intent i = new Intent(getApplicationContext(),
        LihatBiodata.class);
        i.putExtra("nama", selection);
        startActivity(i);
        break;
        case 1 :
            Intent in = new Intent(getApplicationContext(),
            UpdateBiodata.class);
            in.putExtra("nama", selection);
            startActivity(in);
            break;
        case 2 :
            SQLiteDatabase db = dbcenter.getWritableDatabase();
            db.execSQL("delete from biodata where nama = '"+selection+"'");
            RefreshList();
            break;
    }
    });
    builder.create().show();
    ((ArrayAdapter)ListView01.getAdapter()).notifyDataSetChanged();
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
}

```

activity_main.xml

```

<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"
    android:background="#ecf0f1"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/button1"
        style="?android:attr/borderlessButtonStyle"
        android:drawableLeft="@drawable/icon_add"
        android:text="Buat Biodata Baru" />

    <ListView
        android:id="@+id/listView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button2" android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">
    </ListView>

</RelativeLayout>

```

SQLiteDatabase is a basic working class, for database sqlite in Android devices. SQLiteDatabase will run a command SQL directly with the method exceSQL (). And will also perform other database management in General, the methods used, such as: (a) Insert, Update() and Delete ().

Biodata.java

```

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Biodata extends AppCompatActivity {
    protected Cursor cursor;
    DataHelper dbHelper;
    Button ton1, ton2;
    EditText text1, text2, text3, text4, text5;

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

        dbHelper = new DataHelper(this);
        text1 = (EditText) findViewById(R.id.editText1);
        text2 = (EditText) findViewById(R.id.editText2);
        text3 = (EditText) findViewById(R.id.editText3);
        text4 = (EditText) findViewById(R.id.editText4);
        text5 = (EditText) findViewById(R.id.editText5);
        ton1 = (Button) findViewById(R.id.button1);
        ton2 = (Button) findViewById(R.id.button2);

        ton1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                SQLiteDatabase db = dbHelper.getWritableDatabase();
                db.execSQL("insert into biodata(no, nama, tgl, jk, alamat) values('" +
                    text1.getText().toString() + "','" +
                    text2.getText().toString() + "','" +
                    text3.getText().toString() + "','" +
                    text4.getText().toString() + "','" +
                    text5.getText().toString() + "')");
                Toast.makeText(getApplicationContext(), "success", Toast.LENGTH_LONG).show();
                MainActivity.ma.RefreshList();
                finish();
            }
        });
        ton2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

```

activity_biodata.xml

```

<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=".BuatBiodata" >

<EditText
android:id="@+id/editText1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_below="@+id/textView1" >
<requestFocus/>
</EditText>

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:text="Nomor" />

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText1"
android:layout_below="@+id/editText1"
android:layout_marginTop="10dp"
android:text="Nama" />

<EditText
android:id="@+id/editText2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView2"
android:layout_below="@+id/textView2" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText2"
android:layout_below="@+id/editText2"
android:layout_marginTop="10dp"
android:text="TanggalLahir" />

<EditText
android:id="@+id/editText3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView3"
android:layout_below="@+id/textView3" />

<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText3"
android:layout_below="@+id/editText3"
android:layout_marginTop="10dp"
android:text="JenisKelamin" />

<EditText
android:id="@+id/editText4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView4"
android:layout_below="@+id/textView4" />

<TextView
android:id="@+id/textView5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText4"
android:layout_below="@+id/editText4"
android:layout_marginTop="10dp"
android:text="Alamat" />

```



```
<EditText
android:id="@+id/editText5"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView5"
android:layout_below="@+id/textView5" />
```

```
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText5"
android:layout_alignParentBottom="true"
android:text="Simpan" />
```

```
<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button1"
android:layout_alignBottom="@+id/button1"
android:layout_toRightOf="@+id/textView4"
android:text="Back" />
```

```
</RelativeLayout>
```

ViewBiodata.java

```
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class LihatBiodata extends AppCompatActivity {
    protected Cursor cursor;
    DataHelper dbHelper;
    Button ton2;
    TextView text1, text2, text3, text4, text5;

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

        dbHelper = new DataHelper(this);
        text1 = (TextView) findViewById(R.id.textView1);
        text2 = (TextView) findViewById(R.id.textView2);
        text3 = (TextView) findViewById(R.id.textView3);
        text4 = (TextView) findViewById(R.id.textView4);
        text5 = (TextView) findViewById(R.id.textView5);
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        cursor = db.rawQuery("SELECT * FROM biodata WHERE nama = '" +
            getIntent().getStringExtra("nama") + "'", null);
        cursor.moveToFirst();
        if (cursor.getCount() > 0)
        {
            cursor.moveToPosition(0);
            text1.setText(cursor.getString(0).toString());
            text2.setText(cursor.getString(1).toString());
            text3.setText(cursor.getString(2).toString());
            text4.setText(cursor.getString(3).toString());
            text5.setText(cursor.getString(4).toString());
        }
        ton2 = (Button) findViewById(R.id.button1);
        ton2.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }
}
```

```

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

```

activity_lihat_biodata.xml

```

<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=".LihatBiodata" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="104dp"
        android:layout_marginTop="20dp"
        android:text="TextView" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="20dp"
        android:text="TextView" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="20dp"
        android:text="TextView" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView3"
        android:layout_below="@+id/textView3"
        android:layout_marginTop="20dp"
        android:text="TextView" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textView4"
        android:layout_below="@+id/textView4"
        android:layout_marginTop="20dp"
        android:text="TextView" />

    <TextView
        android:id="@+id/TextView05"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView5"
        android:layout_alignBottom="@+id/textView5"
        android:layout_alignLeft="@+id/TextView03"
        android:text="Alamat" />

```

```

<TextView
android:id="@+id/TextView03"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/textView4"
android:layout_alignBottom="@+id/textView4"
android:layout_alignLeft="@+id/TextView04"
android:text="JenisKelamin" />

<TextView
android:id="@+id/TextView04"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/textView3"
android:layout_alignBottom="@+id/textView3"
android:layout_alignLeft="@+id/TextView02"
android:text="TanggalLahir" />

<TextView
android:id="@+id/TextView02"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/textView2"
android:layout_alignBottom="@+id/textView2"
android:layout_alignLeft="@+id/TextView01"
android:text="Nama" />

<TextView
android:id="@+id/TextView01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/textView2"
android:layout_alignParentLeft="true"
android:text="Nomor" />

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/TextView05"
android:layout_below="@+id/TextView05"
android:layout_marginTop="34dp"
android:text="Back" />

</RelativeLayout>
    
```

UpdateBiodata.java

```

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class UpdateBiodata extends AppCompatActivity {
    protected Cursor cursor;
    DBHelper dbHelper;
    Button ton1, ton2;
    EditText text1, text2, text3, text4, text5;

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

        dbHelper = new DBHelper(this);
        text1 = (EditText) findViewById(R.id.editText1);
        text2 = (EditText) findViewById(R.id.editText2);
        text3 = (EditText) findViewById(R.id.editText3);
        text4 = (EditText) findViewById(R.id.editText4);
        text5 = (EditText) findViewById(R.id.editText5);
        SQLiteDatabase db = dbHelper.getReadableDatabase();
    }
}
    
```

```

cursor = db.rawQuery("SELECT * FROM biodata WHERE nama = '" +
getIntent().getStringExtra("nama") + "'",null);
cursor.moveToFirst();
if (cursor.getCount()>0)
{
    cursor.moveToPosition(0);
    text1.setText(cursor.getString(0).toString());
    text2.setText(cursor.getString(1).toString());
    text3.setText(cursor.getString(2).toString());
    text4.setText(cursor.getString(3).toString());
    text5.setText(cursor.getString(4).toString());
}
ton1 = (Button) findViewById(R.id.button1);
ton2 = (Button) findViewById(R.id.button2);
// daftarkan even onClickpadaabtnSimpan
ton1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
SQLiteDatabasedb = dbHelper.getWritableDatabase();
db.execSQL("update biodata set nama='"+
text2.getText().toString() +"', tgl='"+
text3.getText().toString()+"', jk='"+
text4.getText().toString() +"', alamat='"+
text5.getText().toString() + "' where no='"+
text1.getText().toString()+"'");
Toast.makeText(getApplicationContext(), "Berhasil", Toast.LENGTH_LONG).show();
MainActivity.ma.RefreshList();
finish();
}
});
ton2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
finish();
}
});
}

@Override
public booleanonCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.main, menu);
return true;
}
}

```

activity_update_biodata.xml

```

<RelativeLayoutxmlns: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=".UpdateBiodata" >
<EditText
android:id="@+id/editText1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_below="@+id/textView1" >

<requestFocus/>
</EditText>

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:text="Nomor" />

<TextView

```

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText1"
android:layout_below="@+id/editText1"
android:layout_marginTop="10dp"
android:text="Nama" />

<EditText
android:id="@+id/editText2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView2"
android:layout_below="@+id/textView2" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText2"
android:layout_below="@+id/editText2"
android:layout_marginTop="10dp"
android:text="TanggalLahir" />

<EditText
android:id="@+id/editText3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView3"
android:layout_below="@+id/textView3" />

<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText3"
android:layout_below="@+id/editText3"
android:layout_marginTop="10dp"
android:text="JenisKelamin" />

<EditText
android:id="@+id/editText4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView4"
android:layout_below="@+id/textView4" />

<TextView
android:id="@+id/textView5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText4"
android:layout_below="@+id/editText4"
android:layout_marginTop="10dp"
android:text="Alamat" />

<EditText android:id="@+id/editText5"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView5"
android:layout_below="@+id/textView5" />

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText5"
android:layout_alignParentBottom="true"
android:text="Update" />

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button1"
android:layout_alignBottom="@+id/button1"
android:layout_toRightOf="@+id/textView4"
android:text="Back" />

</RelativeLayout>
```

3. Running The Project Application

After completion of all, implementation of the code, can now try to run the application in the Android's Studio.

The result looks like this:



Figure 7.1 Result Display apps

If the user (users) suppressing the + Create a new Entry, it will appear as shown below:



Figure 7.2 Result Display Input Form

If the user (users) suppressing one of the name, it will pop up the options in the form of a Context Menu, which consists of whether the user (users) want to view, update, or remove personal information. As in the picture below:

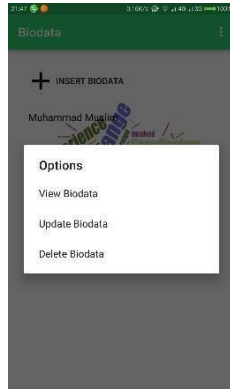


Figure 7.3 Result Display Options

MODUL 8

Database Real Time Using Firebase

Firestore is a cloud service provider's Backend and also as a Service (BaaS), a company based in San Francisco, California. The company makes a product that helps the Software Developer in building Mobile or Web Applications. The founders of this company were Andrew Lee and James Tamplin. Firestore main products have a Realtime Database that is available in the form of API, which lets Developers in storing and synchronizing data across multiple Client. Now the company has in its acquisition by Google in October 2014. Features that have now been developed and in snap in tools. This is Firestore tools which help the work of an Android developer, in developing Android applications, in which case the backend services. Firestore is free, and you can add it on Android applications you guys, on a tutorial that has been provided by the Team's Firestore (read: how to Setup). Firestore also available not only for Android, but it could also be for iOS and Web Applications.



Figure 8.1 Logo Firestore

There are many features in the Firestore in the Tools help in building the backend service, as well as help in terms of the analysis of the target user (users) to memonetize(Admob). and also build up a user base in terms of promoting applications (Adwords).

Here are a few features that are in the Firestore:

Develop :

- Cloud Messaging
- Authentication
- Realtime Database
- Storage
- Hosting
- Test Lab
- Crash Reporting

Grow :

- Notification
- Remote Config
- App Indexing
- Dynamic Links
- Invites
- Adwords

Earn:

- Admob

More information if you would like to see the features on offer at Firebase:

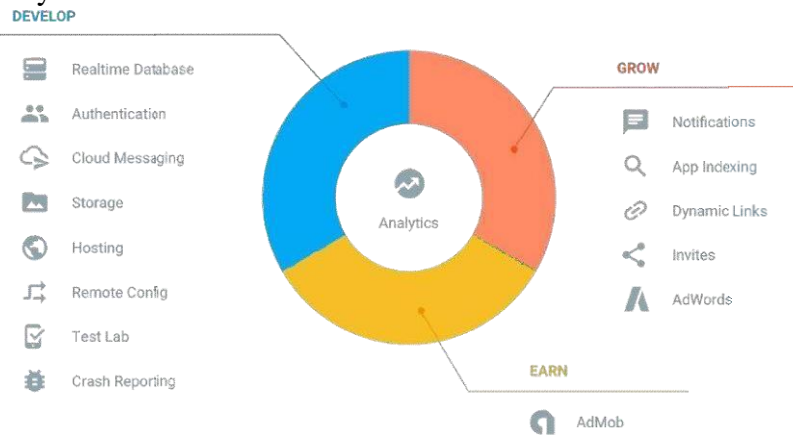


Figure 8.2 Some of the features that are on Firebase (by Google Firebase)

If you want to use the latest version of Android's Studio. Can see these Tools on the Main Menu.

Main Menu: Tools > Firebase

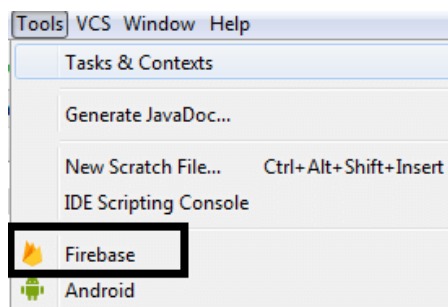


Figure 8.3 Tools Firebase Android Studio

Then on the Tabs right Assistants, can be seen the features provided.

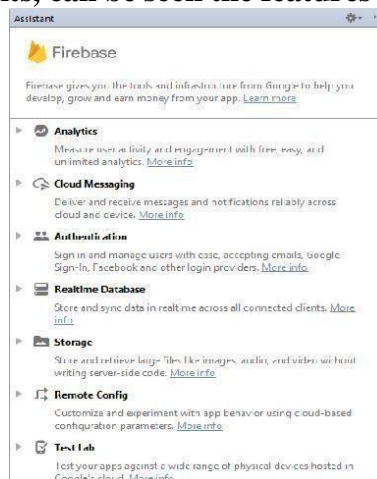


Figure 8.4 Tab Assistants

Well now you already know right what is a Firebase, this all depends on the needs of the application are you guys develop. If the application wants to require a Backend Service as a Service (BaaS) and also data analysis users (users). could use the Tools Firebase.

Link :

<https://firebase.google.com/features/>