

BAB 3. PENDALAMAN TEKNIK PEMROGRAMAN DI ANDROID STUDIO

Daftar Isi

- A. Operasi File System**
- B. Aplikasi Login**
- C. Aplikasi Input ke Database MySQL**
- D. Aplikasi Memilih Image dari Storage**
- E. Aplikasi Pengambil Foto dari Kamera**
- F. Aplikasi Pengambil Video**
- G. Aplikasi Perekam Suara**
- H. Geolocation dan Google Map**
- I. Pengambilan Data Json dari Internet untuk Ditampilkan di Layout**
- J. Aplikasi Menggunakan Webview**
- K. Servis di Android**
- L. Broadcast Receiver di Android**
- M. Bluetooth di Android**

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

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

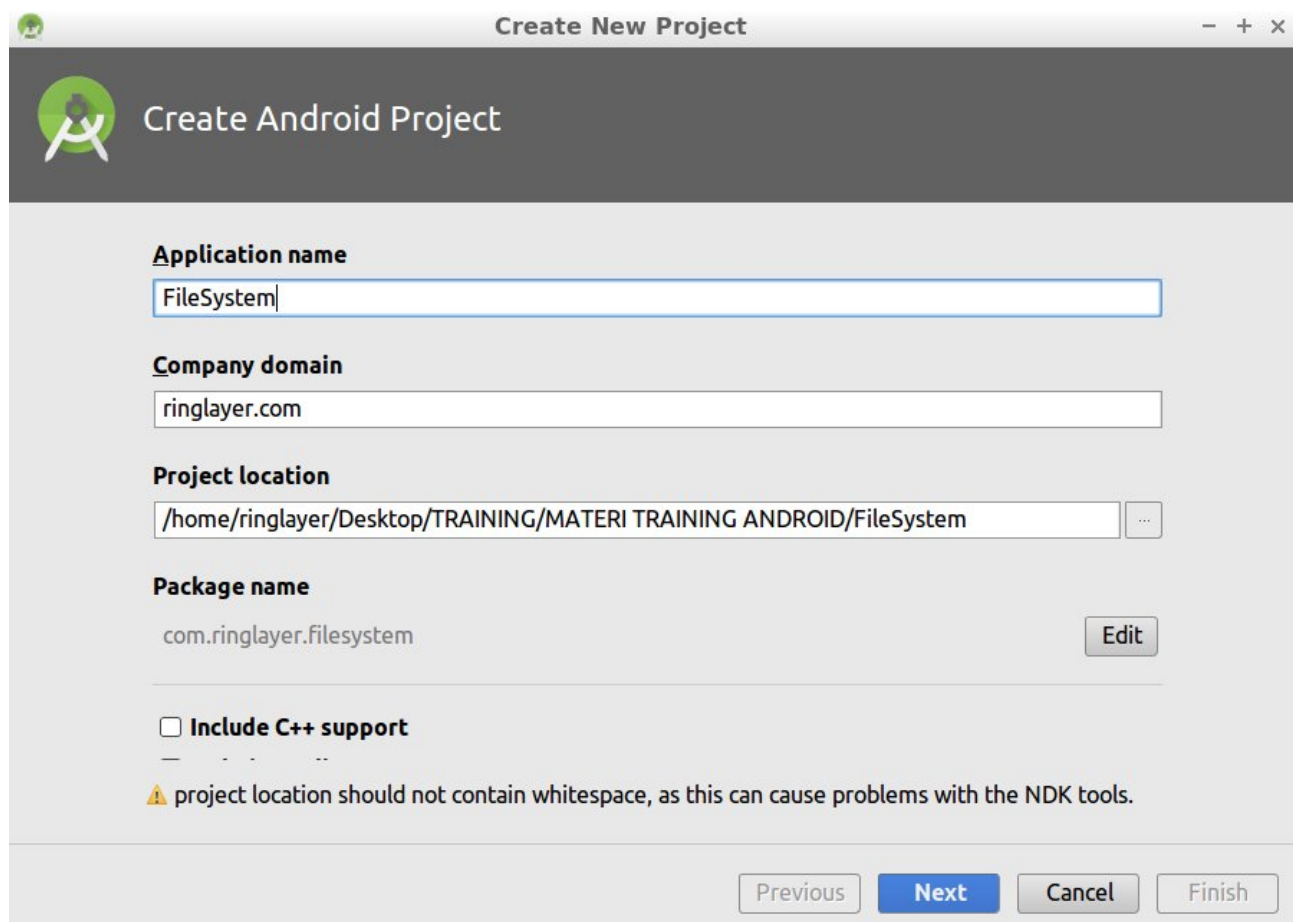
BAB 3. PENDAHULUAN TEKNIK PEMROGRAMAN DI ANDROID STUDIO

A. Operasi File System

Pengetahuan dasar pemrograman file system pada device android merupakan pengetahuan fundamental untuk pengembangan berbagai macam aplikasi.

Sistem file (*file system*) atau sistem berkas merupakan struktur logika yang digunakan untuk mengendalikan akses terhadap data yang ada pada disk. Dengan kata lain, sistem file merupakan database khusus untuk penyimpanan, pengelolaan, manipulasi dan pengambilan data, agar mudah ditemukan dan diakses.

Pada contoh kali ini kita akan membuat aplikasi untuk mengelola file system pada device android. Buat proyek baru dengan nama FileSystem



Create New Project

Create Android Project


Application name
FileSystem

Company domain
ringlayer.com

Project location
/home/ringlayer/Desktop/TRAINING/MATERI TRAINING ANDROID/FileSystem

Package name
com.ringlayer.filesystem Edit

☐ **Include C++ support**

 project location should not contain whitespace, as this can cause problems with the NDK tools.

Previous Next Cancel Finish

Cara Kerja Aplikasi

Aplikasi ini ditujukan untuk menguji operasi file system pada android. Pertama tama aplikasi akan merequest permisi untuk baca tulis file system (pada device android marshmallow ke atas).

Selanjutnya aplikasi akan membuat direktori FileSystem. Jika berhasil, Aplikasi akan membuat file teks log.txt di /sdcard/FileSystem/log.txt. Jika berhasil, aplikasi akan menulisi file dengan teks : "overwrite test" dan "append test"

Bagian XML

Pertama tama agar bisa melakukan operasi baca tulis pada device android, maka tambahkan 2 permissi pada AndroidManifest.xml

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Berikut ini adalah isi lengkap AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.filesystem">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Berikut ini adalah isi file MainActivity.xml

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

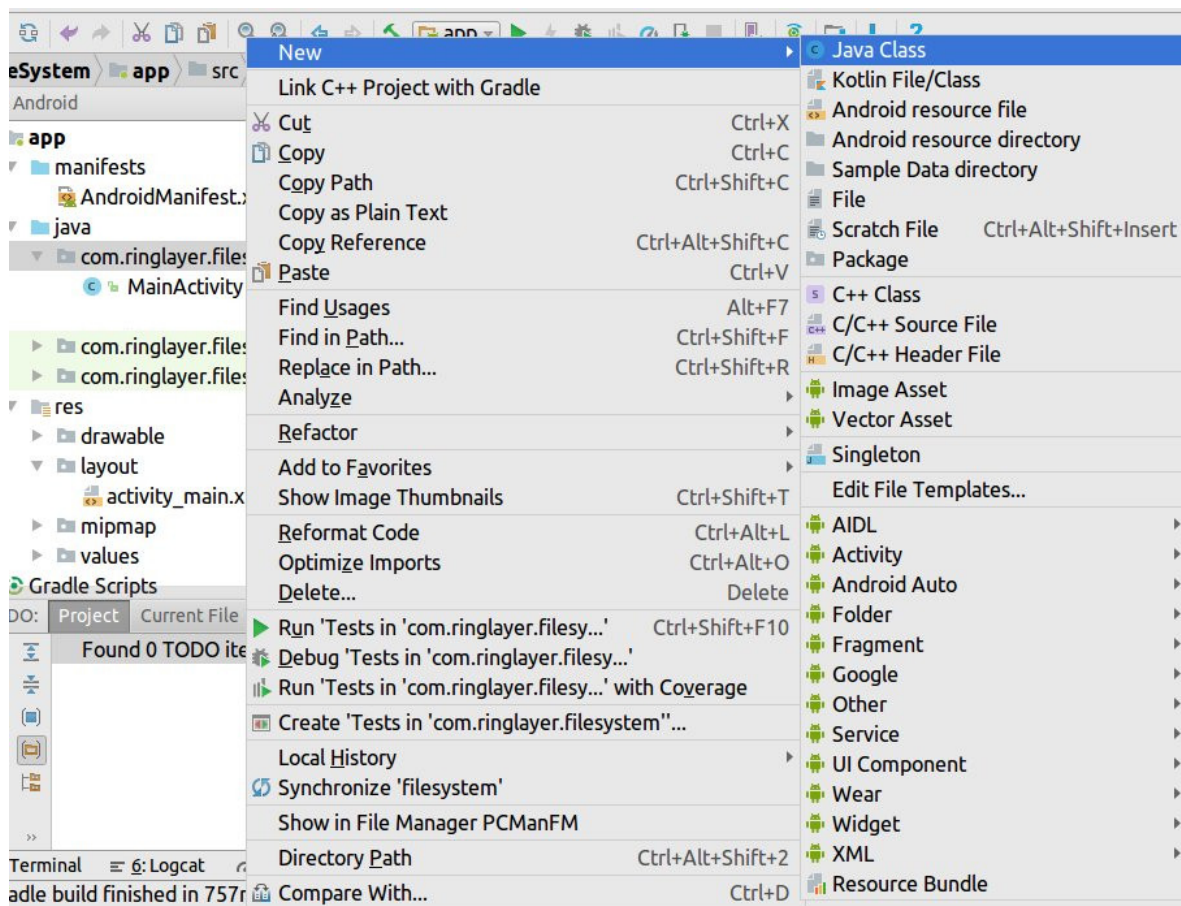
</android.support.constraint.ConstraintLayout>
```

Bagian Source Code

Pada aplikasi kali ini, kita akan menggunakan 1 activity : MainActivity dengan layoutnya activity_main.xml dan 1 kelas yang bernama Util.

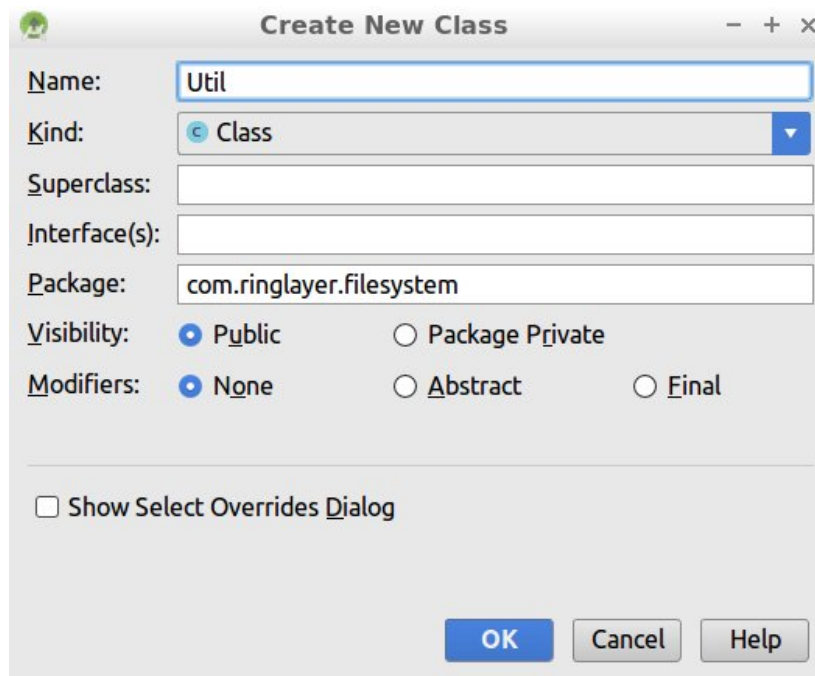
Pada android dengan versi 6 ke atas memerlukan notif request permission ke user agar aplikasi mendapatkan permisi sesuai AndroidManifest. Oleh karena itu, kita siapkan suatu kelas baru dengan nama “Util” yang berguna untuk melakukan request permission ke user untuk permisi yang dibutuhkan aplikasi.

Pertama tama buat kelas baru dengan nama Util



Caranya adalah klik kanan path nama proyek Anda di sub direktori java (perhatikan gambar di atas). Setelah itu, klik kanan lalu pilih New → Java Class

Beri nama Util



Klik ok. Berikut ini isi kelas Util.java:

```
package com.ringlayer.filesystem;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
/**
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        public void askForPermission(String permission, Integer requestCode) {
            if (ContextCompat.checkSelfPermission(mContext, permission) !=
                PackageManager.PERMISSION_GRANTED) {
                if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                    permission)) {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                } else {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                }
            }
        }
    }
}
```

Penjelasan

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```
if (ContextCompat.checkSelfPermission(mContext, permission) !=  
PackageManager.PERMISSION_GRANTED) {
```

Pada rutin di atas, jika permissi bukanlah `PERMISSION_GRANTED` , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```
ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);
```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Berikut ini isi MainActivity:

```
package com.ringlayer.filesystem;  
import android.Manifest;  
import android.os.Build;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.io.PrintWriter;  
import java.util.ArrayList;  
import java.util.List;  
import static android.Manifest.permission.CAMERA;  
public class MainActivity extends AppCompatActivity {  
    public Util util = new Util(MainActivity.this);  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
            util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);  
            util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);  
        }  
        buat_dir("/sdcard/FileSystem");  
        boolean ada = CheckIfDirectoryExists("/sdcard/FileSystem");  
        if (ada == true) {  
            createFileInSDCard("/sdcard/FileSystem/log.txt");  
            if (check_file_exists("/sdcard/FileSystem/log.txt") == 1) {  
                write("/sdcard/FileSystem/log.txt", "overwrite test");  
                append_file("/sdcard/FileSystem/log.txt", "append test");  
            }  
        }  
    }  
}
```

```

}
public int buat_dir(String Dirname) {
    int retme = 0;
    try {
        boolean exists = false;
        exists = CheckIfDirectoryExists(Dirname);
        if (exists == false) {
            new File(Dirname).mkdirs();
        }
    }
    catch (Exception e) {
        Log.e("error",e.toString());
    }
    return retme;
}
public void createFileInSDCard(String path) {
    try {
        int exists = check_file_exists(path);
        if (exists == 0) {
            File file = new File(path);
            file.createNewFile();
        }
    }
    catch (Exception e) {
        Log.e("error",e.toString());
    }
}
public boolean CheckIfDirectoryExists(String path) {
    boolean exists = false;
    try {
        File dir = new File(path);
        exists = dir.exists();
        return exists;
    }
    catch (Exception e) {
        Log.e("error",e.toString());
        return false;
    }
}
public boolean deleteDirectory(File path) {
    boolean delres = true;
    try {
        if (path.exists()) {
            File[] files = path.listFiles();
            for (int i = 0; i < files.length; i++) {
                if (files[i].isDirectory()) {
                    deleteDirectory(files[i]);
                } else {
                    files[i].delete();
                }
            }
        }
        delres = path.delete();
    }
    catch (Exception e) {
        Log.e("error",e.toString());
    }
    return delres;
}
public void copyFile(File sourceLocation, File targetLocation) {
    try {
        InputStream in = new FileInputStream(sourceLocation);
        OutputStream out = new FileOutputStream(targetLocation);
        // Copy the bits from instream to outstream
        byte[] buf = new byte[1024];
        int len;
        while ((len = in.read(buf)) > 0) {
            out.write(buf, 0, len);
        }
    }
}

```

```

        in.close();
        out.close();
    }
    catch (Exception e) {
        Log.e("error",e.toString());
    }
}

public void copyDirectory(File sourceLocation, File targetLocation) {
    try {
        if (sourceLocation.isDirectory()) {
            if (!targetLocation.exists()) {
                targetLocation.mkdirs();
            }
            String[] children = sourceLocation.list();
            for (int i = 0; i < children.length; i++) {
                copyDirectory(new File(sourceLocation, children[i]), new File(
                    targetLocation, children[i]));
            }
        } else {
            copyFile(sourceLocation, targetLocation);
        }
    }
    catch (Exception e) {
        Log.e("error",e.toString());
    }
}

public void append_file(String path, String content)
{
    try {
        File file = new File(path);
        if (!file.exists()) {
            file.createNewFile();
        }
        FileWriter fw = new FileWriter(file,true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter pw = new PrintWriter(bw);
        pw.println(content);
        pw.close();
    } catch(Exception e) {
        Log.e("error",e.toString());
    }
}

public String read(String fname){
    BufferedReader br = null;
    String line = null;
    try {
        br = new BufferedReader(new FileReader(fname));
        line = br.readLine();
        if (line == null) {
            line = "";
        }
    } catch(IOException e) {
        Log.e("error",e.toString());
        return null;
    }
    return line;
}

public Boolean write(String fname, String fcontent){
    try {
        String fpath = fname;
        File file = new File(fpath);
        // If file does not exists, then create it
        if (!file.exists()) {
            file.createNewFile();
        }
        FileWriter fw = new FileWriter(file.getAbsolutePath());
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write(fcontent);
        bw.close();
    }
}

```



```

        Log.d("Sucesss", "Sucesss");
        return true;
    } catch (IOException e) {
        Log.e("error", e.toString());
        return false;
    }
}
public int check_file_exists(String full_path) {
    int retme = 0;
    try {
        File file = new File(full_path);
        if (file.exists()) {
            retme = 1;
        }
    }
    catch (Exception e) {
        Log.e("error", e.toString());
    }
    return retme;
}
}

```

Pada bagian source code kali ini terdapat beberapa metode yang intinya adalah operasi file system :

1. Metode untuk Membuat Direktori Baru

```

public int buat_dir(String Dirname) {
    int retme = 0;
    try {
        boolean exists = false;
        exists = CheckIfDirectoryExists(Dirname);
        if (exists == false) {
            new File(Dirname).mkdirs();
        }
    }
    catch (Exception e) {
        Log.e("error", e.toString());
    }
    return retme;
}

```

Pada rutin di atas kode ini digunakan untuk membuat direktori baru :

```
new File(Dirname).mkdirs();
```

2. Membuat file baru

```

public void createFileInSDCard(String path) {
    try {
        int exists = check_file_exists(path);
        if (exists == 0) {
            File file = new File(path);
            file.createNewFile();
        }
    }
    catch (Exception e) {
        Log.e("error", e.toString());
    }
}

```

Pada source di atas, memiliki parameter dengan tipe String dengan nama variabel path yang merupakan path lengkap file yang ingin dibuat beserta nama file.

Berikut ini contoh path :

```
/sdcard/log.txt  
/sdcard/data.txt
```

Perhatikan ! Karena android merupakan keluarga linux maka untuk memisahkan direktori dan nama file menggunakan tanda slash bukan back slash seperti pada windows.

Rutin di bawah ini digunakan untuk membuat file :

```
File file = new File(path);  
file.createNewFile();
```

3. Mengecek Keberadaan Direktori

```
public boolean CheckIfDirectoryExists(String path) {  
    boolean exists = false;  
    try {  
        File dir = new File(path);  
        exists = dir.exists();  
        return exists;  
    }  
    catch (Exception e) {  
        Log.e("error",e.toString());  
        return false;  
    }  
}
```

Metode di atas akan mengecek keberadaan direktori :

```
exists = dir.exists();
```

di mana exists merupakan boolean logic true atau false , jika direktori ada maka akan berisi true sedangkan jika tidak ada maka akan berisi false

4. Menghapus Direktori

```
public boolean deleteDirectory(File path) {  
    boolean delres = true;  
    try {  
        if (path.exists()) {  
            File[] files = path.listFiles();  
            for (int i = 0; i < files.length; i++) {  
                if (files[i].isDirectory()) {  
                    deleteDirectory(files[i]);  
                } else {  
                    files[i].delete();  
                }  
            }  
        }  
        delres = path.delete();  
    }  
    catch (Exception e) {
```

```

        Log.e("error",e.toString());
    }
    return delres;
}

```

Pada contoh di atas, metode menghapus direktori tidak digunakan, jika anda ingin mencoba bisa ditambahkan sendiri ke MainActivity di mana parameter path merupakan path lengkap direktori yang ingin dihapus

5. Mengopi File

```

public void copyFile(File sourceLocation, File targetLocation) {
    try {
        InputStream in = new FileInputStream(sourceLocation);
        OutputStream out = new FileOutputStream(targetLocation);
        // Copy the bits from instream to outstream
        byte[] buf = new byte[1024];
        int len;
        while ((len = in.read(buf)) > 0) {
            out.write(buf, 0, len);
        }
        in.close();
        out.close();
    }
    catch (Exception e) {
        Log.e("error",e.toString());
    }
}

```

Pada source di atas pertama tama kita membaca isi keseluruhan file dan disimpan di dalam Input Stream in , Selanjutnya Output Stream untuk nama file dan path hasil kopi, disiapkan potongan kode ini:

```
OutputStream out = new FileOutputStream(targetLocation);
```

Potongan kode ini akan membaca isi file yang sebelumnya disimpan di input stream untuk kemudian ditulisi ke output stream

```

byte[] buf = new byte[1024];
int len;
while ((len = in.read(buf)) > 0) {
    out.write(buf, 0, len);
}

```

6. Mengkopi Direktori

```

public void copyDirectory(File sourceLocation, File targetLocation) {
    try {
        if (sourceLocation.isDirectory()) {
            if (!targetLocation.exists()) {
                targetLocation.mkdirs();
            }
            String[] children = sourceLocation.list();
            for (int i = 0; i < children.length; i++) {
                copyDirectory(new File(sourceLocation, children[i]), new File(
                    targetLocation, children[i]));
            }
        }
    }
}

```

```

    }
    } else {
        copyFile(sourceLocation, targetLocation);
    }
}
catch (Exception e) {
    Log.e("error",e.toString());
}
}

```

Pada source di atas rutin yang mengkopi direktori terdapat pada baris ini :

```

copyDirectory(new File(sourceLocation, children[i]), new File(
    targetLocation, children[i]));

```

alternatif dari metode ini adalah

```

CopyFile(sourceLocation, targetLocation);

```

yang bisa juga digunakan untuk mengkopi direktori

di mana parameter pertama berisi source direktori, kemudian parameter kedua berisi target direktori.

7. Menambah Isi File

```

public void append_file(String path, String content)
{
    try {
        File file = new File(path);
        if (!file.exists()) {
            file.createNewFile();
        }
        FileWriter fw = new FileWriter(file,true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter pw = new PrintWriter(bw);
        pw.println(content);
        pw.close();
    } catch (Exception e) {
        Log.e("error",e.toString());
    }
}

```

Pada source di atas pertama tama kita siapkan objek file writer yang berguna untuk menulis file yang ingin kita tambah isinya. Selanjutnya kita siapkan objek buffered writer. Objek buffered writer berguna agar penulisan file menjadi efisien. Perbedaan antara FileWriter dan BufferedWriter adalah BufferedWriter akan menulis potongan potongan data file lebih besar dari FileWriter. BufferedWriter di sini merupakan file descriptor untuk file yang akan kita tambahkan isinya.

Pada tahap akhir kita siapkan objek akhir yang berisi metode metode untuk penulisan file yaitu PrintWriter yang parameternya kita isi dengan file descriptor yang telah kita siapkan sebelumnya yaitu berupa objek BufferedWriter

8. Membaca File

```
public String read(String fname){
    BufferedReader br = null;
    String line = null;
    try {
        br = new BufferedReader(new FileReader(fname));
        line = br.readLine();
        if (line == null) {
            line = "";
        }
    } catch (IOException e) {
        Log.e("error",e.toString());
        return null;
    }
    return line;
}
```

Pada source di atas, baris untuk membaca isi file terdapat pada baris ini:

```
line = br.readLine();
```

Variabel line akan berisi hasil string pembacaan isi file.

9. Menulis File

```
public Boolean write(String fname, String fcontent){
    try {
        String fpath = fname;
        File file = new File(fpath);
        // If file does not exists, then create it
        if (!file.exists()) {
            file.createNewFile();
        }
        FileWriter fw = new FileWriter(file.getAbsolutePath());
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write(fcontent);
        bw.close();
        Log.d("Sucess", "Sucess");
        return true;
    } catch (IOException e) {
        Log.e("error",e.toString());
        return false;
    }
}
```

Pada source di atas, baris untuk menulis file terdapat pada baris ini:

```
bw.write(fcontent);
```

10. Mengecek Keberadaan File

```
public int check_file_exists(String full_path) {
    int retme = 0;
    try {
        File file = new File(full_path);
        if(file.exists()) {
```

```

        retme = 1;
    }
}
catch (Exception e) {
    Log.e("error",e.toString());
}
return retme;
}
}

```

Pada source di atas, rutin di atas pertama tama akan menyiapkan objek file deskriptor untuk file yang akan kita cek. Pada tahap selanjutnya kita panggil metode exists() yang terdapat ada objek file tadi untuk mengecek keberadaan file. Hasilnya berupa boolean true atau false. Jika file ada maka menghasilkan nilai boolean true sebaliknya menghasilkan nilai boolean false

B. Aplikasi Login

Pada contoh kali ini, kita akan membuat aplikasi login.

Cara Kerja Aplikasi

Sebelum memulai membuat aplikasi android, kita perlu memahami struktur aplikasi login ini. Aplikasi login di android ini bekerja dengan mengirimkan http post yang berisi username dan password ke API berupa aplikasi web yang sudah kita siapkan sebelumnya. Selanjutnya aplikasi android akan mengecek response dari aplikasi web, jika response berupa login valid maka user akan diizinkan mengakses activity yang telah kita tentukan, jika tidak maka akan muncul pesan toast login error.

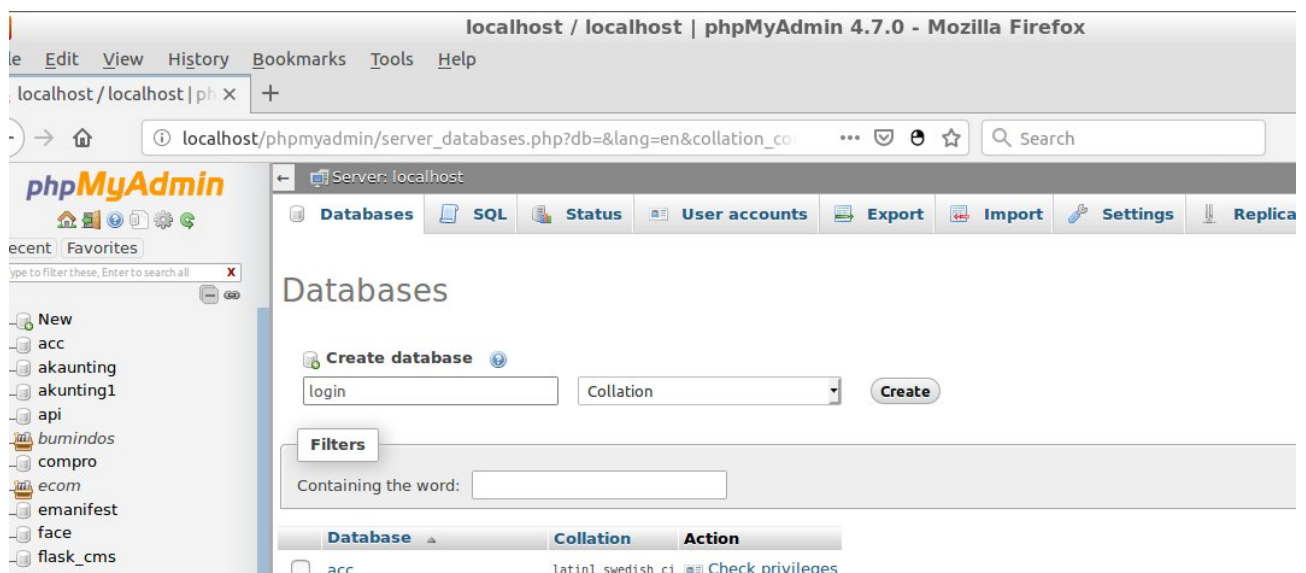
Aplikasi android ini pada saat pertama kali MainActivity berjalan akan melakukan register broadcast receiver yang gunanya nanti akan menangkap pesan hasil login dari server, di mana broadcast receiver akan dipasang di onStart().

Langkah selanjutnya aplikasi akan meregister onClickListener pada tombol login yang fungsinya akan ditrigger saat user mengklik tombol login. Pada saat user mengklik tombol login maka aplikasi ini akan menjalankan servis di belakang layar untuk mengirimkan inputan login dan password berupa plain text data.

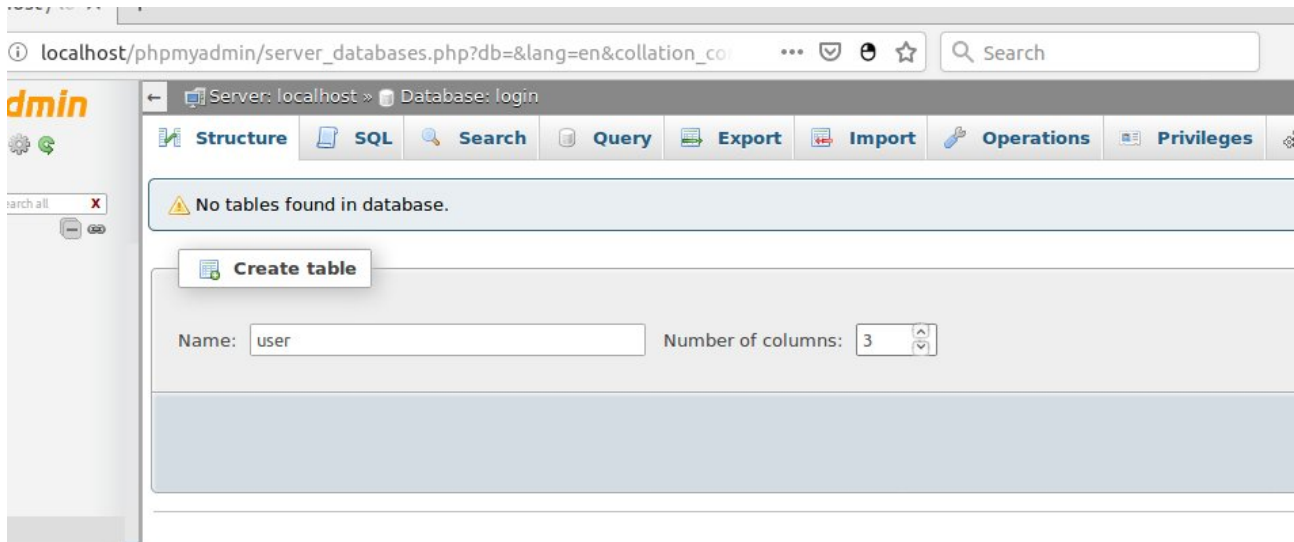
Setelah servis menangkap hasil http response dari server maka langkah selanjutnya, servis akan melakukan broadcast yang mana selanjutnya broadcast ini akan ditangkap activity MainActivity yang sebelumnya telah melakukan register Broadcast Receiver untuk menangkap pesan broadcast hasil login.

Jika login gagal maka akan muncul toast pesan login gagal. Jika login berhasil maka MainActivity akan menjalankan activity kedua bernama MemberArea.

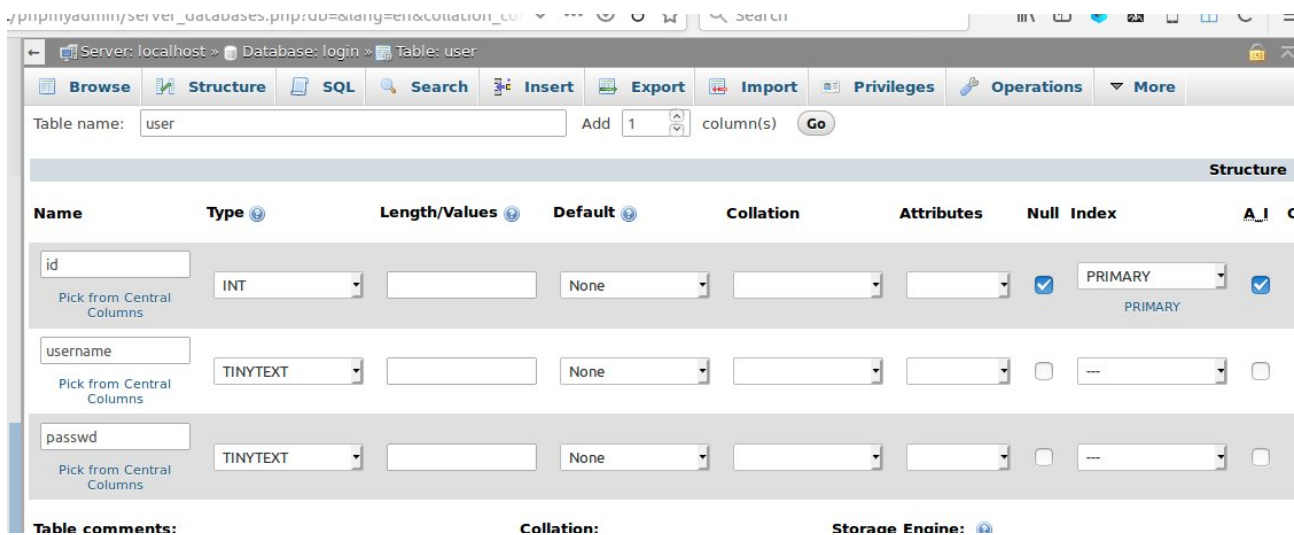
Pertama tama kita siapkan terlebih dahulu database mysql pada server yang akan diakses aplikasi android, buat database mysql dengan nama login melalui phpmyadmin:



Setelah itu siapkan tabel dengan nama user dengan jumlah 3 kolom :



Berikut ini struktur database user:



Struktur di atas menggunakan 3 kolom : id, username dan passwd.

Perhatikan ! Pada contoh di atas, centang Null, pilih index : Primary, centang AI. Setelah selesai, klik “save”.

Selanjutnya masukkan sql ini :

```
INSERT INTO `user` (`id`, `username`, `passwd`) VALUES (NULL, 'admin', '21232f297a57a5a743894a0e4a801fc3');
```

perintah sql di atas akan mengisi tabel user dengan user admin dan password admin yang sudah dienkripsi dengan md5

Setelah database siap, selanjutnya kita siapkan aplikasi web yang akan kita gunakan. Aplikasi web yang akan kita simpan di c:\xampp\htdocs\login

Sebelumnya simpan file ini https://github.com/ringlayer/simple-php-to-prevent-basic-web-attack/blob/master/get_and_post.php ke direktori c:\xampp\htdocs\login

Siapkan file login.php dengan isi berikut ini :

```
<?php
require_once("get_and_post.php");
$user = trim($_POST['user']);
$pass = trim($_POST['pass']);

$con = mysqli_connect("localhost","root","","login");

$password = md5($pass);

if ($con) {
    $sql = "select * from `user` where `username` like '$user' and `password` like '$password'";
    $result = mysqli_query($con, $sql);

    if (mysqli_num_rows($result) > 0) {
        print "success login";
    }
    else {
        print "failed login";
    }
}
else {
    print "failed login";
}
?>
```

Source code di atas adalah source code php yang berguna sebagai API, digunakan menangkap data username dan password yang dipost untuk kemudian dicari di tabel user apakah ada username dan password yang dipost di tabel. Jika ditemukan, kirimkan response sukses.

Selanjutnya, buat proyek baru dengan nama AplikasiLogin.

Bagian XML

Buka res → values, buka file strings.xml. Pastekan variable ini :

```
<resources>
    <string name="app_name">AplikasiLogin</string>
    <string name="memberarea">Member Area</string>
    <string name="url">http://192.168.43.140/login/login.php</string>
</resources>
```

Selanjutnya buat activity baru dengan nama MemberArea. Berikut ini isi file activity_member_area.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```

        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:id="@+id/background"
        android:gravity="center"
        tools:context="com.ringlayer.aplikasilogin.MemberArea">
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
<TextView
    android:id="@+id/desc"
    android:background="@color/colorAccent"
    android:text="@string/memberarea"
    android:textColor="#ffffff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingRight="10dp"
    android:paddingLeft="10dp"
/>
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
</LinearLayout>

```

Berikut ini isi activity_main.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.aplikasilogin.MainActivity">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/background"
    android:gravity="center">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="400dp"
    android:layout_height="60dp"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="400dp"
    android:layout_height="150dp"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    >
<TextView
    android:layout_width="match_parent"
    android:layout_height="5dp" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="10dp" />
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="300dp"
    android:layout_height="50dp"
    android:layout_gravity="center"

```

```

        android:orientation="horizontal">
        <EditText
            android:id="@+id/username"
            android:layout_width="250dp"
            android:layout_height="50dp"
            android:layout_gravity="center"
            android:inputType="textPersonName"
            android:hint="Username"
            android:paddingLeft="10dp"
            android:textColor="@color/colorAccent" />
    </LinearLayout>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="5dp" />
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="300dp"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:orientation="horizontal">
        <EditText
            android:paddingLeft="10dp"
            android:id="@+id/password"
            android:layout_width="250dp"
            android:layout_height="50dp"
            android:layout_gravity="center"
            android:inputType="textPassword"
            android:textColor="@color/colorAccent"
            android:hint="Password" />
    </LinearLayout>
</LinearLayout>
<TextView
    android:layout_width="match_parent"
    android:layout_height="10dp" />
<Button
    android:id="@+id/tombol_login"
    android:layout_width="300dp"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:textColor="@color/colorAccent"
    android:text="Login" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="10dp" />
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

Sebelum mengedit AndroidManifest.xml, ada baiknya membuat terlebih dahulu servis dengan nama ServisLogin karena akan kita tambahkan pada AndroidManifest.xml

Karena kita akan mengirimkan http post ke server, maka aplikasi android ini memerlukan permisi untuk internet, tambahkan baris berikut ini pada AndroidManifest.xml:

```

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

```

Berikut ini adalah isi file AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.aplikasilogin">
    <uses-permission android:name="android.permission.INTERNET" />
    <application

```

```

        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <service android:name=".ServisLogin" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MemberArea"></activity>
    </application>
</manifest>

```

Bagian Source Code

Selanjutnya kita akan menggunakan kelas Util.java yang sebelumnya kita pakai. Buat kelas baru dengan nama “Util” . Berikut ini isi Util.java :

```

package com.ringlayer.aplikasilogin;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
public class Util {
    Activity mContext;
    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);}
    }
    public void askForPermission(String permission, Integer requestCode) {
        if (ContextCompat.checkSelfPermission(mContext, permission) !=
        PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
            permission)) {
                ActivityCompat.requestPermissions(mContext, new String[]{permission},
            requestCode);
            } else {
                ActivityCompat.requestPermissions(mContext, new String[]{permission},
            requestCode);
            }
        }
    }
}

```

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```
if (ContextCompat.checkSelfPermission(mContext, permission) !=  
PackageManager.PERMISSION_GRANTED) {
```

Pada rutin di atas, jika permissi bukanlah `PERMISSION_GRANTED` , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```
ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);
```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Selanjutnya kita memerlukan java class untuk mengirimkan http request. Buat java class baru dengan nama “SendPostRequest”

Berikut ini isi file SendPostRequest.java:

```
package com.ringlayer.aplikasilogin;  
  
import android.os.StrictMode;  
import android.util.Log;  
import org.json.JSONObject;  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.net.URLEncoder;  
import java.util.Arrays;  
import java.util.Iterator;  
import java.util.List;  
/**  
 * Created by ringlayer on 23/06/18.  
 *  
 * modified from https://www.studytutorial.in/android-httpurlconnection-post-and-get-request-tutorial  
 */  
public class SendPostRequest {  
    public String _execute(String url_post,String[] key_str, String[] val_str) {  
        try {  
            if (android.os.Build.VERSION.SDK_INT > 9) {  
                try {  
                    StrictMode.ThreadPolicy policy = new  
StrictMode.ThreadPolicy.Builder().permitAll().build();  
                    StrictMode.setThreadPolicy(policy);  
                }  
            }  
            catch (Exception e) {  
                String stackTrace = Log.getStackTraceString(e);  
                Log.e("[-] Strictmode", stackTrace);  
            }  
        }  
    }  
}
```

```

    }
    //StrictMode.ThreadPolicy policy = new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    //StrictMode.setThreadPolicy(policy);
    URL url = new URL(url_post);
    int i;
    JSONObject postDataParams = new JSONObject();
    for (i = 0; i < key_str.length; i++) {
        postDataParams.put(key_str[i], val_str[i]);
    }
    Log.e("params", postDataParams.toString());
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(15000 /* milliseconds */);
    conn.setConnectTimeout(15000 /* milliseconds */);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);
    conn.connect();
    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, "UTF-
8"));
    writer.write(getPostDataString(postDataParams));
    writer.flush();
    writer.close();
    os.close();
    InputStreamReader in = new InputStreamReader((InputStream)
conn.getContent());
    BufferedReader buff = new BufferedReader(in);
    String line = "";
    StringBuffer text = new StringBuffer();
    while (line != null) {
        line = buff.readLine();
        text.append(line + "\n");
    }
    Log.e("res:", text.toString());
    String res = text.toString();
    return res;
}
catch (Exception e){
    return new String("Exception: " + e.getMessage());
}
}
/*
taken from https://www.studytutorial.in/android-httpurlconnection-post-and-get-
request-tutorial
*/
public String getPostDataString(JSONObject params) throws Exception {
    StringBuilder result = new StringBuilder();
    boolean first = true;
    Iterator<String> itr = params.keys();
    while(itr.hasNext()){
        String key= itr.next();
        Object value = params.get(key);
        if (first)
            first = false;
        else
            result.append("&");
        result.append(URLEncoder.encode(key, "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(value.toString(), "UTF-8"));
    }
    return result.toString();
}
}
}

```

Pada kelas di atas kita melakukan non blocking dengan memanfaatkan fitur kelas StrictMode untuk mencegah blocking pada User Interface yang sedang aktif saat pengiriman post request ke server

```
StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
```

Selanjutnya kita membuat koneksi http ke server melalui rutin di bawah ini :

```
URL url = new URL(url_post);
    int i;
    JSONObject postDataParams = new JSONObject();
    for (i = 0; i < key_str.length; i++) {
        postDataParams.put(key_str[i], val_str[i]);
    }
    Log.e("params", postDataParams.toString());
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(15000 /* milliseconds */);
    conn.setConnectTimeout(15000 /* milliseconds */);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);
    conn.connect();
```

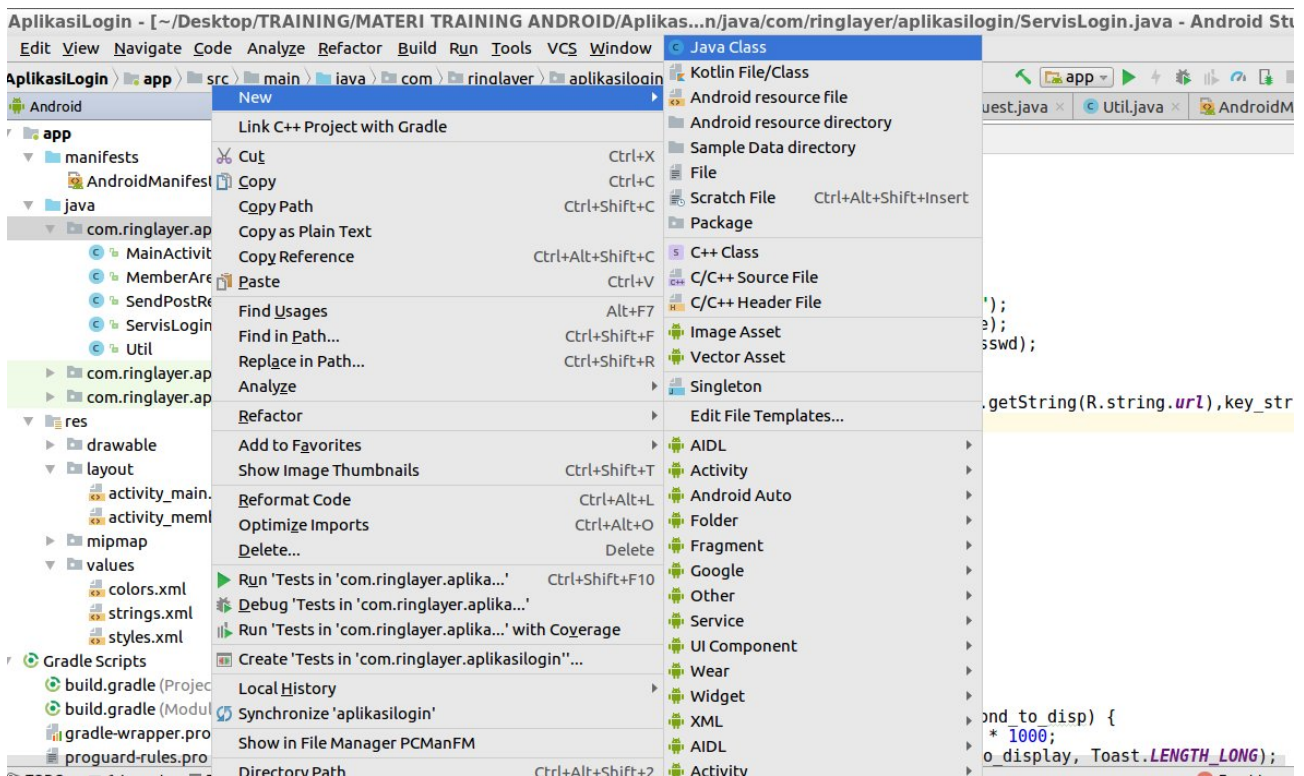
Rutin di atas berguna mengatur properti untuk koneksi http berupa read timeout, connect timeout, jenis request yaitu POST dan parameter post yang akan dikirim ke server. Setelah semuanya siap koneksi dan pengiriman data ke server dieksekusi pada baris `conn.connect()`;

Rutin selanjutnya berguna untuk mempersiapkan buffer yang akan digunakan untuk menangkap data response balasan dari server

```
BufferedReader buff = new BufferedReader(in);
    String line = "";
    StringBuffer text = new StringBuffer();
    while (line != null) {
        line = buff.readLine();
        text.append(line + "\n");
    }
    Log.e("res:", text.toString());
    String res = text.toString();
```

Pada contoh di atas, balasan dari server pada akhirnya akan disimpan di variabel bertipe String dengan nama res. Di mana sebelumnya response dari server ditangkap di variabel objek StringBuffer text kemudian pada bagian baris `"String res = text.toString();"` objek buffer dikonversi menjadi String.

Jika sebelumnya, ServisLogin belum dibuat, buat servis baru dengan nama ServisLogin, dengan cara klik kanan pada folder proyek di jendela kiri. Pada kesempatan kali ini kita akan memilih class saja agar template source code kosong. Perhatikan gambar di bawah ini :



Berikut ini isi ServisLogin.java :

```
package com.ringlayer.aplikasilogin;
import android.app.IntentService;
import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.util.Log;
import android.view.Gravity;
import android.widget.Toast;
/**
 * Created by ringlayer on 18/03/19.
 */
public class ServisLogin extends IntentService {
    SendPostRequest spr = new SendPostRequest();
    private Toast mToastToShow;
    public ServisLogin() {super("ServisLogin");}
    @Override
    protected void onHandleIntent(Intent intent) {
        try {
            Log.e("ohi", "login proc");
            String pesan = "";
            Bundle b = intent.getExtras();
            if (b != null) {
                String uname = (String) b.get("login");
                String passwd = (String) b.get("password");
                Log.e("login", "login : " + uname);
                Log.e("passwd", "passwd : " + passwd);
                String[] key_str = {"user", "pass"};
                String[] val_str = {uname, passwd};
                String res = spr._execute(getResources().getString(R.string.url), key_str,
                val_str);

                if (res.indexOf("success") != -1) {
                    pesan = "berhasil";
                }
                else {

```



```

        pesan = "gagal";
    }
    publishResults(pesan);
}
}
catch (Exception e) {
    Log.e("error", e.toString());
}
}
public void showToast(String msg_to_display, int second_to_disp) {
    int toastDurationInMilliseconds = second_to_disp * 1000;
    mToastToShow = Toast.makeText(this, msg_to_display, Toast.LENGTH_LONG);
    mToastToShow.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0,
0);
    CountDownTimer toastCountDown;
    toastCountDown = new CountDownTimer(toastDurationInMilliseconds, 1000 /*Tick
duration*/) {
        public void onTick(long millisUntilFinished) {
            mToastToShow.show();
        }
        public void onFinish() {
            mToastToShow.cancel();
        }
    };
    mToastToShow.show();
    toastCountDown.start();
}
private void publishResults(String pesan) {
    try {
        Log.e("[+] publishResults", pesan);
        Intent intent = new Intent("1");
        intent.putExtra("BROADCAST", pesan);
        sendBroadcast(intent);
    }
    catch (Exception e) {
        Log.e("publishResults", e.toString());
    }
}
}
}

```

Pada contoh di atas kita membuat servis baru yang merupakan turunan dari IntentService :

```
public class ServisLogin extends IntentService {
```

Selanjutnya kita melakukan override pada bagian metode onHandleIntent yang merupakan metode pada IntentService.

```
@Override
protected void onHandleIntent(Intent intent) {
```

Pada servis ini kita ambil variabel pesan dari putExtra dengan metode getExtras()

```
Bundle b = intent.getExtras();
```

Jika objek bundle bukanlah null artinya variabel intent diinisialisasi sebelumnya melalui putExtra.

```
if (b != null) {
```

Langkah selanjutnya adalah menyimpan variabel dari putExtra ke dalam 2 variabel string :

```
String uname = (String) b.get("login");
```

```
String passwd = (String) b.get("password");
```

Pada contoh di atas variabel dari putExtra kita simpan di 2 variabel bertipe string dengan nama variabel uname dan passwd

Langkah selanjutnya adalah mengirimkan data username dan password tersebut ke server :

```
String[] key_str = {"user", "pass"};
String[] val_str = {uname, passwd};
String res = spr._execute(getResources().getString(R.string.url),key_str, val_str);
```

Pada contoh di atas kita eksekusi metode _execute untuk melakukan HTTP POST data username dan password.

Langkah akhir dari servis adalah mengecek hasil response balasan dari api php di server. Jika berisi string success maka login berhasil, langkah selanjutnya hasil login apakah berhasil atau gagal dibroadcast ke activity MainActivity melalui metode publishResults(pesan);

```
if (res.indexOf("success") != -1) {
    pesan = "berhasil";
}
else {
    pesan = "gagal";
}
publishResults(pesan);
```

Selanjutnya berikut ini isi MainActivity.java:

```
package com.ringlayer.aplikasilogin;
import android.Manifest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    private Toast mToastToShow;
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            util.requestPermission(Manifest.permission.INTERNET, 3);
            final Button go = (Button) findViewById(R.id.tombol_login);
            go.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Submit();
                }
            });
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    public void showToast(String msg_to_display, int second_to_disp) {
```

```

        int toastDurationInMilliseconds = second_to_disp * 1000;
        mToastToShow = Toast.makeText(this, msg_to_display, Toast.LENGTH_LONG);
        mToastToShow.setGravity(Gravity.CENTER_VERTICAL | Gravity.CENTER_HORIZONTAL, 0,
0);
        CountDownTimer toastCountDown;
        toastCountDown = new CountDownTimer(toastDurationInMilliseconds, 1000 /*Tick
duration*/) {
            public void onTick(long millisUntilFinished) {
                mToastToShow.show();
            }
            public void onFinish() {
                mToastToShow.cancel();
            }
        };
        mToastToShow.show();
        toastCountDown.start();
    }
    void Submit() {
        try {
            EditText e_username;
            EditText e_password;
            showToast("Proses login", 4);
            e_username = (EditText) findViewById(R.id.username);
            e_password = (EditText) findViewById(R.id.password);
            Intent msgIntent = new Intent(this, ServisLogin.class);
            msgIntent.putExtra("login", e_username.getText().toString());
            msgIntent.putExtra("password", e_password.getText().toString());
            startService(msgIntent);
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    private BroadcastReceiver br = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String act = "";
            final Context konteks = context;
            try {
                Bundle bundle = intent.getExtras();
                if (bundle != null) {
                    act = bundle.getString("BROADCAST").trim();
                    Log.e("[+] pgonReceive", act);
                    if (act.indexOf("berhasil") != -1) {
                        Intent intentmem = new Intent(MainActivity.this,
MemberArea.class);
                        startActivity(intentmem);
                        finish();
                    }
                    else {
                        showToast("Login Success", 4);
                    }
                }
            }
            catch (Exception e) {
                Log.e("onReceive", e.toString());
            }
        }
    };
    @Override
    protected void onStart() {
        super.onStart();
        registerReceiver(br, new IntentFilter("1"));
    }
    @Override
    protected void onPause() {
        super.onPause();
        unregisterReceiver(br);
    }
}

```

```
}
```

Pada contoh di atas, pertama tama diinisialisasi objek BroadcastReceiver dengan nama br di mana objek ini akan digunakan menangkap pesan broadcast dari servis yang melakukan post login ke api php di atas.

```
private BroadcastReceiver br = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String act = "";
        final Context konteks = context;
        try {
            Bundle bundle = intent.getExtras();
            if (bundle != null) {
                act = bundle.getString("BROADCAST").trim();
                Log.e("[+] pgonReceive", act);
                if (act.indexOf("berhasil") != -1) {
                    Intent intentmem = new Intent(MainActivity.this,
MemberArea.class);

                    startActivity(intentmem);
                    finish();
                }
            }
            else {
                showToast("Login Success", 4);
            }
        }
        catch (Exception e) {
            Log.e("onReceive", e.toString());
        }
    }
};
```

Perhatikan baris ini :

```
Bundle bundle = intent.getExtras();
```

getExtras() digunakan untuk menangkap pesan broadcast di mana jika pesan broadcast login berhasil maka activity MemberArea akan dijalankan :

```
Intent intentmem = new Intent(MainActivity.this, MemberArea.class);
                    startActivity(intentmem);
                    finish();
```

Pada activity MemberArea, kita hanya menampilkan toast pesan. Berikut ini isi MemberArea.java :

```
package com.ringlayer.aplikasilogin;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;
public class MemberArea extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_member_area);
        Toast toas = Toast.makeText(this, "Welcome to member area", Toast.LENGTH_LONG);
    }
}
```

C. Aplikasi Input ke Database MySQL

Pada contoh kali ini, kita akan membuat aplikasi sederhana yang fungsinya melakukan input ke database mysql.

Cara Kerja Aplikasi

Cara kerja aplikasi ini adalah sebagai berikut : Aplikasi android akan mengirimkan http post berupa data dari inputan form di android ke aplikasi web (php) yang berfungsi untuk menyimpan data di server ke dalam database mysql.

Pertama-tama siapkan database mysql dengan nama "input", selanjutnya siapkan tabel dengan nama crud, dengan struktur di bawah ini:

```
CREATE TABLE `crud` (  
  `nama` text NOT NULL,  
  `pesan` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
COMMIT;
```

Selanjutnya adalah membuat aplikasi web crud sederhana, pertama tama download file get_and_post.php dari https://raw.githubusercontent.com/ringlayer/simple-php-to-prevent-basic-web-attack/master/get_and_post.php lalu simpan di c:\xampp\htdocs\input\get_and_post.php

Selanjutnya siapkan aplikasi web dengan php yang fungsinya adalah crud, yang akan kita simpan di c:\xampp\htdocs\crud\crud.php

Isi file crud.php:

```
<?php  
require_once("get_and_post.php");  
$nama = trim($_POST['nama']);  
$pesan = trim($_POST['pesan']);  
  
$con = mysqli_connect("localhost","root","","input");  
  
if ($con) {  
  $sql = "INSERT INTO `crud` (`nama`, `pesan`) VALUES ('$nama', '$pesan')";  
  $result = mysqli_query($con, $sql);  
  if ($result) {  
    print "success";  
  }  
}  
?>
```

Source php di atas berguna untuk menangkap http post variabel nama dan pesan yang akan kita isikan ke tabel crud.

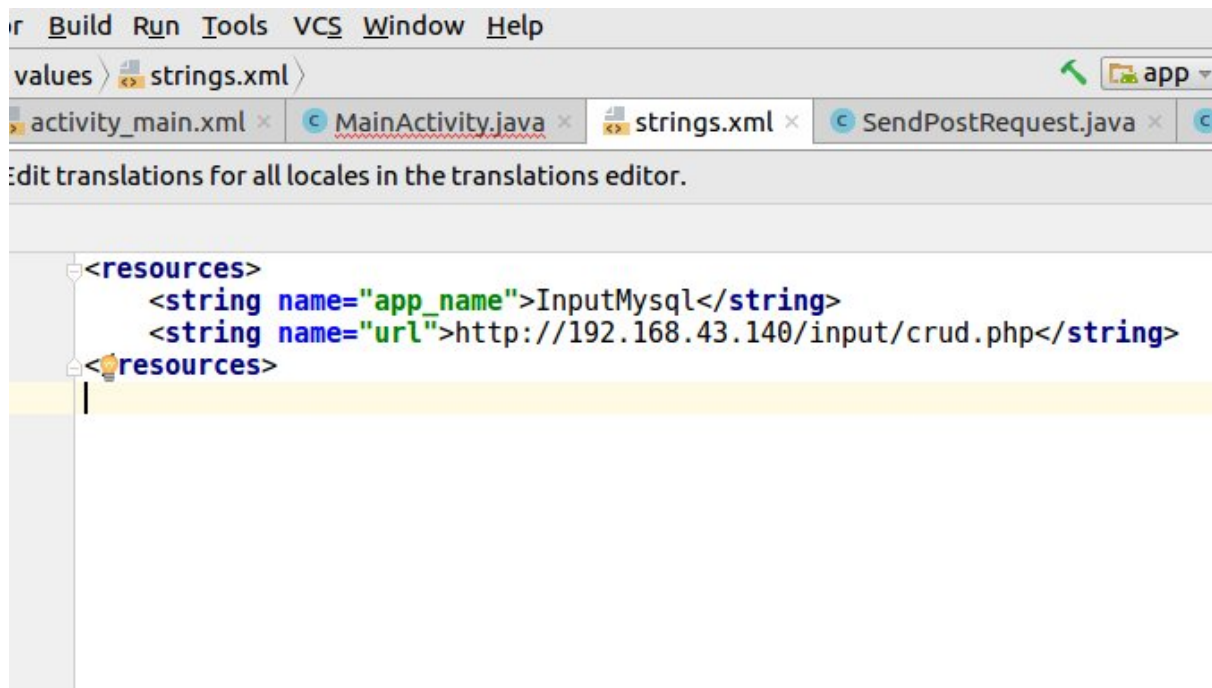
Selanjutnya, buka android studio dan buat proyek baru dengan nama InputMysql

Bagian XML

Buka file strings.xml di folder res → values. Tambahkan url untuk api crud :

```
<string name="url">http://192.168.43.140/input/crud.php</string>
```

Sesuaikan alamat url di atas dengan alamat url api crud server anda.



Karena kita akan mengirimkan http post ke server, maka aplikasi android ini memerlukan permissi untuk internet, tambahkan baris berikut ini pada AndroidManifest.xml:

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

Berikut ini isi lengkap AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.inputmysql">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

```

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

```

Pada aplikasi kali ini hanya memerlukan 1 layout yaitu activity_main.xml yang merupakan form inputan. Berikut ini isi activity_main.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/background"
    android:gravity="center" tools:context="com.ringlayer.inputmysql.MainActivity">
    <Space
        android:layout_width="1dp"
        android:layout_height="30dp"/>
    <EditText
        android:id="@+id/nama"
        android:layout_width="250dp"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:inputType="textPersonName"
        android:hint="Nama"
        android:paddingLeft="10dp"
        android:textColor="@color/colorAccent" />
    <Space
        android:layout_width="1dp"
        android:layout_height="30dp"/>
    <EditText
        android:id="@+id/pesan"
        android:layout_width="250dp"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:inputType="textPersonName"
        android:hint="Pesan"
        android:paddingLeft="10dp"
        android:textColor="@color/colorAccent" />
    <Space
        android:layout_width="1dp"
        android:layout_height="30dp"/>
    <Button
        android:id="@+id/submit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="Submit"
    />
</LinearLayout>

```

Bagian Source Code

Sama seperti aplikasi login sebelumnya, kita tambahkan 2 kelas untuk request permissi dan melakukan post request.

Buat kelas baru dengan nama Util yang akan digunakan untuk request permissi internet.

Isi file Util.java :

```
package com.ringlayer.inputmysql;
/*
 * Created by ringlayer on 26/11/18.
 */
import android.app.Activity;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.os.Build;
import android.os.Environment;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.Toast;
import java.io.File;
import java.nio.charset.Charset;
import java.util.Random;
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        try {
            mContext = context;
            if (android.os.Build.VERSION.SDK_INT > 9) {
                StrictMode.ThreadPolicy policy = new
                StrictMode.ThreadPolicy.Builder().permitAll().build();
                StrictMode.setThreadPolicy(policy);
                StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
                StrictMode.setVmPolicy(builder.build());
            }
        }
        catch (Exception e) {
            Log.e("_do_log_debug", e.toString());
        }
    }

    public void askForPermission(String permission, Integer requestCode) {
        try {
            if (ContextCompat.checkSelfPermission(mContext, permission) !=
                PackageManager.PERMISSION_GRANTED) {
                if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                    permission)) {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                } else {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                }
            }
        }
        catch (Exception e) {
```



```

        Log.e("_do_log_debug", e.toString());
    }
}

```

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```

if (ContextCompat.checkSelfPermission(mContext, permission) !=
PackageManager.PERMISSION_GRANTED) {

```

Pada rutin di atas, jika permissi bukanlah `PERMISSION_GRANTED` , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```

ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);

```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Buat kelas baru dengan nama SendPostRequest. Berikut ini isi file SendPostRequest.java :

```

package com.ringlayer.inputmysql;
import android.os.StrictMode;
import android.util.Log;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Arrays;
import java.util.Iterator;
import java.util.List;
/**
 * Created by ringlayer on 23/06/18.
 *
 * modified from https://www.studytutorial.in/android-httpurlconnection-post-and-get-request-tutorial
 */
public class SendPostRequest {
    public String _execute(String url_post,String[] key_str, String[] val_str) {
        try {
            if (android.os.Build.VERSION.SDK_INT > 9) {
                try {
                    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
                    StrictMode.setThreadPolicy(policy);
                    StrictMode.VmPolicy.Builder builder = new
StrictMode.VmPolicy.Builder();
                    StrictMode.setVmPolicy(builder.build());
                }
                catch (Exception e) {
                    String stackTrace = Log.getStackTraceString(e);
                    Log.e("[-] Strictmode", stackTrace);
                }
            }

```

```

    }
}
URL url = new URL(url_post);
int i;
JSONObject postDataParams = new JSONObject();
for (i = 0; i < key_str.length; i++) {
    postDataParams.put(key_str[i], val_str[i]);
}
Log.e("params", postDataParams.toString());
URLConnection conn = (URLConnection) url.openConnection();
conn.setReadTimeout(15000 /* milliseconds */);
conn.setConnectTimeout(15000 /* milliseconds */);
conn.setRequestMethod("POST");
conn.setDoInput(true);
conn.setDoOutput(true);
conn.connect();
OutputStream os = conn.getOutputStream();
BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, "UTF-
8"));
writer.write(getPostDataString(postDataParams));
writer.flush();
writer.close();
os.close();
InputStreamReader in = new InputStreamReader((InputStream)
conn.getContent());
BufferedReader buff = new BufferedReader(in);
String line = "";
StringBuffer text = new StringBuffer();
while (line != null) {
    line = buff.readLine();
    text.append(line + "\n");
}
Log.e("res:", text.toString());
String res = text.toString();
return res;
}
catch (Exception e){
    return new String("Exception: " + e.getMessage());
}
}

public String getPostDataString(JSONObject params) throws Exception {
    StringBuilder result = new StringBuilder();
    boolean first = true;
    Iterator<String> itr = params.keys();
    while(itr.hasNext()){
        String key= itr.next();
        Object value = params.get(key);
        if (first)
            first = false;
        else
            result.append("&");
        result.append(URLEncoder.encode(key, "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(value.toString(), "UTF-8"));
    }
    return result.toString();
}
}

```

Pada kelas di atas kita melakukan non blocking dengan memanfaatkan fitur kelas StrictMode untuk mencegah blocking pada User Interface yang sedang aktif saat pengiriman post request ke server

```

StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

```

Selanjutnya kita membuat koneksi http ke server melalui rutin di bawah ini :

```

URL url = new URL(url_post);
    int i;
    JSONObject postDataParams = new JSONObject();
    for (i = 0; i < key_str.length; i++) {
        postDataParams.put(key_str[i], val_str[i]);
    }
    Log.e("params", postDataParams.toString());
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(15000 /* milliseconds */);
    conn.setConnectTimeout(15000 /* milliseconds */);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);
    conn.connect();

```

Rutin di atas berguna mengatur properti untuk koneksi http berupa read timeout, connect timeout, jenis request yaitu POST dan parameter post yang akan dikirim ke server. Setelah semuanya siap koneksi dan pengiriman data ke server dieksekusi pada baris `conn.connect()`;

Rutin selanjutnya berguna untuk mempersiapkan buffer yang akan digunakan untuk menangkap data response balasan dari server

```

BufferedReader buff = new BufferedReader(in);
    String line = "";
    StringBuffer text = new StringBuffer();
    while (line != null) {
        line = buff.readLine();
        text.append(line + "\n");
    }
    Log.e("res:", text.toString());
    String res = text.toString();

```

Pada contoh di atas, balasan dari server pada akhirnya akan disimpan di variabel bertipe String dengan nama `res`. Di mana sebelumnya response dari server ditangkap di variabel objek `StringBuffer` `text` kemudian pada bagian baris `"String res = text.toString();"` objek buffer dikonversi menjadi String.

Berikut ini isi `MainActivity.java` :

```

package com.ringlayer.inputmysql;
import android.Manifest;
import android.os.Build;
import android.os.CountDownTimer;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

```

```

import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import static android.Manifest.permission.CAMERA;
public class MainActivity extends AppCompatActivity {
    public Util util = new Util(MainActivity.this);
    SendPostRequest spr = new SendPostRequest();
    private Toast mToastToShow;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            util.askForPermission(Manifest.permission.INTERNET, 3);
        }
        final Button go = (Button) findViewById(R.id.submit);
        go.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Submit();
            }
        });
    }
    public void showToast(String msg_to_display, int second_to_disp) {
        int toastDurationInMilliseconds = second_to_disp * 1000;
        mToastToShow = Toast.makeText(this, msg_to_display, Toast.LENGTH_LONG);
        mToastToShow.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0,
0);

        CountDownTimer toastCountDown;
        toastCountDown = new CountDownTimer(toastDurationInMilliseconds, 1000 /*Tick
duration*/) {
            public void onTick(long millisUntilFinished) {
                mToastToShow.show();
            }
            public void onFinish() {
                mToastToShow.cancel();
            }
        };
        mToastToShow.show();
        toastCountDown.start();
    }
    void Submit() {
        try {
            EditText e_nama;
            EditText e_pesanan;
            String nama, pesan;
            showToast("Proses pengiriman data", 4);
            e_nama = (EditText) findViewById(R.id.nama);
            e_pesanan = (EditText) findViewById(R.id.pesanan);
            nama = e_nama.getText().toString();
            pesan = e_pesanan.getText().toString();
            String[] key_str = {"nama", "pesanan"};
            String[] val_str = {nama, pesan};
            String res = spr._execute(getResources().getString(R.string.url),key_str,
val_str);

            if (res.indexOf("success") != -1) {
                showToast("input berhasil", 3);
            }
            else {
                showToast("input gagal diinput", 3);
            }
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
}

```

Pada callback onCreate kita buat referensi ke tombol dengan id submit :

```
final Button go = (Button) findViewById(R.id.submit);
```

Selanjutnya kita pasang metode callback setOnClickListener agar ketika tombol diklik mentrigger rutin yang ingin kita eksekusi.

```
go.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        Submit();  
    }  
});
```

Pada saat tombol diklik, aplikasi akan menjalankan metode Submit.

```
public void onClick(View v) {  
    Submit();  
}
```

Metode submit digunakan untuk mengirimkan data inputan form yang diisi user ke server.

D. Aplikasi Memilih Image dari Storage

Pada contoh kali ini kita akan membuat aplikasi untuk memilih image dari storage penyimpanan android untuk kemudian kita sertakan dalam form isian.

Pertama tama buat dahulu proyek android baru dengan nama : “PilhFoto”

Pada contoh kali ini kita akan menggunakan 2 activity :

1. MainActivity dengan layoutnya activity_main.xml
2. PilihFotoFragment dengan layoutnya fragment_pilih_foto.xml

Cara Kerja Aplikasi

Pada saat pertama kali dijalankan aplikasi ini akan merequest permissi baca tulis device android. Selanjutnya aplikasi akan meload fragment yang isinya form untuk memilih foto dari perangkat android. Saat user mengklik tombol “Pilih Foto”, maka user bisa memilih image di device android, Foto kemudian akan disimpan di direktori penyimpanan foto dengan nama tmp_from_local.jpg. Setelah selesai image tmp_from_local.jpg akan dirender di imageview pada fragment.

Aplikasi ini berguna saat form inputan dengan attachment foto.

Bagian XML

Isi file activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    tools:context="com.ringlayer.pilhfoto.MainActivity">
    <FrameLayout
        android:id="@+id/frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </FrameLayout>
</LinearLayout>
```

Isi file fragment_pilih_foto.xml:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.pilhfoto.PilihFotoFragment">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```

xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:id="@+id/background"
android:gravity="center">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
    <ImageView
        android:layout_gravity="center"
        android:id="@+id/foto_view"
        android:layout_width="200dp"
        android:layout_height="250dp"
        android:layout_centerHorizontal="true"></ImageView>
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
    <Button
        android:id="@+id/tombol_foto"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:text="Pilih Photo" />
    </LinearLayout>
</LinearLayout>
</FrameLayout>

```

Bagian Source Code

Buat kelas baru dengan nama Util. Berikut ini isi file Util.java :

```

package com.ringlayer.pilhfoto;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
/*
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }

        public void askForPermission(String permission, Integer requestCode) {

```

```

        if (ContextCompat.checkSelfPermission(mContext, permission) !=
PackageManager.PERMISSION_GRANTED) {

            if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
permission)) {

                ActivityCompat.requestPermissions(mContext, new String[]{permission},
requestCode);
            } else {
                ActivityCompat.requestPermissions(mContext, new String[]{permission},
requestCode);
            }
        }
    }
}

```

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```

if (ContextCompat.checkSelfPermission(mContext, permission) !=
PackageManager.PERMISSION_GRANTED) {

```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```

ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);

```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Berikut ini isi file MainActivity.java

```

package com.ringlayer.pilhfoto;

import android.Manifest;
import android.content.Intent;
import android.graphics.Bitmap;
import android.media.MediaScannerConnection;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.os.StrictMode;
import android.provider.MediaStore;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.ImageView;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
/*
 * Created by ringlayer on 24/11/18.
 */
public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    public Util util = new Util(MainActivity.this);

```



```

private int PICK_IMAGE_REQUEST = 1;
@Override
protected void onCreate(Bundle savedInstanceState) {
    try {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
            util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
        }
        FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
        ft.replace(R.id.frame, new PilihFotoFragment(), "FOTO");
        ft.commit();
        ft.addToBackStack(null);
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}

public void PilihFoto() {
    try {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST);
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    try {
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
            StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
            StrictMode.setVmPolicy(builder.build());
        }
        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data !=
null && data.getData() != null) {
            Uri uri = data.getData();
            try {
                Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), uri);
                String path = saveImage(bitmap);
                PilihFotoFragment fotof = (PilihFotoFragment)
getSupportFragmentManager().findFragmentByTag("FOTO");
                fotof.SetFotoImageViewFromPath(path);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}

public String saveImage(Bitmap myBitmap) {
    String path = "";
    try {
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        ByteArrayOutputStream bytes = new ByteArrayOutputStream();

```

```

        myBitmap.compress(Bitmap.CompressFormat.JPEG, 90, bytes);
        File TmpDirectory = new File(Environment.getExternalStorageDirectory() +
"/foto");
        if (!TmpDirectory.exists()) {
            TmpDirectory.mkdirs();
        }
        try {
            File f = new File(TmpDirectory, "tmp_from_local.jpg");
            f.createNewFile();
            FileOutputStream fo = new FileOutputStream(f);
            fo.write(bytes.toByteArray());
            MediaScannerConnection.scanFile(this, new String[]{f.getPath()}, new
String[]{"image/jpeg"}, null);
            fo.close();
            path = f.getAbsolutePath();
        }
        catch (Exception e) {
            Log.e("SaveImage Err : ", e.toString());
        }
    }
    catch (Exception e) {
        Log.e("mkdirs Err : ", e.toString());
    }
    return path;
}
}

```

PENJELASAN

Pada activity di atas, jika didapati versi android adalah marshmallow ke atas maka aplikasi akan melakukan request permisi ke user untuk baca dan tulis file system.

```

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    //util.askForPermission(CAMERA,0);
    util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
    util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
}

```

Perhatikan rutin di bawah ini :

```

FragmentManager ft = getSupportFragmentManager().beginTransaction();
ft.replace(R.id.frame, new PilihFotoFragment(), "FOTO");
ft.commit();
ft.addToBackStack(null);

```

Rutin di atas akan memanggil metode beginTransaction pada getSupportFragmentManager untuk memulai transaksi pengaturan fragment.

Rutin ini

```
ft.replace(R.id.frame, new PilihFotoFragment(), "FOTO");
```

digunakan untuk menggantikan **FrameLayout** yang memiliki id frame dengan objek baru dari kelas PilihFotoFragment di mana pada bagian onCreateView, PilihFotoFragment akan merender layout fragment_pilih_foto.

Metode PilihFoto digunakan untuk memilih foto dari file system android.

```

public void PilihFoto() {
    try {
        Intent intent = new Intent();
        intent.setType("image/*");
    }
}

```

```

        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST);

```

Selanjutnya pada onActivityResult (yang merupakan metode bawaan parent class yang dioverride), jika requestCode berupa PICK_IMAGE_REQUEST maka intent akan disimpan ke dalam bentuk objek Uri, untuk penjelasan detail tentang Uri bisa dibaca di <https://developer.android.com/reference/android/net/Uri>

```

        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data !=
null && data.getData() != null) {
            Uri uri = data.getData();
            try {
                Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), uri);
                String path = saveImage(bitmap);
                PilihFotoFragment fotof = (PilihFotoFragment)
getSupportFragmentManager().findFragmentByTag("FOTO");
                fotof.SetFotoImageViewFromPath(path);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
}

```

```

PilihFotoFragment fotof =
(PilihFotoFragment)getSupportFragmentManager().findFragmentByTag("FOTO");

```

Pada bagian ini akan dicari referensi fragment berdasarkan TAG, di sini dicari yang menggunakan tag FOTO yang akan disimpan ke objek berupa PilihFotoFragment dengan nama fotof.

Selanjutnya

```
fotof.SetFotoImageViewFromPath(path);
```

SetFotoImageViewFromPath akan merender foto agar tampil

Berikut ini isi activity fragment PilihFotoFragment.java :

```

package com.ringlayer.pilhfoto;
import android.content.Context;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.graphics.BitmapFactory;
import com.ringlayer.pilhfoto.R;
import java.io.File;
/*
 * Created by ringlayer on 24/11/18.
 */
public class PilihFotoFragment extends Fragment {

```

```

private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
private String mParam1;
private String mParam2;
private OnFragmentInteractionListener mListener;
public PilihFotoFragment() {
}
public static PilihFotoFragment newInstance(String param1, String param2) {
    PilihFotoFragment fragment = new PilihFotoFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_pilih_foto, container, false);
    try {
        Button button = (Button) v.findViewById(R.id.tombol_foto);
        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                ((MainActivity) getActivity()).PilihFoto();
            }
        });
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
    return v;
}
public void SetFotoImageViewFromPath(String full_path) {
    try {
        File imgFile = new File(full_path);
        if(imgFile.exists()) {
            ImageView foto_view = (ImageView)
getActivity().findViewById(R.id.foto_view);
            foto_view.setImageBitmap(loadImage(full_path));
        }
        Log.d("SetFototoImageView : ", full_path);
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}
private Bitmap loadImage(String imgPath) {
    BitmapFactory.Options options;
    try {
        options = new BitmapFactory.Options();
        options.inSampleSize = 2;
        Bitmap bitmap = BitmapFactory.decodeFile(imgPath, options);
        return bitmap;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

```

```

    public void onPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
    }
    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }
    public interface OnFragmentInteractionListener {
        void onFragmentInteraction(Uri uri);
    }
}

```

PENJELASAN

```

public class PilihFotoFragment extends Fragment {

```

Rutin ini akan membuat kelas baru yang merupakan kelas turunan dari parent class "Fragment".

```

    public static PilihFotoFragment newInstance(String param1, String param2) {
        PilihFotoFragment fragment = new PilihFotoFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

```

digunakan untuk membuat instance baru dari kelas PilihFotoFragment

```

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

```

Pada tubuh metode onCreateView dilakukan beberapa rutin :

1. merender tampilan layout *fragment_pilih_foto*
2. menyiapkan callback setOnClickListener pada tombol dengan id tombol_foto

```

@Override
    public void onClick(View v)
    {
        ((MainActivity) getActivity()).PilihFoto();
    }

```

Metode onClick akan dioverride sehingga menjalankan metode PilihFoto pada MainActivity

```

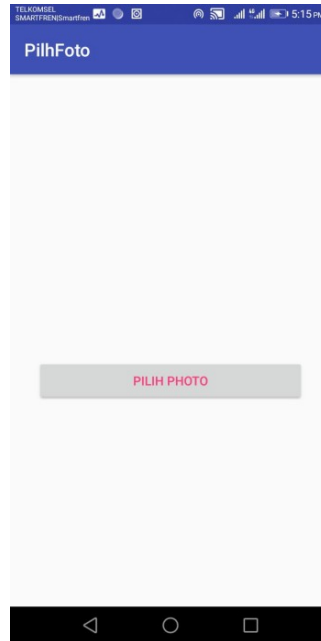
    public void SetFotoImageViewFromPath(String full_path) {

```

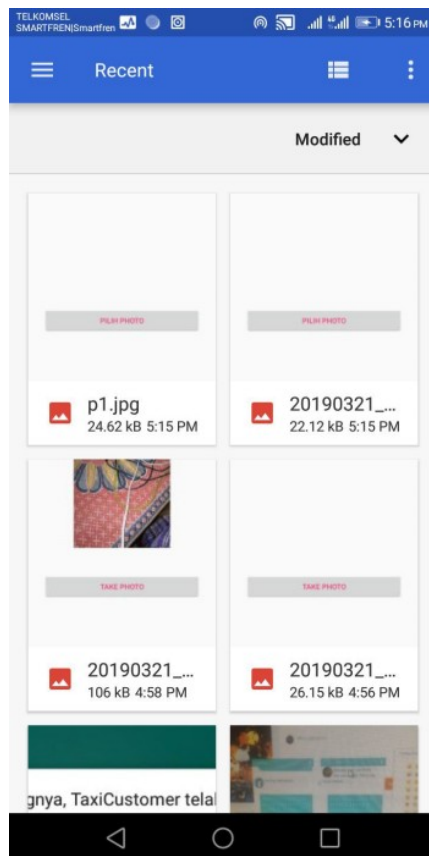
Metode ini digunakan untuk merender image dari full_path pada ImageView dengan id *foto_view*

Uji Coba Aplikasi

Pada saat aplikasi ini pertama dijalankan, aplikasi ini akan merequest permisi untuk mengakses file system, klik allow.



Ketika tombol pilih foto diklik, aplikasi akan mengakses file system android.support.v4



Pilih salah satu image yang akan diattach

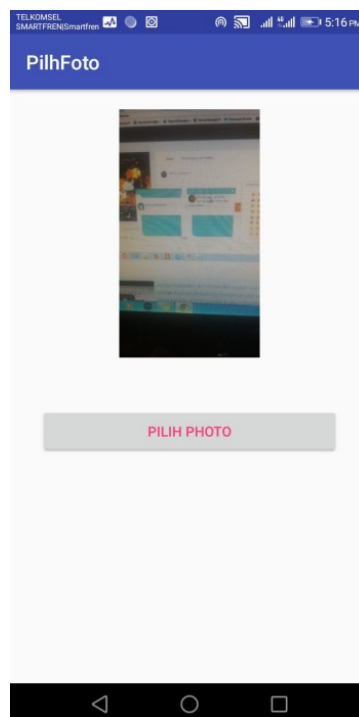
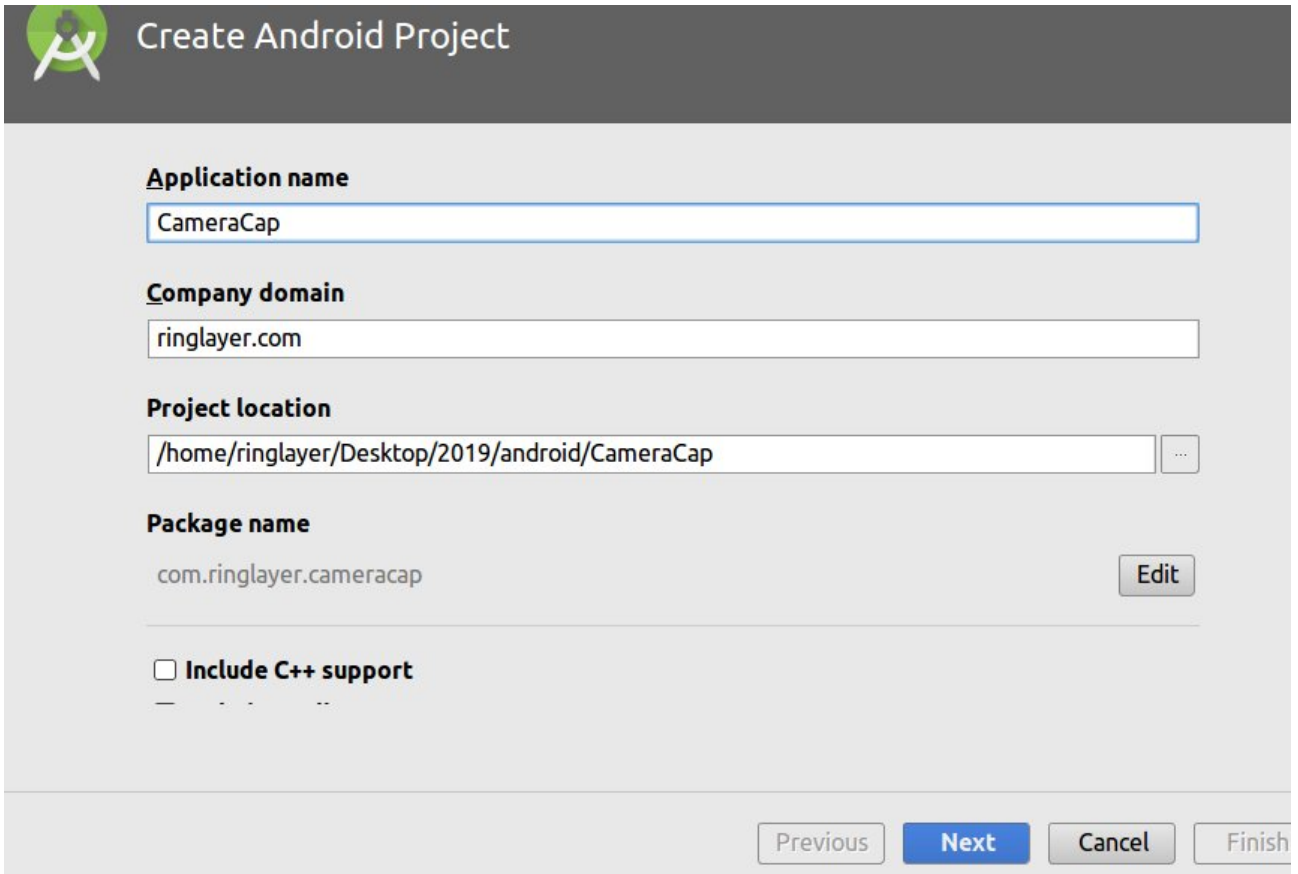


Image akan disimpan di penyimpanan sementara dan diload di imageview.

E. Aplikasi Pengambil Foto dari Kamera

Pada contoh kali ini, kita akan membuat proyek aplikasi sederhana untuk mengambil foto dari kamera android. Pertama tama buat dahulu proyek baru dengan nama “CameraCap”



Create Android Project

Application name
CameraCap

Company domain
ringlayer.com

Project location
/home/ringlayer/Desktop/2019/android/CameraCap

Package name
com.ringlayer.cameracap Edit

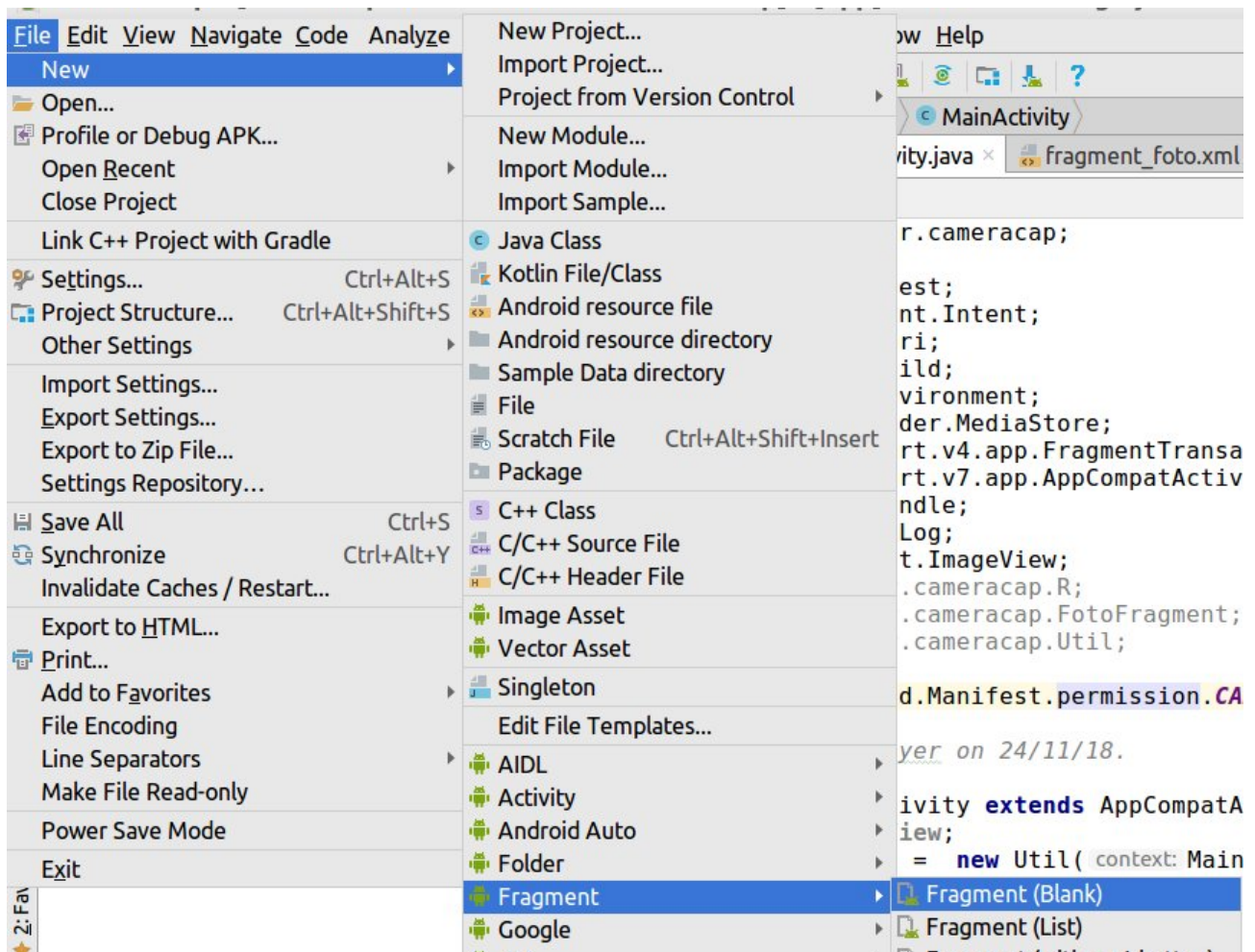
☐ **Include C++ support**

Previous Next Cancel Finish

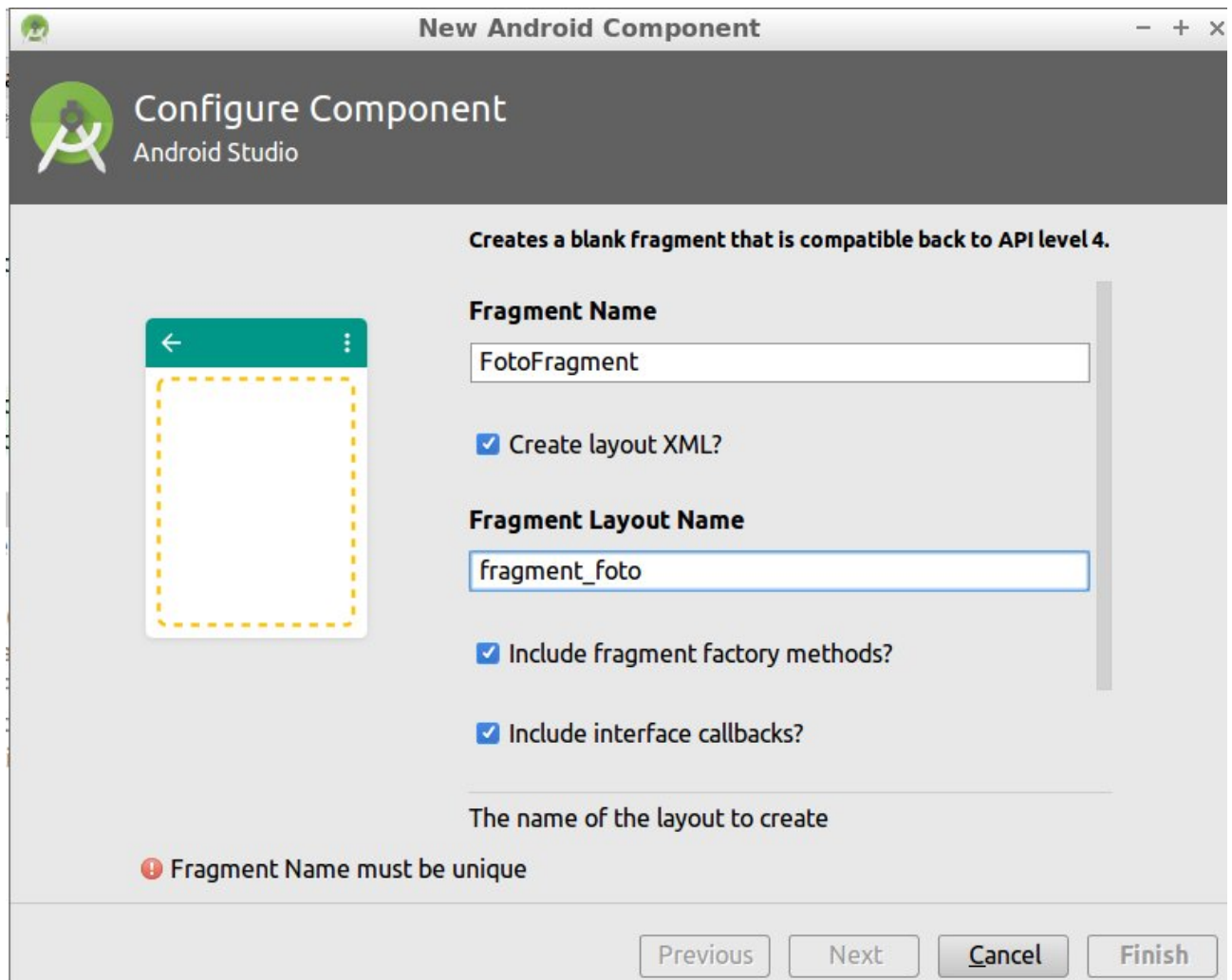
Cara Kerja Aplikasi

Pada saat pertama kali dijalankan aplikasi akan meminta izin request kamera dan operasi baca tulis file system. Selanjutnya fragment yang berisi tombol untuk mengakses kamera di load. Saat user mengklik maka user bisa mengakses kamera, ketika selesai hasil jepretan disimpan di penyimpanan android.

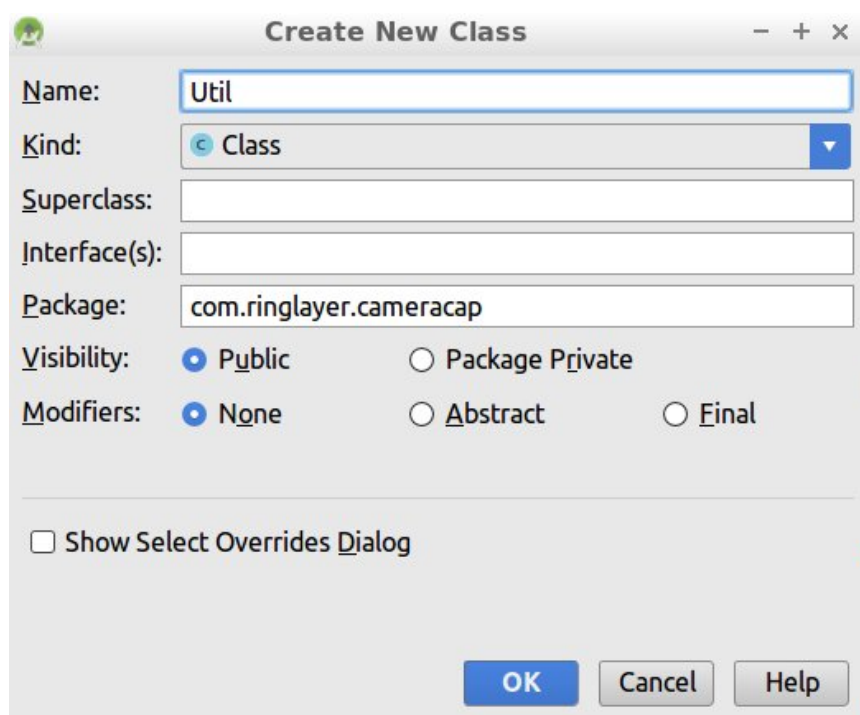
Pada contoh kali ini, kita akan menggunakan frame layout. Selanjutnya siapkan framelayout:



Gunakan nama fragment : FotoFragment



Buat kelas baru dengan nama Util dengan cara Klik Menu File → New → Java Class.



Bagian Source Code

Kopi dan pastekan kode ini pada Util.java

```
package com.ringlayer.cameracap;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
/*
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);}

        public void askForPermission(String permission, Integer requestCode) {
            if (ContextCompat.checkSelfPermission(mContext, permission) !=
                PackageManager.PERMISSION_GRANTED) {

                if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                    permission)) {

                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                } else {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                }
            }
        }
    }
}
```

Pada baris **package** com.ringlayer.cameracap;

Sesuaikan dengan nama paket aplikasi Anda.

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```
if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {
```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```
ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);
```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permisi.

Berikut ini isi MainActivity.java

```
package com.ringlayer.cameracap;
import android.Manifest;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.ImageView;
import com.ringlayer.cameracap.R;
import com.ringlayer.cameracap.FotoFragment;
import com.ringlayer.cameracap.Util;
import java.io.File;
import static android.Manifest.permission.CAMERA;
/*
 * Created by ringlayer on 24/11/18.
 */
public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                util.askForPermission(CAMERA, 0);
                util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
                util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
            }
            FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
            ft.replace(R.id.frame, new FotoFragment(), "FOTO");
            ft.commit();
            ft.addToBackStack(null);
        }
        catch (Exception e) {
            Log.e("Err : ", e.toString());
        }
    }
    public void AmbilFoto() {
        try {
            String imageFolderPath = Environment.getExternalStorageDirectory().toString() + "/foto";
            File imagesFolder = new File(imageFolderPath);
            imagesFolder.mkdirs();
            String imageName = "tmp_f.jpg";
            Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(new File(imageFolderPath,
            imageName)));
            startActivityForResult(takePictureIntent, 0);
        }
        catch (Exception e) {
```

```

        Log.e("Err : ", e.toString());
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    try {
        if (requestCode == 0 && resultCode == RESULT_OK) {
            String imageFolderPath = Environment.getExternalStorageDirectory().toString() + "/foto";
            String imageName = "tmp_f.jpg";
            String full_path = imageFolderPath + "/" + imageName;
            File imgFile = new File(full_path);
            FotoFragment fotof = (FotoFragment)
getSupportFragmentManager().findFragmentByTag("FOTO");
            fotof.SetFototoImageView();
        }
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}
}

```

PENJELASAN

Metode AmbilFoto digunakan untuk mengambil foto dari kamera dengan intent.

```

Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(new File(imageFolderPath,
imageName)));

```

Hasil dari intent diproses oleh metode callback onActivityResult. Foto hasil jepretan akan disimpan di folder sdcard foto dengan nama tmp_f.jpg. Kemudian file tmp_f.jpg akan dibaca dan dirender di ImageView

Selanjutnya buka FotoFragment.java lalu kopi dan pastekan kode ini:

```

package com.ringlayer.cameracap;
import android.content.Context;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.graphics.BitmapFactory;
import com.ringlayer.cameracap.R;
import java.io.File;
/*
 * Created by ringlayer on 24/11/18.
 */
public class FotoFragment extends Fragment {
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

```

```

private String mParam1;
private String mParam2;
private OnFragmentInteractionListener mListener;
public FotoFragment() {
}
public static FotoFragment newInstance(String param1, String param2) {
    FotoFragment fragment = new FotoFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                          Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_foto, container, false);
    try {
        Button b = (Button) v.findViewById(R.id.tombol_foto);
        boolean front_cam_enable = CheckCamera("front");
        if (front_cam_enable == true) {
            Log.d("onCreate : ", "front camera enabled");
        }
        boolean rear_cam_enable = CheckCamera("rear");
        if (rear_cam_enable == true) {
            Log.d("onCreate : ", "rear camera enabled");
        }
        if (front_cam_enable == true || rear_cam_enable == true) {
            Button button = (Button) v.findViewById(R.id.tombol_foto);
            button.setOnClickListener(new View.OnClickListener()
            {
                @Override
                public void onClick(View v)
                {
                    ((MainActivity) getActivity()).AmbilFoto();
                }
            });
        }
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
    return v;
}
public void SetFototoImageView() {
    try {
        String imageFolderPath = Environment.getExternalStorageDirectory().toString()
+ "/foto";
        String imageName = "tmp_f.jpg";
        String full_path = imageFolderPath + "/" + imageName;
        File imgFile = new File(full_path);
        if(imgFile.exists()) {
            ImageView foto_view = (ImageView)
getActivity().findViewById(R.id.foto_view);
            foto_view.setImageBitmap(loadImage(full_path));
        }
        Log.d("SetFototoImageView : ", full_path);
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}

```

```

    }
    private Bitmap loadImage(String imgPath) {
        BitmapFactory.Options options;
        try {
            options = new BitmapFactory.Options();
            options.inSampleSize = 2;
            Bitmap bitmap = BitmapFactory.decodeFile(imgPath, options);
            return bitmap;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
    public void onPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
    }
    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }
    public interface OnFragmentInteractionListener {
        void onFragmentInteraction(Uri uri);
    }
    private boolean CheckCamera(String front_or_back) {
        boolean res;
        if (front_or_back == "front") {
            res =
getActivity().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_FRONT);
        }
        else {
            res =
getActivity().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA);
        }
        return res;
    }
}

```

Perhatikan baris ini:

```
package com.ringlayer.cameracap;
```

Ganti nama paket aplikasi “com.ringlayer.cameracap” dengan nama paket aplikasi anda.

Pada dasarnya cara kerja fragment di atas sama dengan fragment pada materi training sebelumnya pada bagian fragment pilih foto.

Ada beberapa bagian yang berbeda, misal :

```

Button button = (Button) v.findViewById(R.id.tombol_foto);
button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        ((MainActivity) getActivity()).AmbilFoto();
    }
});

```


Pada source di atas objek berupa Button dengan nama button akan diisi dengan referensi ke Button di layout dengan id tombol_foto.

Selanjutnya akan dipasang metode callback yang terdapat pada objek button yaitu :

setOnClickListener, di mana parameter dari metode setOnClickListener merupakan interface View.OnClickListener (penggunaan interface di sini berfungsi untuk abstraksi).

Pada interface View.OnClickListener, metode onClick akan dioverride di mana di contoh kali ini akan dipanggil metode AmbilFoto yang terdapat pada kelas MainActivity.

```
((MainActivity) getActivity()).AmbilFoto();
```

SetFototoImageView merupakan metode yang digunakan untuk merender foto hasil tangkapan kamera ke ImageView dengan id foto_view

```
ImageView foto_view = (ImageView) getActivity().findViewById(R.id.foto_view);  
foto_view.setImageBitmap(loadImage(full_path));
```

Bagian XML

Berikut ini adalah isi file layout activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:orientation="vertical"  
    tools:context="com.ringlayer.cameracap.MainActivity">  
    <FrameLayout  
        android:id="@+id/frame"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent">  
    </FrameLayout>  
</LinearLayout>
```

Ganti nama paket : com.ringlayer.Cameracap dengan nama paket aplikasi Anda.

Berikut ini isi file fragment_foto.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context="com.ringlayer.cameracap.FotoFragment"  
    android:id="@+id/fotofrag">  
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        xmlns:tools="http://schemas.android.com/tools"  
        android:orientation="vertical"  
        android:id="@+id/background"  
        android:gravity="center">  
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```



```

xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_marginTop="20dp"
android:layout_width="match_parent"
android:layout_height="300dp"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical">
<ImageView
    android:layout_gravity="center"
    android:id="@+id/foto_view"
    android:layout_width="200dp"
    android:layout_height="250dp"
    android:layout_centerHorizontal="true"></ImageView>
</LinearLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
    <Button
        android:id="@+id/tombol_foto"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/colorAccent"
        android:text="Take Photo" />
    </LinearLayout>
</LinearLayout>
</FrameLayout>

```

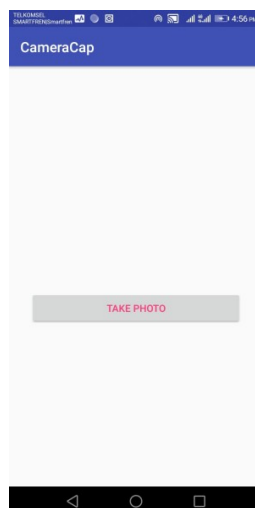
Pada line ini:

```
tools:context="com.ringlayer.cameracap.FotoFragment"
```

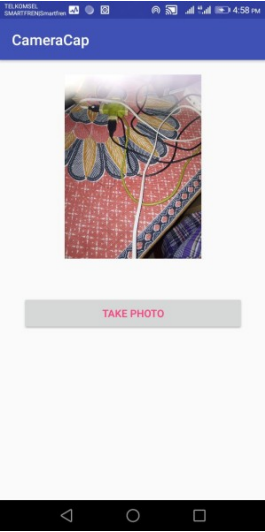
ganti **com.ringlayer.cameracap** dengan nama paket aplikasi Anda

Uji Coba Aplikasi

Pada saat dijalankan aplikasi ini akan meload fragment yang berisi tombol untuk mengakses kamera

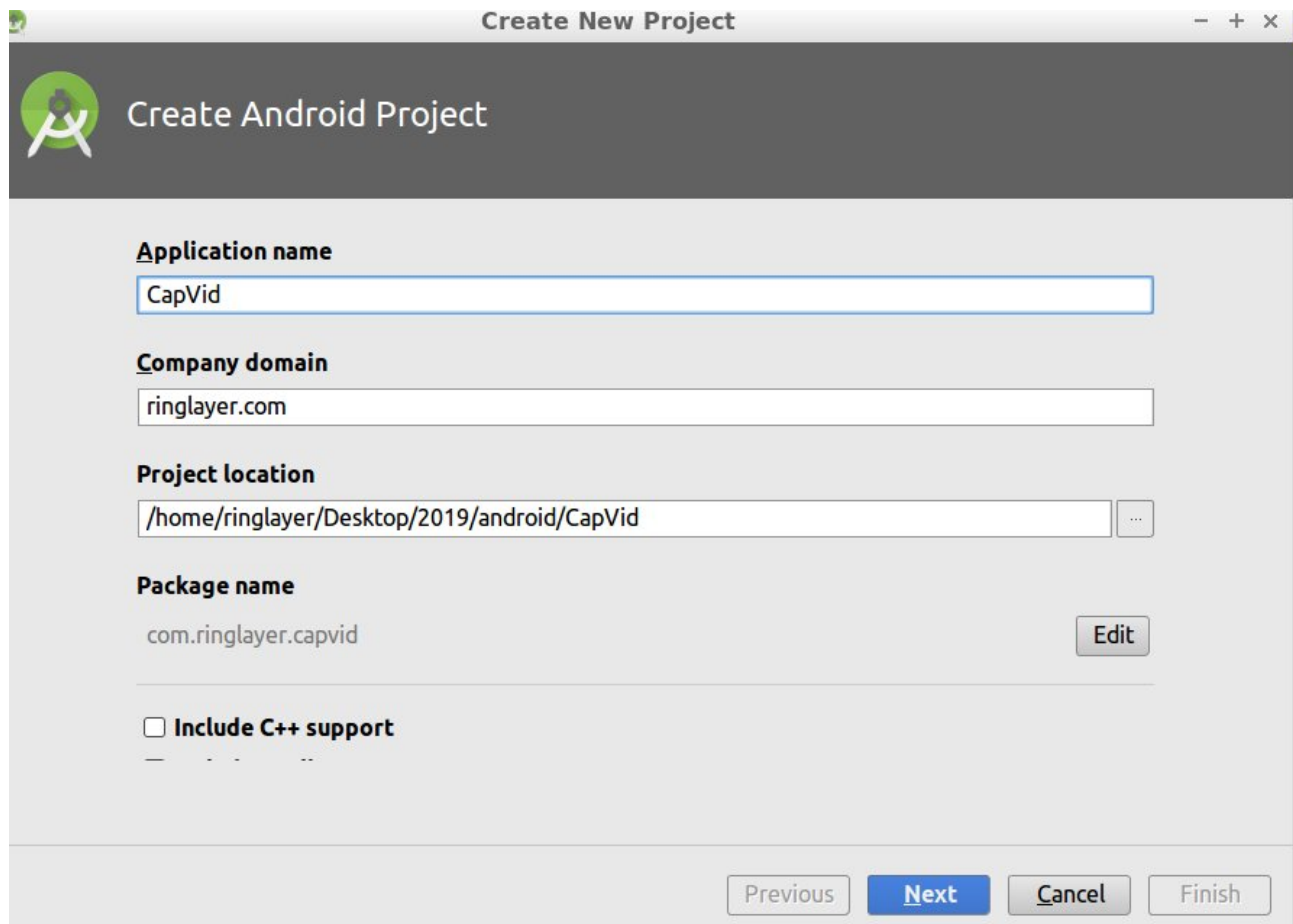


Klik tombol “TAKE PHOTO” untuk mengambil foto dari kamera. Hasil pemotretan akan dirender di ImageView



F. Aplikasi Pengambil Video

Pada contoh kali ini, kita akan membuat proyek aplikasi sederhana untuk mengambil video dari kamera android. Pertama tama buat dahulu proyek baru dengan nama "CapVid"



Create New Project

Create Android Project

Application name
CapVid

Company domain
ringlayer.com

Project location
/home/ringlayer/Desktop/2019/android/CapVid

Package name
com.ringlayer.capvid Edit

☐ **Include C++ support**

Previous Next Cancel Finish

Bagian XML

Aplikasi kali ini menggunakan 2 layout :

1. activity_main.xml yang digunakan oleh activity MainActivity

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

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

2. fragment_take_vid.xml yang digunakan oleh activity fragment TakeVidFragment

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.capvid.TakeVidFragment">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:id="@+id/background"
        android:gravity="center">
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            xmlns:tools="http://schemas.android.com/tools"
            android:orientation="vertical">
            <Button
                android:id="@+id/tombol_foto"
                android:layout_width="300dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:textColor="@color/colorAccent"
                android:text="Take Video" />
        </LinearLayout>
    </LinearLayout>
</FrameLayout>
```

Bagian Source Code

Berikut ini isi dari MainActivity.java

```
package com.ringlayer.capvid;
import android.Manifest;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.ImageView;
import android.widget.Toast;
import com.ringlayer.capvid.R;
import com.ringlayer.capvid.TakeVidFragment;
import com.ringlayer.capvid.Util;
import java.io.File;
import static android.Manifest.permission.CAMERA;
/**
 * Created by ringlayer on 24/11/18.
 */
public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    public Util util = new Util(MainActivity.this);
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    try {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        /*
        * if (Build.VERSION.SDK_INT >= 23) {
        */
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            util.askForPermission(CAMERA, 0);
            util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
            util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
        }
        FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
        ft.replace(R.id.frame, new TakeVidFragment(), "VIDEO");
        ft.commit();
        ft.addToBackStack(null);
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}

public void RekamVideo() {
    try {
        Intent captureVideoIntent = new
Intent(android.provider.MediaStore.ACTION_VIDEO_CAPTURE);
        startActivityForResult(captureVideoIntent, 1);
        File mediaFile = new
File(Environment.getExternalStorageDirectory().getAbsolutePath() + "/foto/tmp_v.mp4");
        Uri videoUri = Uri.fromFile(mediaFile);
        captureVideoIntent.putExtra(MediaStore.EXTRA_OUTPUT, videoUri);
        startActivityForResult(captureVideoIntent, 1);
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    try {
        if (requestCode == 1 && resultCode == RESULT_OK) {
            String imageFolderPath =
Environment.getExternalStorageDirectory().toString() + "/foto";
            Uri videoFileUri = data.getData();
            String video_name = "tmp_v.mp4";
            String full_path = imageFolderPath + "/" + video_name;

            Log.e("onActivityResult", full_path);
            File VideoFile = new File(full_path);
            if (VideoFile.exists()) {
                Toast.makeText(this, "Video has been saved to:\n" +
data.getData(), Toast.LENGTH_LONG).show();
            }
        }
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
}
}

```

PENJELASAN

```

public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    public Util util = new Util(MainActivity.this);
}

```

Pada contoh di atas, kita buat kelas MainActivity yang merupakan kelas turunan dari AppCompatActivity.

```
public Util util = new Util(MainActivity.this);
```

Disini kita membuat objek baru dari kelas Util yang akan kita gunakan untuk merequest permissi dari MainActivity

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    util.askForPermission(CAMERA, 0);  
    util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);  
    util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);  
}
```

Jika versi android adalah marshmallow ke atas, maka dilakukan pemanggilan metode askForPermission dari kelas Util yang akan menampilkan pop up untuk meminta izin permissi baca tulis file system android.

```
FragmentTransaction ft = getSupportFragmentManager().beginTransaction();  
    ft.replace(R.id.frame, new TakeVidFragment(), "VIDEO");  
    ft.commit();  
    ft.addToBackStack(null);
```

Selanjutnya dilakukan transaksi fragment untuk menggantikan fragment dengan id frame dengan fragment TakeVidFragment.

Metode RekamVideo digunakan untuk merekam video dari kamera dengan menggunakan Intent.

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    try {
```

Selanjutnya pada onActivityResult hasil dari perekaman video disimpan pada file system pada folder foto :

```
String imageFolderPath = Environment.getExternalStorageDirectory().toString() +  
"/foto";
```

Selanjutnya buat kelas baru dengan nama Util dengan cara klik menu File → New → Java Class lalu beri nama Util.

Isi file Util.java :

```
package com.ringlayer.capvid;  
import android.app.Activity;  
import android.content.pm.PackageManager;  
import android.os.StrictMode;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.content.ContextCompat;  
/**  
 * Created by ringlayer on 24/11/18.  
 */  
public class Util {  
    Activity mContext;
```

```

public Util (Activity context)
{
    mContext = context;
    StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
    StrictMode.setVmPolicy(builder.build());
    if (android.os.Build.VERSION.SDK_INT > 9) {
        StrictMode.ThreadPolicy policy =
            new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);}
}

public void askForPermission(String permission, Integer requestCode) {
    if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {

        if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
permission)) {

            ActivityCompat.requestPermissions(mContext, new String[]{permission},
requestCode);
        } else {
            ActivityCompat.requestPermissions(mContext, new String[]{permission},
requestCode);
        }
    }
}
}

```

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```

if (ContextCompat.checkSelfPermission(mContext, permission) !=
PackageManager.PERMISSION_GRANTED) {

```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```

ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);

```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Berikut ini isi dari TakeVidFragment.java :

```

package com.ringlayer.capvid;
import android.content.Context;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
/*
 * Created by ringlayer on 24/11/18.
 */
public class TakeVidFragment extends Fragment {

```

```

private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
private String mParam1;
private String mParam2;
private OnFragmentInteractionListener mListener;
public TakeVidFragment() {
}
public static TakeVidFragment newInstance(String param1, String param2) {
    TakeVidFragment fragment = new TakeVidFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_take_vid, container, false);
    try {
        Button b = (Button) v.findViewById(R.id.tombol_foto);
        boolean front_cam_enable = CheckCamera("front");
        if (front_cam_enable == true) {
            Log.e("onCreate : ", "front camera enabled");
        }
        boolean rear_cam_enable = CheckCamera("rear");
        if (rear_cam_enable == true) {
            Log.e("onCreate : ", "rear camera enabled");
        }
        if (front_cam_enable == true || rear_cam_enable == true) {
            Button button = (Button) v.findViewById(R.id.tombol_foto);
            button.setOnClickListener(new View.OnClickListener()
            {
                @Override
                public void onClick(View v)
                {
                    ((MainActivity) getActivity()).RekamVideo();
                }
            });
        }
    }
    catch (Exception e) {
        Log.e("Err : ", e.toString());
    }
    return v;
}
public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}
@Override
public void onAttach(Context context) {
    super.onAttach(context);
}
@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}
public interface OnFragmentInteractionListener {

```



```

        void onFragmentInteraction(Uri uri);
    }
    private boolean CheckCamera(String front_or_back) {
        boolean res;
        if (front_or_back == "front") {
            res =
getActivity().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_FRONT);
        }
        else {
            res =
getActivity().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA);
        }
        return res;
    }
}

```

PENJELASAN

```
public class TakeVidFragment extends Fragment {
```

Disini kita membuat kelas baru dengan nama TakeVidFragment yang merupakan kelas turunan dari Fragment.

```
public static TakeVidFragment newInstance(String param1, String param2) {
```

di sini kita siapkan Instance baru dari kelas TakeVidFragment

```

@Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

```

pada bagian metode onCreateView kita override di mana kita tambahkan rutin untuk :

1. Mengecek kamera yang tersedia apakah tersedia kamera depan atau belakang

```

boolean front_cam_enable = CheckCamera("front");
if (front_cam_enable == true) {
    Log.e("onCreate : ", "front camera enabled");
}
boolean rear_cam_enable = CheckCamera("rear");
if (rear_cam_enable == true) {
    Log.e("onCreate : ", "rear camera enabled");
}

```

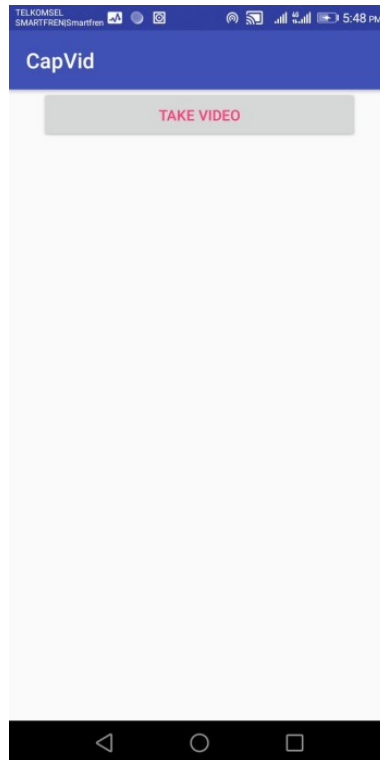
2. Selanjutnya jika kamera depan atau kamera belakang tersedia, maka akan dipasang callback setOnClickListener di mana parameter metode ini merupakan interface View.OnClickListener. Pada interface ini terdapat metode onClick. Pada contoh aplikasi kali ini kita override metode onClick dengan rutin untuk memanggil metode RekamVideo pada MainActivity.

```

if (front_cam_enable == true || rear_cam_enable == true) {
    Button button = (Button) v.findViewById(R.id.tombol_foto);
    button.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            ((MainActivity) getActivity()).RekamVideo();
        }
    });
}

```

Uji Coba Aplikasi

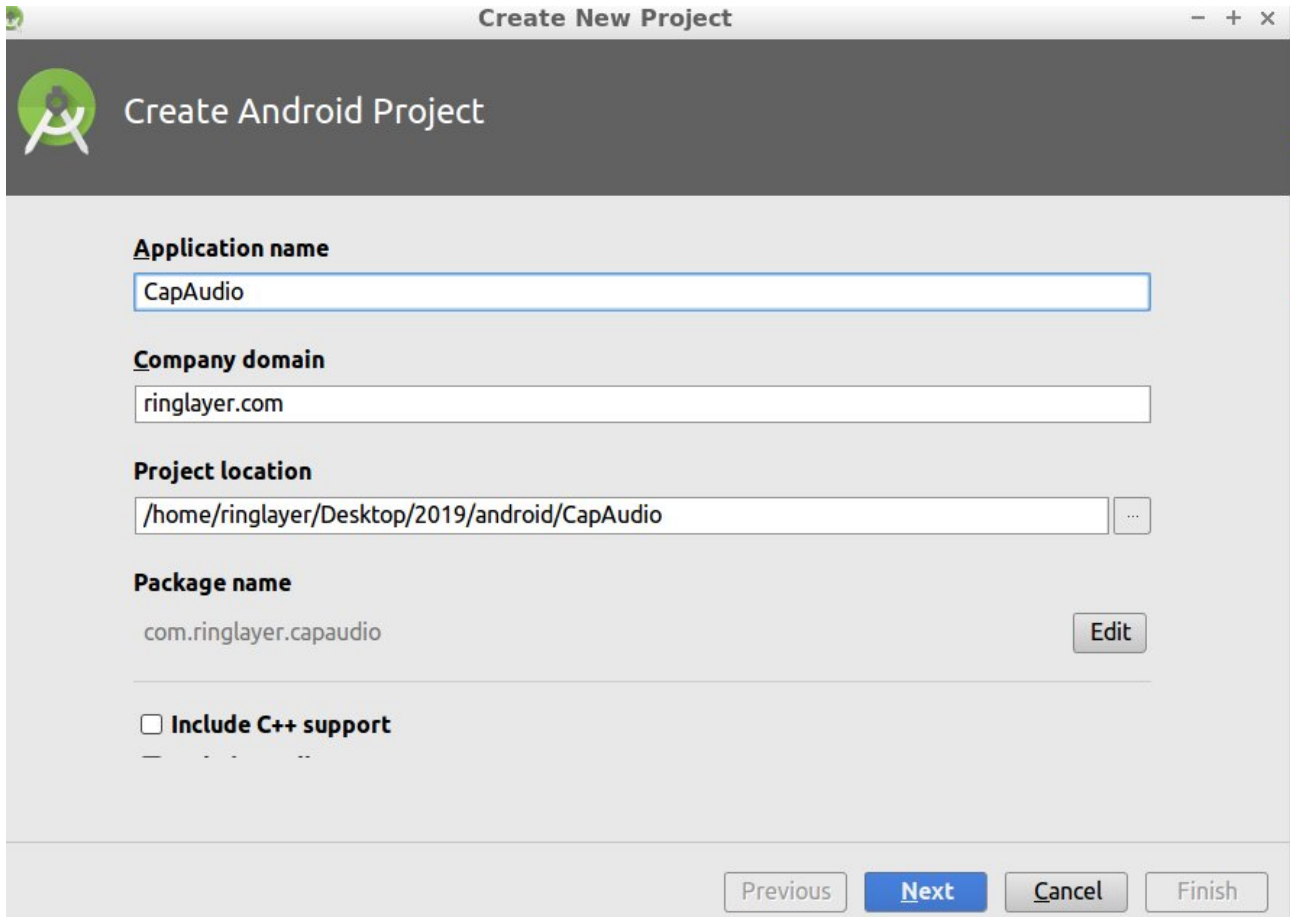


Tombol take video digunakan untuk mengambil video. Setelah selesai pengambilan video, akan muncul pesan toast.



G. Aplikasi Perekam Suara

Pada contoh kali ini, kita akan membuat proyek aplikasi sederhana untuk merekam suara dari mic android. Pertama tama buat dahulu proyek baru dengan nama “CapAudio”



The screenshot shows the 'Create New Project' window in Android Studio. The title bar says 'Create New Project'. The main area has a header 'Create Android Project' with the Android logo. Below are several fields: 'Application name' with 'CapAudio', 'Company domain' with 'ringlayer.com', 'Project location' with '/home/ringlayer/Desktop/2019/android/CapAudio', and 'Package name' with 'com.ringlayer.capaudio'. There is an 'Edit' button next to the package name. At the bottom, there is an unchecked checkbox for 'Include C++ support'. At the very bottom are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

Bagian XML

Pada contoh kali ini, kita akan menggunakan 2 layout yaitu activity_main.xml dan satunya lagi adalah fragment layout dengan nama fragment_take_audio.xml

Isi file activity_main.xml

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

```

<FrameLayout
    android:id="@+id/frame"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</FrameLayout>
</android.support.constraint.ConstraintLayout>

```

Isi file fragment_take_audio.xml

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.capaudio.TakeAudioFragment">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:id="@+id/background"
        android:gravity="center">
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:paddingTop="50dp"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            xmlns:tools="http://schemas.android.com/tools"
            android:orientation="vertical">
            <Button
                android:id="@+id/tombol_audio"
                android:layout_width="300dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:textColor="@color/colorAccent"
                android:text="Rekam Audio" />
            </LinearLayout>
        </LinearLayout>
    </FrameLayout>

```

Bagian Source Code

Berikut ini isi file Util.java

```

package com.ringlayer.capaudio;
import android.app.Activity;
import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.widget.Toast;
/*
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        mContext = context;
    }
}

```

```

        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }

        public void askForPermission(String permission, Integer requestCode) {
            if (ContextCompat.checkSelfPermission(mContext, permission) !=
                PackageManager.PERMISSION_GRANTED) {

                if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                    permission)) {

                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                } else {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                }
            }
        }
    }
}

```

PENJELASAN

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```

if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {

```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```

ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);

```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Berikut ini isi file TakeAudioFragment.java

```

package com.ringlayer.capaudio;
import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import com.ringlayer.capaudio.R;
/*
 * Created by ringlayer on 24/11/18.

```

```

*/
public class TakeAudioFragment extends Fragment {
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    private String mParam1;
    private String mParam2;
    private OnFragmentInteractionListener mListener;
    public TakeAudioFragment() {
    }
    public static TakeAudioFragment newInstance(String param1, String param2) {
        TakeAudioFragment fragment = new TakeAudioFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_take_audio, container, false);
        try {
            final Button b = (Button) v.findViewById(R.id.tombol_audio);
            b.setOnClickListener(new View.OnClickListener()
            {
                @Override
                public void onClick(View v)
                {
                    String aksi = b.getText().toString();
                    if (aksi != "Stop") {
                        b.setText("Stop");
                        ((MainActivity) getActivity()).RekamAudio();
                    }
                    else {
                        b.setText("Rekam Audio");
                        ((MainActivity) getActivity()).StopRekamAudio();
                    }
                }
            });
        }
        catch (Exception e) {
            Log.e("Err : ", e.toString());
        }
        return v;
    }
    public void onPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
    }
    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }
    public interface OnFragmentInteractionListener {
        void onFragmentInteraction(Uri uri);
    }
}

```

```
}  
}
```

```
public class TakeAudioFragment extends Fragment {
```

Pada rutin di atas, kita buat kelas baru dengan nama TakeAudioFragment yang merupakan turunan dari kelas Fragment.

Bagian terpenting pada kelas ini adalah pada bagian override metode onCreateView

```
@Override  
public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                          Bundle savedInstanceState) {  
    View v = inflater.inflate(R.layout.fragment_take_audio, container, false);  
    try {  
        final Button b = (Button) v.findViewById(R.id.tombol_audio);  
        b.setOnClickListener(new View.OnClickListener()  
        {  
            @Override  
            public void onClick(View v)  
            {
```

Pada button dengan id tombol_audio, dipasang metode callback setOnClickListener yang parameternya merupakan interface View.OnClickListener() (untuk abstraksi).

Metode onClick pada interface View.OnClickListener dioverride, di mana kita gunakan rutin di bawah ini yang akan ditrigger (dipanggil) saat tombol diklik:

```
String aksi = b.getText().toString();  
if (aksi != "Stop") {  
    b.setText("Stop");  
    ((MainActivity) getActivity()).RekamAudio();  
}  
else {  
    b.setText("Rekam Audio");  
    ((MainActivity) getActivity()).StopRekamAudio();  
}
```

Pada rutin di atas, variabel String aksi akan diisi dengan teks yang sedang tampil di button b yang akan didapat dari getText().toString();

Jika aksi bukan string "Stop: :

```
if (aksi != "Stop") {
```

maka pada tombol b akan diset atribut textnya menjadi Stop:

```
b.setText("Stop");
```

dan dilakukan pemanggilan metode RekamAudio pada MainActivity

Sebaliknya button b akan diset atribut textnya menjadi Rekam Audio dan diikutin pemanggilan metode StopRekamAudio pada MainActivity.

Berikut ini isi file MainActivity.java

```
package com.ringlayer.capaudio;  
import android.Manifest;  
import android.content.Intent;
```



```

import android.media.MediaRecorder;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;
import java.io.File;
import static android.Manifest.permission.CAMERA;
/*
 * Created by ringlayer on 24/11/18.
 */
public class MainActivity extends AppCompatActivity {
    private MediaRecorder recorder = new MediaRecorder();
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            /*
             * if (Build.VERSION.SDK_INT >= 23) {
             */
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                util.askForPermission(Manifest.permission.RECORD_AUDIO, 0);
                util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
                util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
            }
            FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
            ft.replace(R.id.frame, new TakeAudioFragment(), "AUDIO");
            ft.commit();
            ft.addToBackStack(null);
        }
        catch (Exception e) {
            Log.e("Err : ", e.toString());
        }
    }
    public void RekamAudio() {
        try {
            String path = Environment.getExternalStorageDirectory().getAbsolutePath() +
            "/foto/tmp_a.mp3";
            File mediaFile = new File(path);
            //recorder.setMaxDuration(300000);
            recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
            recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
            recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
            recorder.setOutputFile(path);
            recorder.prepare();
            recorder.start();
            Toast.makeText(MainActivity.this, "Recording Started",
                Toast.LENGTH_LONG).show();
        }
        catch (Exception e) {
            Log.e("Err : ", e.toString());
        }
    }
    public void StopRekamAudio() {
        try {
            String path = Environment.getExternalStorageDirectory().getAbsolutePath() +
            "/foto/tmp_a.mp3";
            File mediaFile = new File(path);
            recorder.stop();
            Toast.makeText(MainActivity.this, "Recording Completed, audio saved to : " +
            path,
                Toast.LENGTH_LONG).show();
        }
    }
}

```

```

        catch (Exception e) {
            Log.e("Err : ", e.toString());
        }
    }
}

```

PENJELASAN

Pada kelas MainActivity kita buat objek baru dari kelas MediaRecorder yang akan kita gunakan untuk merekam audio nanti :

```
private MediaRecorder recorder = new MediaRecorder();
```

Selanjutnya :

```

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    util.askForPermission(Manifest.permission.RECORD_AUDIO, 0);
    util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
    util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
}

```

Jika versi android adalah marshmallow ke atas, maka dilakukan pemanggilan metode askForPermission dari kelas Util yang akan menampilkan pop up untuk meminta izin permisi baca tulis file system android dan permisi untuk merekam audio.

```

FragmentManager ft = getSupportFragmentManager().beginTransaction();
ft.replace(R.id.frame, new TakeAudioFragment(), "AUDIO");
ft.commit();
ft.addToBackStack(null);

```

Selanjutnya dilakukan transaksi fragment untuk menggantikan fragment dengan id frame dengan fragment TakeAudioFragment.

```
public void RekamAudio() {
```

Pada metode RekamAudio kita isi dengan rutin untuk merekam audio dengan menggunakan objek recorder dari kelas MediaRecorder.

```

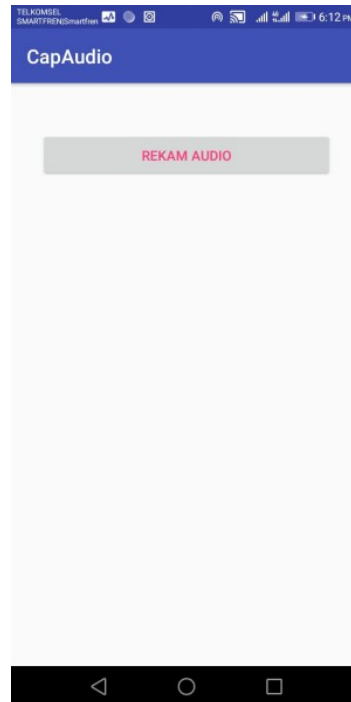
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
recorder.setOutputFile(path);
recorder.prepare();
recorder.start();

```

Pada rutin di atas, kita set properti untuk audio yang akan direkam dengan pemanggilan metode metode di dalam kelas MediaRecorder. Pada metode start() kita mulai perekaman audio.

Metode StopRekamAudio berguna untuk menghentikan proses perekaman audio.

Uji Coba Aplikasi



Untuk memulai merekam audio klik “REKAM AUDIO”. Setelah selesai klik tombol stop

H. Geolocation dan Google Map

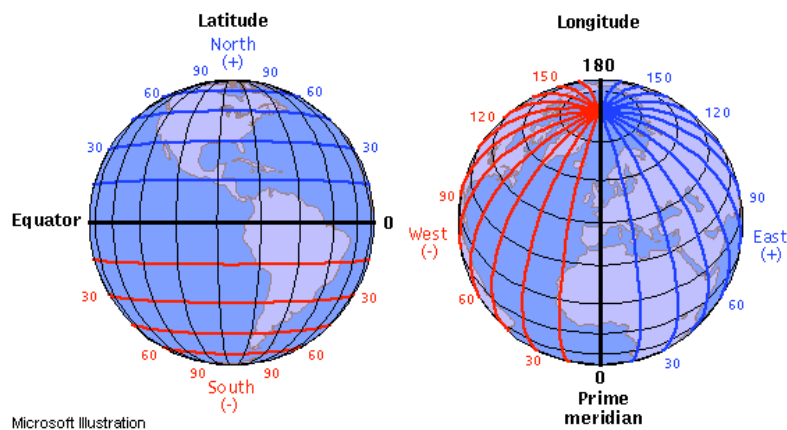
Pada bagian ini kita akan mencoba membuat aplikasi untuk geolocation dan google map. Untuk lebih detailnya kali ini kita akan membuat 2 aplikasi :

1. Aplikasi yang Memanfaatkan Sensor GPS pada Android
2. Aplikasi yang Memanfaatkan Google Map API

Aplikasi Pertama : Aplikasi yang Memanfaatkan Sensor GPS pada Android

Pada contoh pertama ini, kita akan membuat aplikasi yang menggunakan sensor GPS pada android. Pada aplikasi ini kita akan mencoba menggunakan class locationManager untuk mendapatkan data gps berupa latitude dan longitude.

Pada dasarnya latitude dan longitude didapat dari sistem pemetaan grid.



Latitude dan Longitude ini bisa diterjemahkan ke map. Untuk menguji bisa dites di web <https://www.latlong.net>

Buat proyek baru dengan nama GpsSample. Pada contoh kali ini kita memerlukan request permisi ACCESS_FINE_LOCATION pada AndroidManifest.xml :

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

CARA KERJA APLIKASI

Proyek ini memerlukan 2 activity dan 1 java class :

1. MainActivity (merupakan activity launcher)
Nama layout untuk MainActivity adalah activity_main.xml

2. GpsActivity (akan dijalankan oleh MainActivity dalam jangka waktu 4 detik sejak activity_main selesai dirender pada onCreate)

Nama layout untuk GpsActivity adalah activity_gps.xml

3. Util (merupakan kelas yang berguna untuk metode request permissi gps)

Berikut ini cara kerja aplikasi :

Pertama tama aplikasi ini akan menjalankan activity splash screen yaitu MainActivity, pada metode onCreate kita lakukan request permissi untuk mengakses GPS, 4 detik kemudian activity MainActivity akan menjalankan activity kedua yaitu GpsActivity.

Pada GpsActivity, lokasi data dari gps berupa latitude dan longitude akan ditampilkan pada layout.

Bagian XML

Berikut ini isi AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.gpssample">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".GpsActivity"></activity>
    </application>
</manifest>
```

Berikut ini isi activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.gpssample.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Loading GPS ..."
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Berikut ini isi activity_gps.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.ringlayer.gpssample.GpsActivity">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:gravity="center">
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
<TextView
    android:id="@+id/latitude"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:background="@color/colorPrimary"
    android:textColor="#ffffff"
/>
<Space
    android:layout_width="1dp"
    android:layout_height="30dp"/>
<TextView
    android:id="@+id/longitude"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:background="@color/colorPrimary"
    android:textColor="#ffffff"
/>
<Space
    android:layout_width="1dp"
    android:layout_height="50dp"/>
<Button
    android:id="@+id/update"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="Update"
/>
</LinearLayout>
</android.support.constraint.ConstraintLayout>
```

Bagian Source Code

Pada aplikasi kali ini, kita kembali memerlukan kelas Util. Buat kelas baru dengan nama Util

Util.java :

```
package com.ringlayer.gpssample;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.util.Log;
/**
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;
    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        /*
        * taken from
        * https://www.sitepoint.com/requesting-runtime-permissions-in-android-m-and-n/
        */
        public void askForPermission(String permission, Integer requestCode) {
            try {
                if (ContextCompat.checkSelfPermission(mContext, permission) !=
                    PackageManager.PERMISSION_GRANTED) {
                    if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                        permission)) {
                        ActivityCompat.requestPermissions(mContext, new String[]{permission},
                            requestCode);
                    } else {
                        ActivityCompat.requestPermissions(mContext, new String[]{permission},
                            requestCode);
                    }
                }
            } catch (Exception e) {
                Log.e("error", e.toString());
            }
        }
    }
}
```

PENJELASAN

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```
if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {
```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```
ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);
```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Selanjutnya, berikut ini isi MainActivity.java :

```
package com.ringlayer.gpssample;
import android.Manifest;
import android.content.Intent;
import android.os.Build;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            util.askForPermission(Manifest.permission.ACCESS_FINE_LOCATION, 1);
        }
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(MainActivity.this, GpsActivity.class);
                startActivity(intent);
                finish();
            }
        }, 4000);
    }
}
```

PENJELASAN

```
public Util util = new Util(MainActivity.this);
```

Pada baris di atas kita siapkan satu instance objek dari kelas Util yang nanti akan digunakan untuk merequest permissi.

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    util.askForPermission(Manifest.permission.ACCESS_FINE_LOCATION, 1);
}
```


Jika versi android lebih dari android marshmallow, maka request permisi untuk mengakses fitur GPS.

```
Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        Intent intent = new Intent(MainActivity.this, GpsActivity.class);
        startActivity(intent);
        finish();
    }
}, 4000);
```

Setelah 4 detik, aktifkan activity GpsActivity dengan intent.

Selanjutnya, berikut ini isi GpsActivity.java :

```
package com.ringlayer.gpssample;
import android.Manifest;
import android.content.Context;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class GpsActivity extends AppCompatActivity {
    public Util util = new Util(GpsActivity.this);
    TextView tekslatitude;
    TextView tekslongitude;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gps);
        try {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                util.askForPermission(Manifest.permission.ACCESS_FINE_LOCATION, 1);
            }
            GetLocation(GpsActivity.this);
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    public void GetLocation(Context myContext) {
        try {
            LocationManager lm;
            Location l;
            String provider;
            double lng = 0;
            double lat = 0;
            String lat_str;
            String long_str;
            lm = (LocationManager) myContext.getSystemService(Context.LOCATION_SERVICE);
            Criteria c = new Criteria();
            provider = lm.getBestProvider(c, false);
            l = lm.getLastKnownLocation(provider);
            if (l != null) {
```

```

        lng = l.getLongitude();
        lat = l.getLatitude();
        long_str = "Longitude : " + String.valueOf(lng);
        lat_str = "Latitude : " + String.valueOf(lat);
        tekslatitude = (TextView) findViewById(R.id.latitude);
        tekslatitude.setText(long_str);
        Log.d("long", long_str);
        tekslongitude = (TextView) findViewById(R.id.longitude);
        tekslongitude.setText(lat_str);
        Log.d("lat", lat_str);
    }
}
catch (Exception e) {
    Log.e("error", e.toString());
}
}
@Override
protected void onStart() {
    super.onStart();
    try {
        final Button update = (Button) findViewById(R.id.update);
        update.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                GetLocation(GpsActivity.this);
            }
        });
    }
    catch (Exception e) {
        Log.e("error", e.toString());
    }
}
}
}

```

PENJELASAN

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

```

Metode onCreate kita override, pada baris bawahnya kita gunakan keyword super agar rutin di tubuh metode onCreate dari kelas AppCompatActivity dieksekusi terlebih sebelum rutin yang kita tambahkan.

```

try {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        util.askForPermission(Manifest.permission.ACCESS_FINE_LOCATION, 1);
    }
    GetLocation(GpsActivity.this);
}

```

Pada rutin di atas, kita melakukan request permissi jika versi android marshmallow ke atas.

Metode GetLocation digunakan untuk mengupdate Longitude dan Latitude pada TextView.

Metode GetLocation

Metode ini merupakan metode yang kita buat untuk mengupdate TextView dengan latitude dan longitude.

```
public void GetLocation(Context myContext) {  
    try {
```

Metode GetLocation memerlukan parameter berupa konteks activity yang aktif.

```
LocationManager lm;  
Location l;
```

Pada dua baris ini kita siapkan objek LocationManager dan objek Location.

```
lm = (LocationManager) myContext.getSystemService(Context.LOCATION_SERVICE);  
Criteria c = new Criteria();  
provider = lm.getBestProvider(c, false);  
l = lm.getLastKnownLocation(provider);
```

Pada akhir rutin data lokasi akan tersimpan pada objek location, kemudian akan diretrieve ke dalam variabel bertipe double lng dan lat :

```
lng = l.getLongitude();  
lat = l.getLatitude();
```

Selanjutnya

```
long_str = "Longitude : " + String.valueOf(lng);  
lat_str = "Latitude : " + String.valueOf(lat);
```

Latitude dan Longitude dikonversi dari double ke String, datanya kemudian disimpan pada variabel bertipe String.

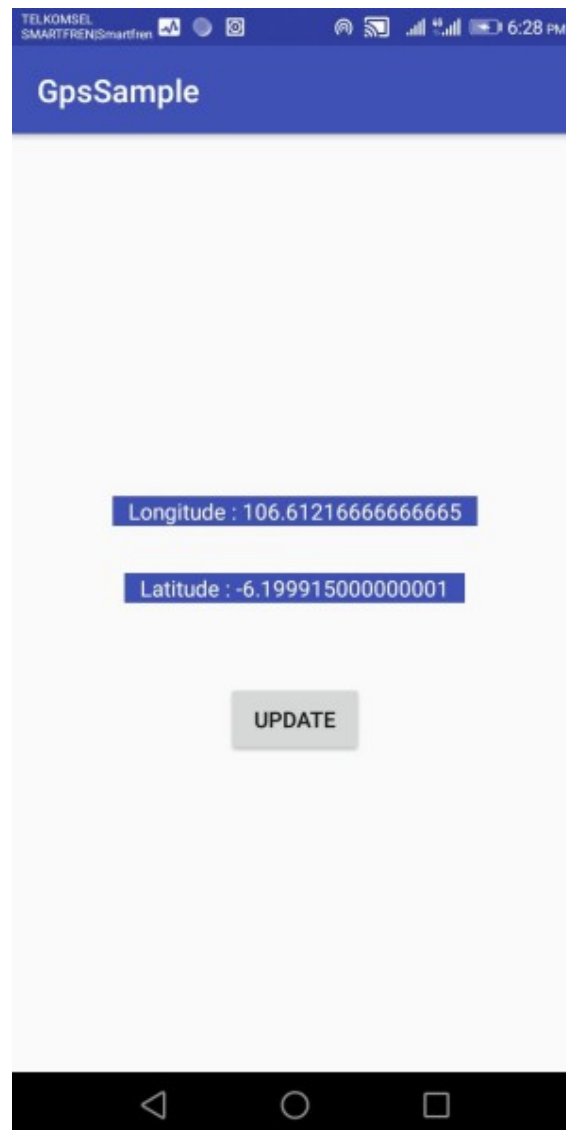
Selanjutnya :

```
tekslatitude = (TextView) findViewById(R.id.latitude);  
tekslatitude.setText(long_str);  
Log.d("long", long_str);  
tekslongitude = (TextView) findViewById(R.id.longitude);  
tekslongitude.setText(lat_str);  
Log.d("lat", lat_str);
```

Simpan referensi untuk TextView latitude dan longitude ke variabel bertipe TextView. Kemudian dengan setText kita update isi teks pada objek TextView dengan latitude dan longitude yang didapat dari metode getLastKnownLocation tadi.

Uji Coba Aplikasi

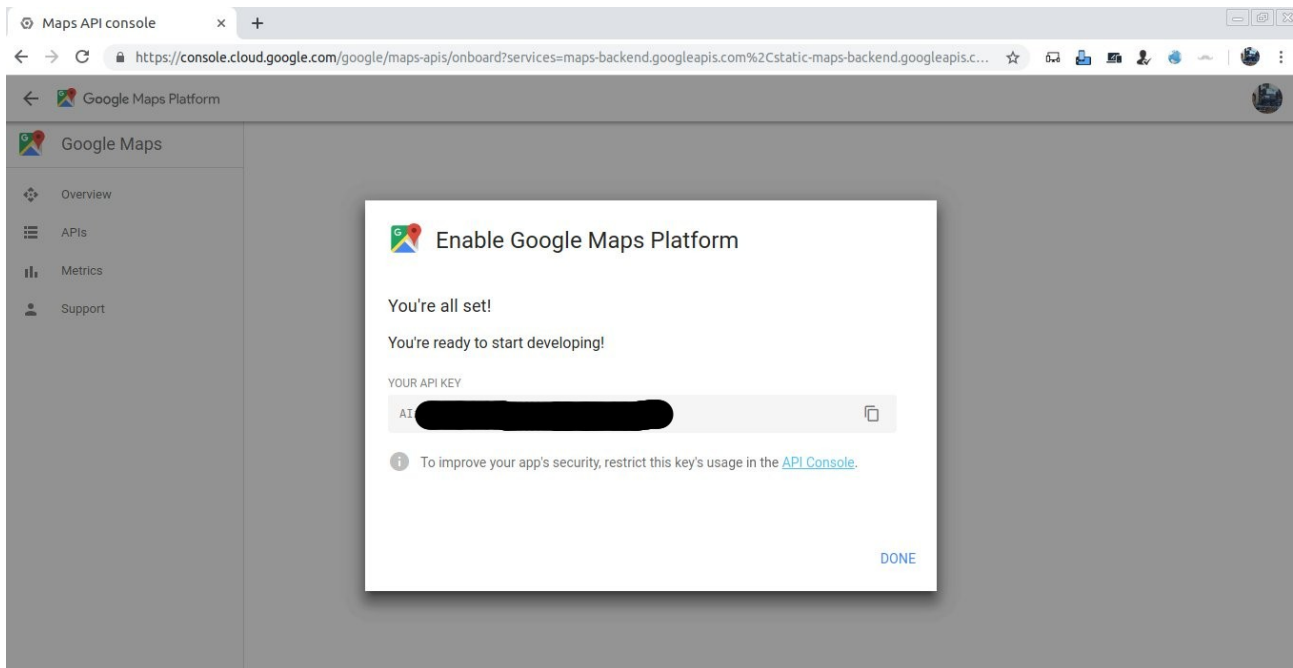
Ketika aplikasi dijalankan maka muncul splash screen selama 4 detik, setelah itu muncul activity yang meload data latitude dan longitude



Aplikasi Kedua : Aplikasi yang Memanfaatkan Google Map API

Pada contoh kedua ini, kita akan membuat aplikasi android yang menggunakan api dari google map.

Sebelum memulai kita perlu mendaftar api dari google. Untuk pendaftaran ini, memerlukan data kartu kredit. Jika berhasil maka Anda akan mendapatkan api key.



Pada contoh kali ini api yang diaktifkan adalah api untuk map. Berikut ini daftar api map yang tersedia di google :

APIs

LEARN

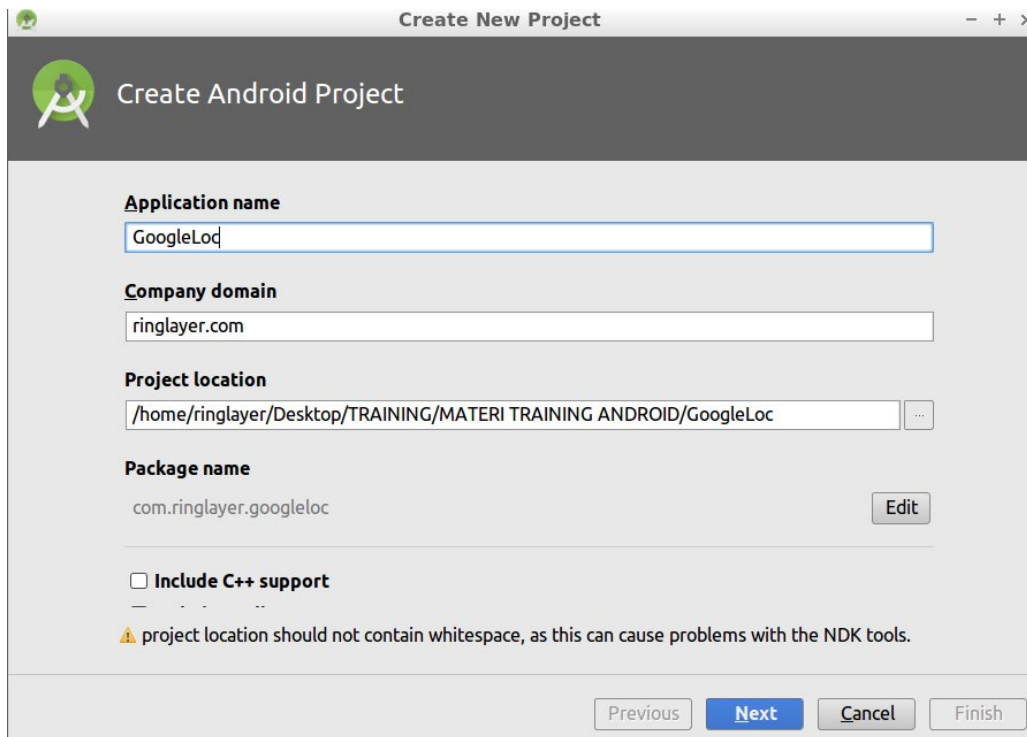
Enabled APIs

Select an API to view details. Figures are for the last 30 days.

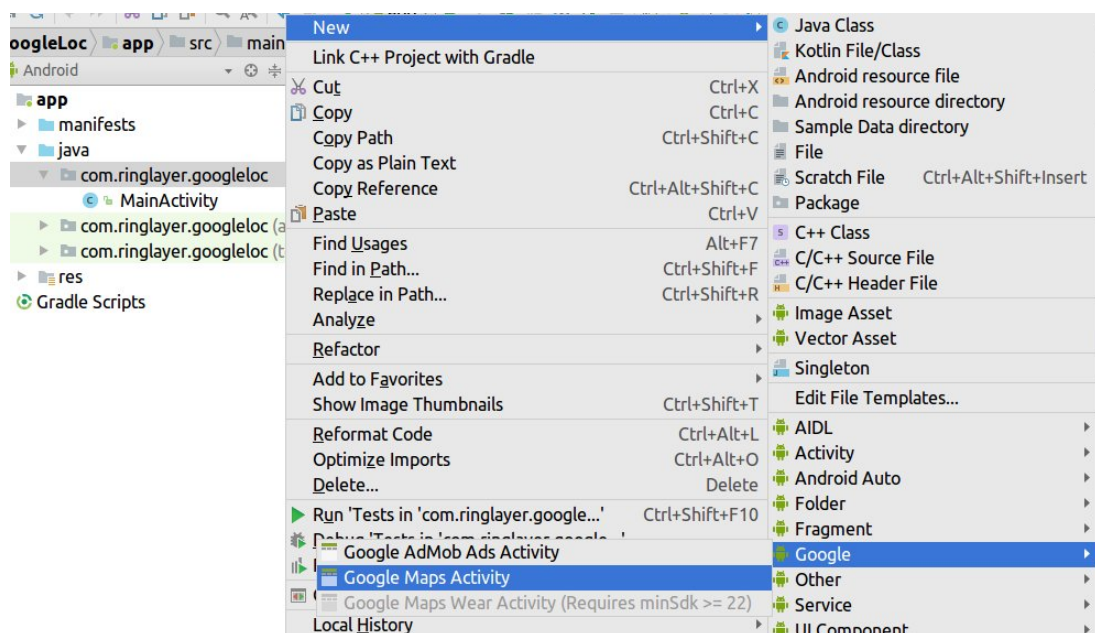
API ↑	Requests	Errors	Avg latency (ms)	
Maps Embed API	0	0	-	Details
Maps JavaScript API	0	0	-	Details
Maps SDK for Android	0	0	-	Details
Maps SDK for iOS	0	0	-	Details
Maps Static API	0	0	-	Details
Street View Static API	0	0	-	Details

Additional APIs

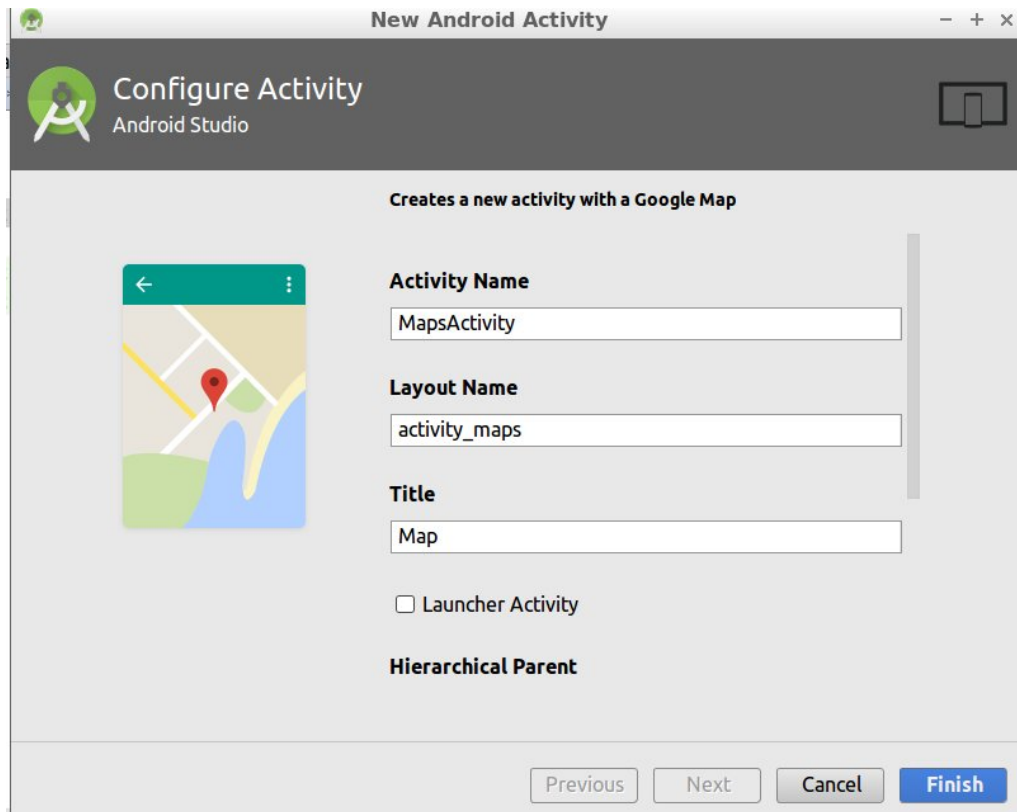
Selanjutnya, buat aplikasi baru dengan nama GoogleLoc



Pada aplikasi ini, MainActivity akan kita gunakan sebagai splash screen. Langkah selanjutnya adalah tambah activity google map dengan cara klik kanan pada direktori proyek lalu pilih New → Google lalu pilih Google Maps Activity.



Selanjutnya akan muncul jendela untuk menambah activity:



Selanjutnya klik tombol “Finish”

Pada activity ini kita memerlukan 1 class lagi yaitu class Util. Buat kelas baru dengan nama kelas : Util.

Bagian XML

Aplikasi ini memerlukan request permisi :

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

Berikut ini isi lengkap AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.googleloc">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </manifest>
```

```

        android:name="com.google.android.geo.API_KEY"
        android:value="@string/google_maps_key" />
    <activity
        android:name=".MapsActivity"
        android:label="@string/title_activity_maps"></activity>
</application>
</manifest>

```

Berikut ini isi activity_main.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.googleloc.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Loading Map ..."
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>

```

Berikut ini isi activity_maps.xml :

```

<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ringlayer.googleloc.MapsActivity" />

```

Pada saat membuat activity google map tadi akan ada 1 file xml lagi untuk mengisi api google map kita. Nama file tersebut adalah google_maps_api.xml

Berikut ini isi google_maps_api.xml :

```

<resources>
    <string name="google_maps_key" templateMergeStrategy="preserve"
    translatable="false">ISIKAN API DARI GOOGLE DI SINI</string>
</resources>

```

Isikan api yang didapat pada saat pendaftaran ke google tadi di file tersebut.

Bagian Source Code

Berikut ini isi file Util.java :

```
package com.ringlayer.googleloc;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.util.Log;
/**
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;
    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        /*
        * taken from
        * https://www.sitepoint.com/requesting-runtime-permissions-in-android-m-and-n/
        */
        public void askForPermission(String permission, Integer requestCode) {
            try {
                if (ContextCompat.checkSelfPermission(mContext, permission) !=
                    PackageManager.PERMISSION_GRANTED) {
                    if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                        permission)) {
                        ActivityCompat.requestPermissions(mContext, new String[]{permission},
                            requestCode);
                    } else {
                        ActivityCompat.requestPermissions(mContext, new String[]{permission},
                            requestCode);
                    }
                }
            } catch (Exception e) {
                Log.e("error", e.toString());
            }
        }
    }
}
```

PENJELASAN

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```
if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {
```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```
ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);
```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permisi.

Berikut ini isi file MainActivity.java :

```
package com.ringlayer.googleloc;
import android.Manifest;
import android.content.Intent;
import android.os.Build;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            util.askForPermission(Manifest.permission.ACCESS_FINE_LOCATION, 1);
            util.askForPermission(Manifest.permission.ACCESS_COARSE_LOCATION, 2);
        }
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(MainActivity.this, MapsActivity.class);
                startActivity(intent);
                finish();
            }
        }, 3000);
    }
}
```

PENJELASAN

Activity MainActivity akan merequest permisi untuk gps dan selanjutnya akan menjalankan activity selanjutnya yaitu MapsActivity dalam waktu 3 detik setelah popup request permisi muncul.

Berikut ini isi file MapsActivity.java :

```
package com.ringlayer.googleloc;
import android.content.Context;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
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;
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
    private GoogleMap mMap;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
    }
}
```

```

        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
        .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        Context myContext = MapsActivity.this;
        LocationManager lm;
        Location l;
        String provider;
        double lng = 0;
        double lat = 0;
        lm = (LocationManager) myContext.getSystemService(Context.LOCATION_SERVICE);
        Criteria c = new Criteria();
        provider = lm.getBestProvider(c, false);
        l = lm.getLastKnownLocation(provider);
        if (l != null) {
            lng = l.getLongitude();
            lat = l.getLatitude();
        }
        mMap = googleMap;
        LatLng curpos = new LatLng(lat, lng);
        mMap.addMarker(new MarkerOptions().position(curpos).title("current position"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(curpos));
    }
}

```

PENJELASAN

Pada contoh di atas, kita akan melakukan override pada onMapReady

```

@Override
    public void onMapReady(GoogleMap googleMap) {

```

Di mana saat google map selesai diloat, kita akan menambahkan marker untuk posisi kita saat ini.

```

lm = (LocationManager) myContext.getSystemService(Context.LOCATION_SERVICE);
Criteria c = new Criteria();
provider = lm.getBestProvider(c, false);
l = lm.getLastKnownLocation(provider);
if (l != null) {
    lng = l.getLongitude();
    lat = l.getLatitude();
}

```

Hasil pembacaan dari receiver gps akan disimpan dalam variabel double lng dan lat.

Selanjutnya adalah membuat marker di google map untuk latitude dan longitude yang didapat dari pembacaan gps receiver dan memindahkan posisi map ke area map sesuai latitude dan longitude :

```

mMap = googleMap;
LatLng curpos = new LatLng(lat, lng);
mMap.addMarker(new MarkerOptions().position(curpos).title("current position"));
mMap.moveCamera(CameraUpdateFactory.newLatLng(curpos))

```

Uji Coba Aplikasi



I. Pengambilan Data Json dari Internet untuk Ditampilkan di Layout

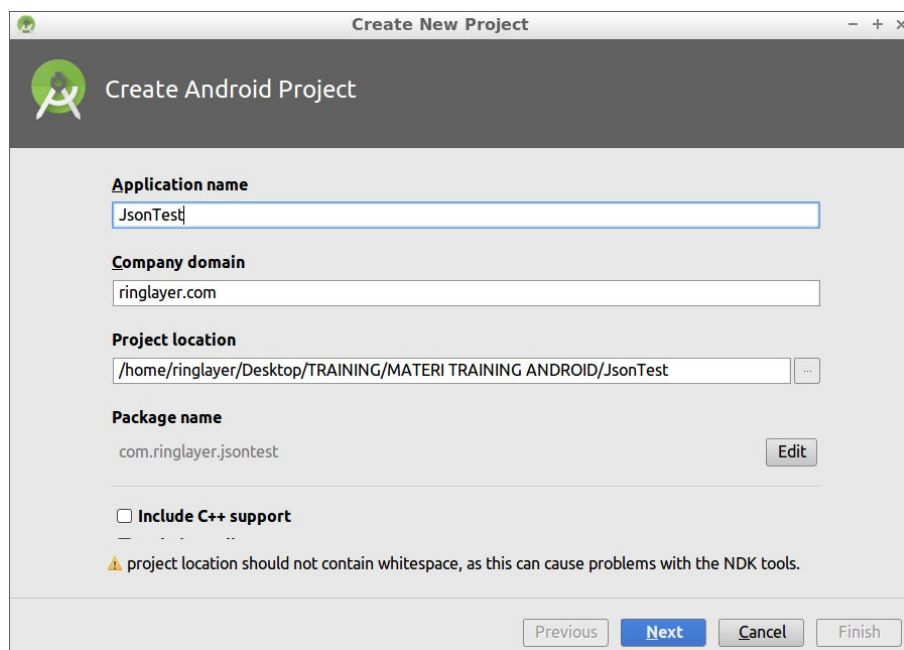
Pada bagian ini kita akan mencoba membuat aplikasi android untuk mengambil data json dari internet untuk kemudian ditampilkan berupa teks di layout android.

Json yang akan diambil adalah dari url

<https://raw.githubusercontent.com/jasapluscom/snippets/master/page.json>

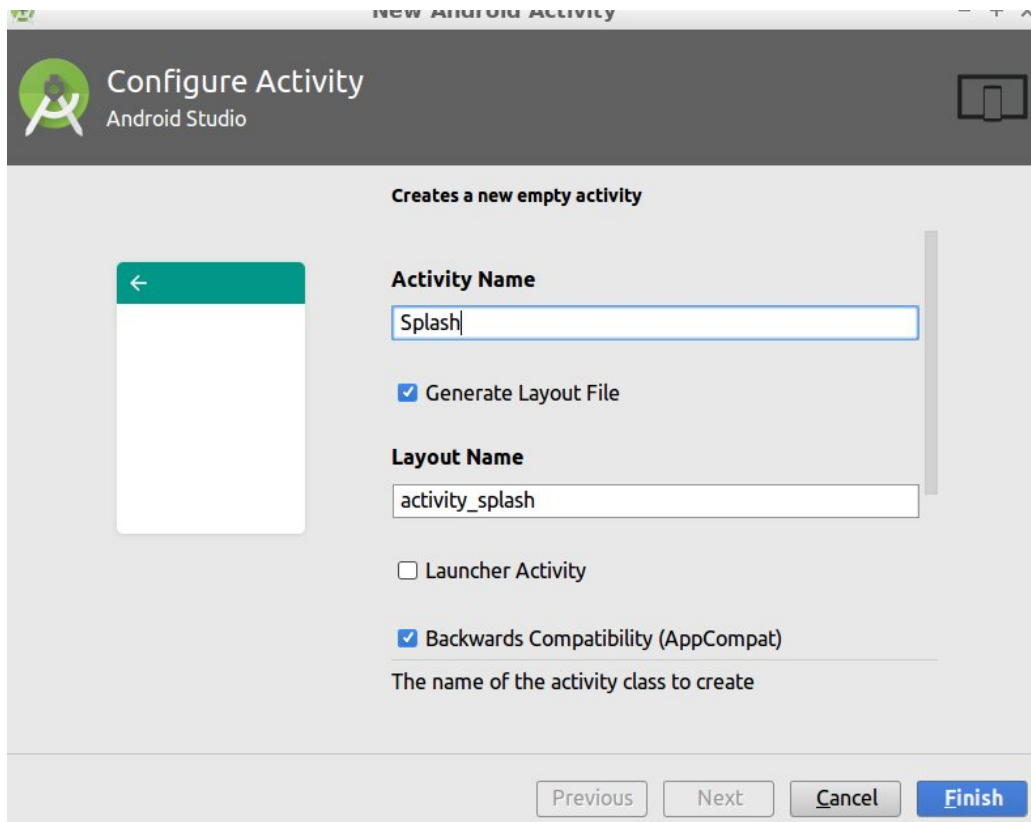
di mana json di atas akan dibaca dan diparse untuk kemudian ditampilkan di layout.

Buat proyek baru di android studio dengan nama JsonTest



Bagian XML

Buat activity baru dengan nama Splash, centang pada bagian checkbox “Launcher Activity”



Pada dasarnya splash di sini berguna untuk menjalankan servis di background yang akan digunakan untuk membaca, melakukan parse data json dari internet kemudian ditampilkan di layout activity_main yang akan ditampilkan setelah loading splash ini selesai. Selanjutnya buat Servis baru dengan nama kelas Servis.

Aplikasi ini memerlukan permisi internet, tambahkan line ini pada AndroidManifest.xml

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

Berikut ini isi lengkap AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.jsonetest">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <service android:name=".Servis" />
        <activity android:name=".MainActivity">
        </activity>
        <activity android:name=".Splash">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Pada folder Res → values, Buka file strings.xml lalu tambahkan baris ini :

```
<string
name="url">https://raw.githubusercontent.com/jasapluscom/snippets/
master/page.json</string>
```

Berikut ini isi lengkap strings.xml

```
<resources>
    <string name="app_name">JsonTest</string>
    <string name="url">https://raw.githubusercontent.com/jasapluscom/snippets/master/
page.json</string>
</resources>
```

Berikut ini isi xml activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.ringlayer.jsontest.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/json"
        android:singleLine="false"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Berikut ini isi xml activity_splash.xml :

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

Bagian Source Code

Servis “Servis” akan dijalankan oleh Splash Activity pada saat activity ini dilaunch. Servis ini bertujuan untuk mengambil data json dari halaman web untuk kemudian dibroadcast ke broadcast receiver yang kita register di MainActivity. Berikut ini isi Servis.java :

```
package com.ringlayer.jsontest;
import android.app.IntentService;
import android.content.Intent;
import android.os.IBinder;
import android.os.StrictMode;
import android.util.Log;
import java.io.BufferedReader;
import java.io.InputStream;
```

```

import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
/**
 * Created by ringlayer on 16/03/19.
 */
public class Servis extends IntentService {
    public Servis() {super("Servis");}
    @Override
    protected void onHandleIntent(Intent intent) {
        try {
            Log.e("servis ohi", "servis ohi");
            String base_url = getResources().getString(R.string.url);
            //get about
            Log.e("servis url", base_url);
            String json = getHtml(base_url);
            if (json.length() > 3) {
                Log.e("servis", json);
                publishResults(json);
            }
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    /* modified from https://stackoverflow.com/questions/2423498/how-to-get-the-html-
    source-of-a-page-from-a-html-link-in-android */
    public static String getHtml(String url) {
        try {
            if (android.os.Build.VERSION.SDK_INT > 9) {
                StrictMode.ThreadPolicy policy = new
                StrictMode.ThreadPolicy.Builder().permitAll().build();
                StrictMode.setThreadPolicy(policy);
                StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
                StrictMode.setVmPolicy(builder.build());
            }
            URLConnection connection = (new URL(url)).openConnection();
            connection.setConnectTimeout(5000);
            connection.setReadTimeout(5000);
            connection.connect();
            InputStream in = connection.getInputStream();
            BufferedReader reader = new BufferedReader(new InputStreamReader(in));
            StringBuilder html = new StringBuilder();
            for (String line; (line = reader.readLine()) != null; ) {
                html.append(line);
            }
            in.close();
            return html.toString();
        }
        catch (Exception e) {
            Log.e("[+] getHtml", e.toString());
            return "";
        }
    }
    private void publishResults(String pesan) {
        try {
            Log.e("[+] publishResults", pesan);
            Intent intent = new Intent("1");
            intent.putExtra("BROADCAST", pesan);
            sendBroadcast(intent);
        }
        catch (Exception e) {
            Log.e("publishResults", e.toString());
        }
    }
}

```


Metode `onHandleIntent` pada kelas `IntentService` akan kita override dengan rutin di bawah ini :

```
Log.e("servis ohi", "servis ohi");
```

Digunakan untuk menulis log saat `onHandleIntent` dijalankan.

```
String base_url = getResources().getString(R.string.url);
```

variabel `String base_url` diisi dengan string url pada `strings.xml`

```
Log.e("servis url", base_url);
String json = getHtml(base_url);
if (json.length() > 3) {
    Log.e("servis", json);
    publishResults(json);
}
```

Pada rutin di atas, kita fetch html dari url dan disimpan ke dalam variabel bertipe `String`. Untuk kemudian dibroadcast dengan metode `publishResults`

Metode `getHtml` berguna untuk mengambil data `String` html dari url.

Pada body metode `getHtml` :

```
StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
```

kita lakukan pencegahan error karena blocking pada saat http request dengan memanfaatkan fitur `StrictMode`.

```
URLConnection connection = (new URL(url)).openConnection();
connection.setConnectTimeout(5000);
connection.setReadTimeout(5000);
connection.connect();
```

Selanjutnya kita siapkan objek `URLConnection` yang merupakan hasil dari pemanggilan metode `openConnection`. Pada baris baris selanjutnya kita atur properti untuk http request berupa connect timeout dan readtimeout, pada akhirnya koneksi ke server dilakukan dengan metode `connect()`.

```
InputStream in = connection.getInputStream();
BufferedReader reader = new BufferedReader(new InputStreamReader(in));
StringBuilder html = new StringBuilder();
for (String line; (line = reader.readLine()) != null; ) {
    html.append(line);
}
```

Pada rutin di atas kita siapkan objek berupa InputStream untuk menangkap response dari server. Selanjutnya hasil response disimpan pada variabel StringBuilder html.

Nilai kembali dari getHtml adalah variabel html yang diubah menjadi String :

```
return html.toString();
```

Metode publishResults digunakan untuk melakukan broadcast source code html. Broadcast ini akan ditangkap oleh broadcast receiver pada MainActivity.

Berikut ini isi Splash Activity (Splash.java) :

```
package com.ringlayer.jsonetest;
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
public class Splash extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_splash);
            Log.e("Splash", "Splash run");
            Intent intent = new Intent(Splash.this, Servis.class);
            startService(intent);
            stopService(intent);
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    Intent intent = new Intent(Splash.this, MainActivity.class);
                    startActivity(intent);
                    finish();
                }
            }, 1000);
        } catch (Exception e) {
            Log.e("Error", e.toString());
        }
    }
}
```

PENJELASAN

Pada Splash Activity, kita menjalankan servis untuk mengambil html dari url yang nanti akan dibroadcast dan ditangkap oleh MainActivity dan ditampilkan di TextView.

```
Intent intent = new Intent(Splash.this, Servis.class);
startService(intent);
stopService(intent);
```

Selanjutnya MainActivity akan dijalankan dengan startActivity setelah 1000 microseconds (1 detik)

```
Handler handler = new Handler();
```

```

        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(Splash.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, 1000);

```

MainActivity.java :

```

package com.ringlayer.jsonetest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Html;
import android.text.Spanned;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.TextView;
import org.json.JSONObject;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            Log.e("main run", "main run");
        }
        catch (Exception e) {
            Log.e("error:", e.toString());
        }
    }
    private BroadcastReceiver br = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String act = "";
            final Context konteks = context;
            try {
                Bundle bundle = intent.getExtras();
                if (bundle != null) {
                    act = bundle.getString("BROADCAST").trim();
                    Log.e("[+] pgonReceive", act);
                    JSONObject reader = new JSONObject(act);
                    String title_teks = reader.getString("title");
                    String contain_teks = reader.getString("contain");
                    String full_str = title_teks + "<br>" + contain_teks;
                    Log.e("title_teks", title_teks);
                    Log.e("contain_teks", contain_teks);
                    TextView json = (TextView) findViewById(R.id.json);
                    json.setText(Html.fromHtml(full_str));
                }
                else {
                    Log.e("fail", "no broadcast received");
                }
            }
            catch (Exception e) {
                Log.e("onReceive", e.toString());
            }
        }
    };
    @Override

```

```

protected void onStart() {
    super.onStart();
    registerReceiver(br, new IntentFilter("1"));
}
@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(br);
}
}

```

Pada MainActivity kita siapkan objek BroadcastReceiver

```
private BroadcastReceiver br = new BroadcastReceiver() {
```

Pada objek BroadcastReceiver br, kita melakukan override metode onReceive :

```

@Override
public void onReceive(Context context, Intent intent) {

```

di mana kita ambil pesan intent broadcast dengan rutin ini :

```
Bundle bundle = intent.getExtras();
```

Rutin selanjutnya di dalam tubuh onReceive berguna untuk melakukan parsing data html yang dibroadcast servis (data html berupa json)

```

JSONObject reader = new JSONObject(act);
String title_teks = reader.getString("title");
String contain_teks = reader.getString("contain");
String full_str = title_teks + "<br>" + contain_teks;
Log.e("title_teks", title_teks);
Log.e("contain_teks", contain_teks);

```

Selanjutnya hasil parsing akan ditampilkan pada TextView :

```

TextView json = (TextView) findViewById(R.id.json);
json.setText(Html.fromHtml(full_str));

```

Broadcast Receiver “br” kemudian kita register pada metode onStart yang kita override :

```

@Override
protected void onStart() {
    super.onStart();
    registerReceiver(br, new IntentFilter("1"));
}

```

penggunaan keyword super di atas akan menyebabkan rutin asli metode onStart pada kelas AppCompatActivity dieksekusi sebelum rutin yang kita tambahkan dibawahnya. Pada metode yang kita override kita tambahkan rutin untuk meregister broadcast receiver “br” yang tadi kita siapkan :

```
registerReceiver(br, new IntentFilter("1"));
```

Jika aplikasi dijalankan maka textview dengan id “json” di MainActivity akan terisi html yang telah dirender, di mana html ini berasal dari data json di url.

J. Aplikasi Menggunakan Webview

Pada contoh kali ini kita akan mencoba membuat aplikasi android untuk menampilkan web dengan alamat <https://www.olmyshop.com>

Buat proyek baru dengan nama “OlMyShop”.

Bagian XML

Setelah proyek siap. Buka, folder app, lalu klik “res” selanjutnya klik folder “values”, Lalu double click “strings.xml”

Isikan xml berikut ini:

```
<resources>
    <string name="app_name">OlMyShop</string>
    <string name="web_title">OlMyShop</string>
    <string name="web_url">https://www.olmyshop.com</string>
    <string name="title_activity_splash">OlMyShop</string>
</resources>
```

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

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

Berikut ini contoh isi styles.xml secara keseluruhan

```
<resources>
    <style name="AppTheme.Fullscreen">
        <item name="android:windowFullscreen">true</item>
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>
    <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar"
/>
    <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
</resources>
```

Pada aplikasi kali ini, kita akan menggunakan 3 layout yaitu:

1. activity_main.xml
2. activity_olmyshop.xml
3. activity_splash.xml

Cara kerja aplikasi ini adalah sebagai berikut:

Activity launcher pada aplikasi ini adalah activity OlMyShop dengan layout activity_olmyshop.xml di mana activity ini selanjutnya akan melaunch / menjalankan activity dengan nama Splash dengan layoutnya activity_splash.xml. Activity splash ini merupakan splash screen sebelum masuk ke webview. Setelah activity Splash diload maka dalam 1 detik kemudian activity ini melaunch / menjalankan activity yang berisi webview dengan nama activity MainActivity dengan layoutnya activity_main.xml.

activity_main.xml sudah terbentuk pada saat tadi kita membuat proyek olmyshop.

Berikut ini isi layout activity_main.xml

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

Perhatikan pada baris ini:

tools:context="com.ringlayer.OlMyShop.MainActivity">

edit / sesuaikan **com.ringlayer.OlMyShop** dengan nama paket aplikasi Anda.

Pada layout di atas kita menggunakan 1 webview :

```
<WebView
    android:id="@+id/webview"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Perhatian ! Pastikan layout parent yang akan digunakan menampilkan webview menggunakan constraint layout agar webview bisa dirender dengan sempurna. Pada contoh di atas kita menggunakan layout constraint layout:

```
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```

    android:layout_height="match_parent"
    tools:context="com.ringlayer.OlMyShop.MainActivity">

```

Selanjutnya kita akan menambahkan activity baru dengan nama activity : OlMyShop dengan layout activity_olmyshop.xml

Berikut ini contoh isi activity_olmyshop.xml

```

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

```

Terakhir, kita tambahkan satu activity yang berguna untuk menampilkan splash screen sebelum membuka activity yang meload website dengan webview. Buat Activity baru dengan nama Splash dengan layoutnya : activity_splash.xml

Isi activity_splash.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/background"
    android:background="@color/colorPrimary"
    tools:context="com.ringlayer.OlMyShop.Splash">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="vertical"
        android:background="@drawable/bg">
    </LinearLayout>
</LinearLayout>

```

Agar aplikasi bisa mengakses internet, Selanjutnya update AndroidManifest.xml, tambahkan baris ini:

```

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

```

di luar tag application.

Berikut ini contoh AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.OlMyShop">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:screenOrientation="portrait"
            android:windowSoftInputMode="stateUnchanged"
            android:theme="@style/AppTheme.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Splash"
            android:label="@string/title_activity_splash"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:theme="@style/AppTheme.Fullscreen"
            >
        </activity>
        <activity android:name=".OlMyShop"
            android:theme="@style/AppTheme.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Bagian Source Code

Berikut ini isi file activity launcher OlMyShop.java:

```

package com.ringlayer.OlMyShop;
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class OlMyShop extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_olmyshop);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(OlMyShop.this, Splash.class));
            }
        }, 0);
    }
}

```


PENJELASAN

Sesaat setelah activity launcher diload selanjutnya akan melaunch / menjalankan activity selanjutnya yaitu activity untuk splash screen dengan nama Splash :

```
startActivity(new Intent(0lMyShop.this,Splash.class));
```

Berikut ini isi file Splash.java

```
package com.ringlayer.0lMyShop;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;
public class Splash extends AppCompatActivity {
    final Activity activity = this;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        /*
        String the_title = getString(R.string.web_title);
        the_title = the_title.replaceAll("!", "");
        */
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(Splash.this, MainActivity.class));
            }
        }, 1000);
    }
}
```

PENJELASAN

```
public class Splash extends AppCompatActivity {
```

kelas Splash merupakan turunan dari kelas AppCompatActivity, di mana pada kelas ini kita override metode callback onCreate :

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

penggunaan keyword super akan menyebabkan rutin metode onCreate pada kelas asli dieksekusi terlebih dahulu sebelum rutin yang kita tambahkan di bawahnya.

```
setContentView(R.layout.activity_splash);
new Handler().postDelayed(new Runnable() {
    @Override
    public void run() {
        startActivity(new Intent(Splash.this, MainActivity.class));
    }
}, 1000);
```

Setelah merender layout activity_splash.xml, activity Splash akan menjalankan activity selanjutnya yaitu activity MainActivity setelah jangka waktu delay 1000 microseconds (1 detik)

Berikut ini isi file MainActivity.java

```
package com.ringlayer.OlMyShop;
import android.app.ActionBar;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Typeface;
import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.KeyEvent;
import android.view.View;
import android.view.Window;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ProgressBar;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    final Activity activity = this;
    /**
     * TextView t_url;
     * TextView t_title;
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            this.getWindow().requestFeature(Window.FEATURE_PROGRESS);
            setContentView(R.layout.activity_main);
            String url = "";
            WebView webView = (WebView) findViewById(R.id.webview);
            webView.getSettings().setJavaScriptEnabled(true);
            webView.getSettings().setAllowFileAccessFromFileURLs(true);
            webView.getSettings().setAllowUniversalAccessFromFileURLs(true);

            webView.setWebViewClient(new WebViewClient() {
                @Override
                public void onReceivedError(WebView view, int errorCode, String
description, String failingUrl)
                {
                    // Handle the error
                }
                @Override
                public boolean shouldOverrideUrlLoading(WebView view, String url)
                {
                    view.loadUrl(url);
                    return true;
                }
            });
            if (url.length() < 3) {
                String title = "OlMyShop";
                url = "https://www.olmyshop.com/";
            }
            else {
                if (url.indexOf("http") < 0) {
                    url = "http://" + url;
                }
            }
            Log.d("[+] url : ",url);
        }
    }
}
```

```

        webView.loadUrl(url);
    }
    catch (Exception e) {
        Log.d("[-] error : ", "error !", e);
    }
}
}

```

PENJELASAN

```

public class MainActivity extends AppCompatActivity {
    final Activity activity = this;

```

Di sini kita buat kelas baru dengan nama MainActivity yang menurunkan karakteristik dari kelas AppCompatActivity, karakteristik di sini bisa berupa method, variabel.

Penggunaan keyword “final” menyebabkan variabel bertipe objek Activity tidak bisa diubah lagi.

Penggunaan keyword “this” digunakan untuk mengisi variabel objek Activity activity dengan instance dari MainActivity

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    try {
        super.onCreate(savedInstanceState);

```

Selanjutnya kita override metode onCreate dari kelas AppCompatActivity. Penggunaan keyword “super” digunakan agar rutin rutin di dalam tubuh kelas onCreate pada kelas AppCompatActivity dieksekusi terlebih dahulu barulah rutin yang kita tambahkan di bawahnya akan dieksekusi.

```

WebView webView = (WebView) findViewById(R.id.webview);

```

Pada metode onCreate yang kita override, kita buat objek baru WebView yang kita isi dengan referensi webview di layout dengan id webview.

```

webView.getSettings().setJavaScriptEnabled(true);
webView.getSettings().setAllowFileAccessFromFileURLs(true);
webView.getSettings().setAllowUniversalAccessFromFileURLs(true);
webView.setWebChromeClient(new WebChromeClient() {

```

Pada rutin di atas, fitur java script pada webview kita aktifkan dan kita izinkan akses.

Baris baris selanjutnya :

```

webView.getSettings().setAllowFileAccessFromFileURLs(true);
webView.getSettings().setAllowUniversalAccessFromFileURLs(true);

```

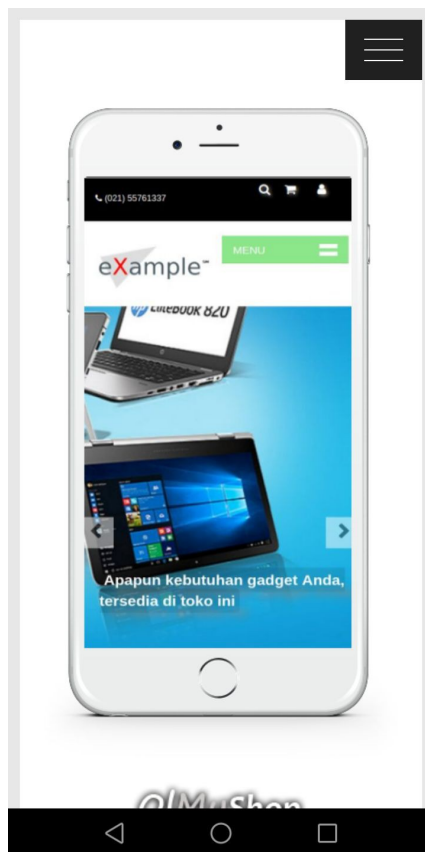
adalah opsional, untuk alasan keamanan baris ini bisa juga dihapus. Baris ini untuk mengizinkan webview mengakses local file pada device android.

Pada rutin selanjutnya, kita siapkan handler `WebViewClient`, `WebViewClient` merupakan kelas yang disediakan android untuk menampilkan halaman web, di mana di dalam kelas `WebViewClient`, kita perlu melakukan override dari default behaviour webview sebelumnya saat mengakses halaman web, pada objek sebelumnya saat mengakses halaman web akan menampilkan pilihan "open with", pada contoh ini kita override pada metode `shouldOverrideUrlLoading` agar web langsung di load di webview :

```
webView.setWebViewClient(new WebViewClient() {  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, String url)  
    {  
        view.loadUrl(url);  
        return true;  
    }  
});
```

Uji Coba Aplikasi

Aplikasi akan meload halaman web olmyshop.com dalam ukuran 1 layar penuh

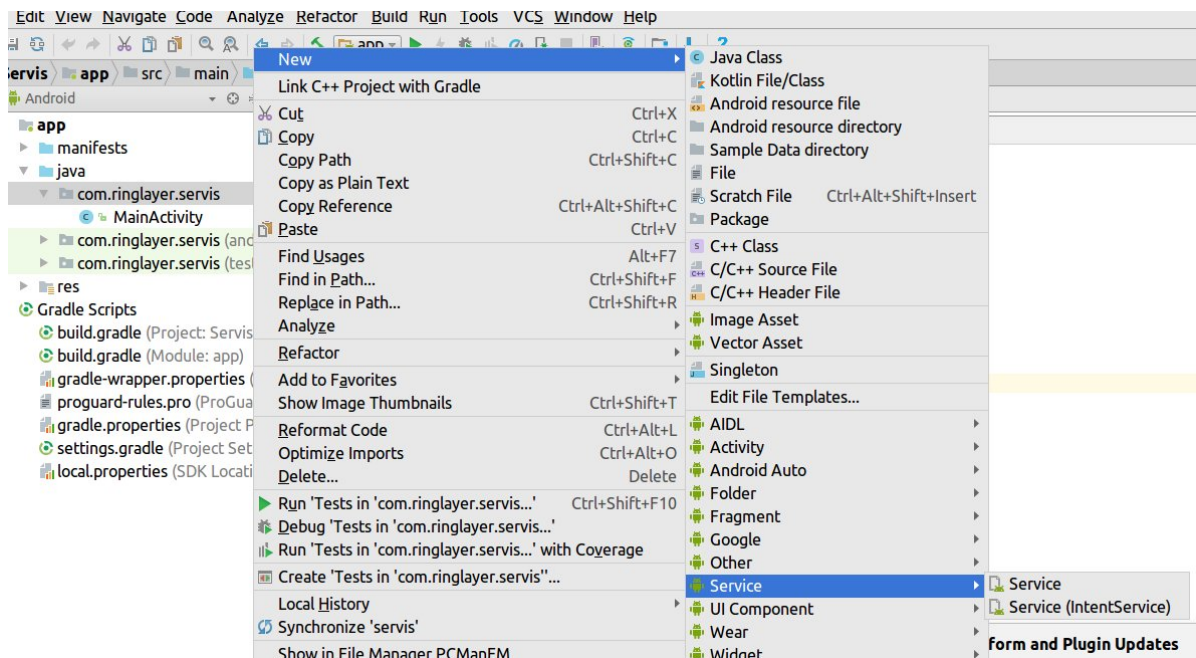


K. Servis di Android

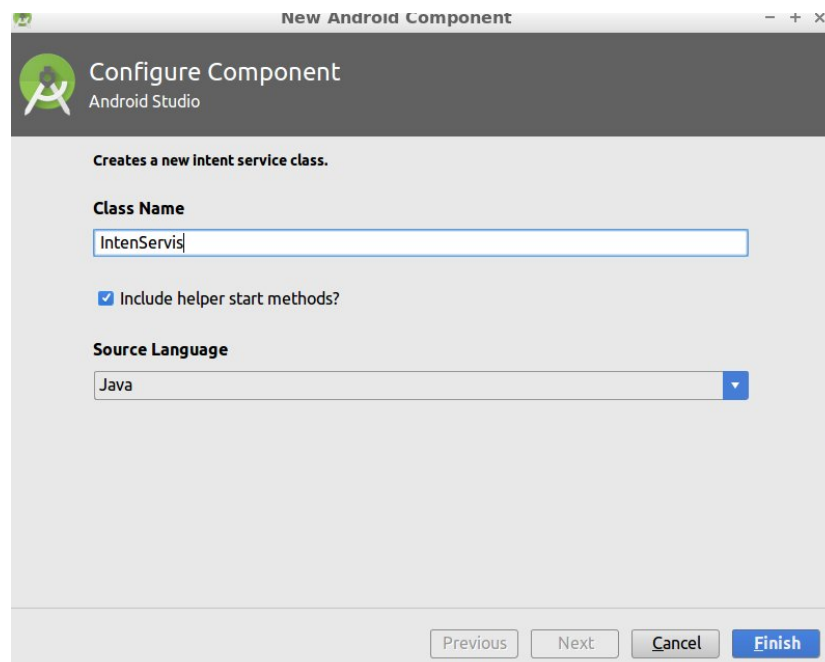
Servis adalah komponen aplikasi yang digunakan untuk menjalankan operasi di belakang layar dalam waktu lama, servis tidak memerlukan user interface.

Pada contoh kali ini kita akan membuat servis di android, yang kita akan buat adalah sebuah Intent Service. Buat proyek baru di android studio dengan nama Servis.

Untuk membuat servis, Klik kanan pada proyek anda di jendela struktur aplikasi lalu pilih new , pilih Service, pilih Intent Service.



Beri nama IntenServis



Bagian XML

Setelah IntenServis anda tambahkan, anda perlu menambahkan deklarasi IntenServis pada AndroidManifest.xml

Tambahkan baris ini pada bagian Application:

```
<service android:name=".IntenServis" />
```



Selain itu kita perlu menambahkan 2 permisi di AndroidManifest yaitu untuk membaca dan menulis file system. Tambahkan 2 baris ini ke AndroidManifest:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Berikut ini isi AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.servis">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <service android:name=".IntenServis" />
```

```

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

```

Bagian Source Code

Sama seperti bagian sebelumnya, pada aplikasi ini kita membutuhkan request permisi dari user android, karena servis akan bekerja di belakang layar untuk membuat direktori.

Agar Intent Servis IntenServis yang telah kita buat sebelumnya bisa berjalan otomatis begitu aplikasi dijalankan, maka kita perlu menambahkan source code untuk trigger agar intent service kita bisa berjalan.

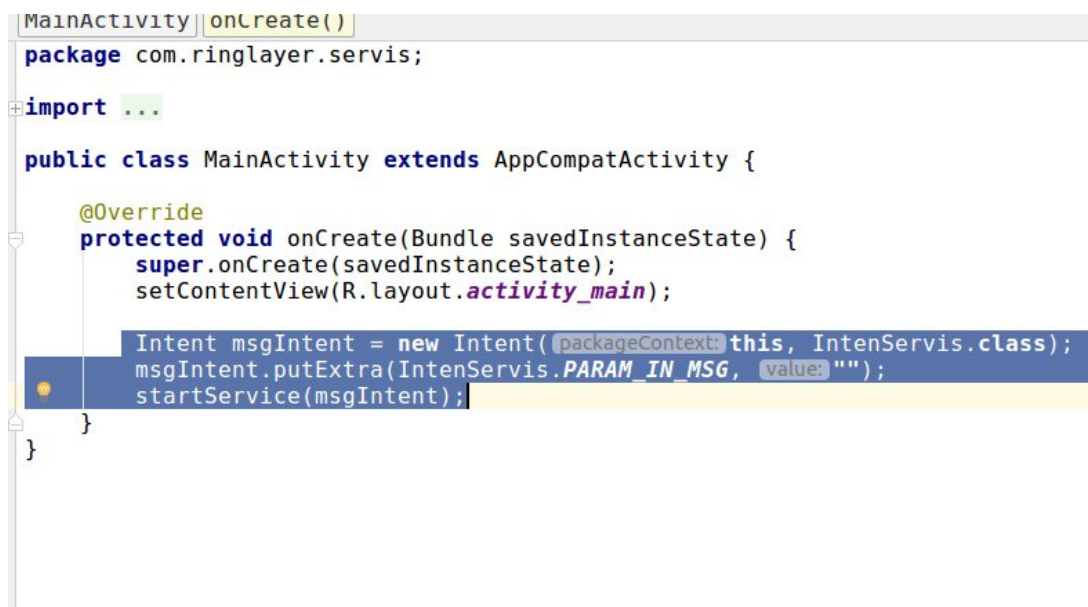
Kita bisa menaruh source untuk trigger intent service pada bagian onCreate pada activity launcher MainActivity. Berikut ini adalah source code yang akan mentrigger jalanya intent service yang telah kita buat :

```

Intent msgIntent = new Intent(this, IntenServis.class);
msgIntent.putExtra(IntenServis.PARAM_IN_MSG, "");
startService(msgIntent);

```

PARAM_IN_MSG merupakan parameter pesan input (opsional) yang bisa kita tambahkan pada intent service.



```

MainActivity | onCreate()
package com.ringlayer.servis;

import ...

public class MainActivity extends AppCompatActivity {

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

        Intent msgIntent = new Intent(packageContext: this, IntenServis.class);
        msgIntent.putExtra(IntenServis.PARAM_IN_MSG, value: "");
        startService(msgIntent);
    }
}

```


Berikut ini source code MainActivity

```
package com.ringlayer.servis;

import android.Manifest;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
        util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
        Intent msgIntent = new Intent(this, IntenServis.class);
        msgIntent.putExtra(IntenServis.PARAM_IN_MSG, "");
        startService(msgIntent);
    }
}
```

PENJELASAN

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

kelas MainActivity diciptakan, di mana kelas ini merupakan turunan yang mewariskan karakteristik dari kelas parent AppCompatActivity

```
public Util util = new Util(MainActivity.this);
```

Pada rutin ini objek Util dengan nama util diinisialisasi dari objek baru dari kelas Util.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

Selanjutnya kita melakukan override metode onCreate. Di sini kita menggunakan keyword super sehingga callback onCreate asli dari parent class akan dieksekusi terlebih dahulu, barulah rutin tambahan yang kita tambahkan di bawahnya.

Selanjutnya layout activity_main.xml akan dirender :

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

Langkah selanjutnya kita melakukan request permisi baca tulis file system ke user :

```
util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);
util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
```

Request permisi ini dibutuhkan pada versi android marshmallow ke atas.

Tahap terakhir adalah menjalankan Servis


```
Intent msgIntent = new Intent(this, IntenServis.class);
msgIntent.putExtra(IntenServis.PARAM_IN_MSG, "");
startService(msgIntent);
```

rutin putExtra di atas bisa digunakan untuk pengiriman data ke Servis.

Berikut ini source code lengkap IntenServis.java :

```
package com.ringlayer.servis;

import android.app.IntentService;
import android.content.Intent;
import android.content.Context;
import android.util.Log;
import java.io.File;
public class IntenServis extends IntentService {
    public static final String PARAM_IN_MSG = "";

    public IntenServis() {super("IntenServis");}
    @Override
    protected void onHandleIntent(Intent intent) {
        try {
            Thread.sleep(5000);
            Log.i("info", "[+] intent servis mulai dijalankan");
            buat_dir("intenservis");
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    public int buat_dir(String Dirname) {
        int retme = 0;
        try {
            boolean exists = false;
            exists = CheckIfDirectoryExists(Dirname);
            if (exists == false) {
                new File(Dirname).mkdirs();
            }
        }
        catch (Exception e) {
            Log.e("error",e.toString());
        }
        return retme;
    }
    public boolean CheckIfDirectoryExists(String path) {
        boolean exists = false;
        try {
            File dir = new File(path);
            exists = dir.exists();
            return exists;
        }
        catch (Exception e) {
            Log.e("error",e.toString());
            return false;
        }
    }
}
```

PENJELASAN

```
public class IntenServis extends IntentService {
```

digunakan untuk membuat kelas baru yang merupakan turunan dari parent class IntentService

```
@Override  
protected void onHandleIntent(Intent intent) {
```

Bagian metode callback onHandleIntent kita override, di mana bodnya kita isi dengan rutin berikut ini :

```
Thread.sleep(5000);  
    Log.i("info", "[+] intent servis mulai dijalankan");  
    buat_dir("intenservis");
```

Pertama tama servis akan idle selama 5 detik (5000 microseconds = 5 seconds). Pemanggilan metode i, pada Log hanyalah untuk kepentingan verbose (opsional). Selanjutnya kita buat direktori baru dengan nama intenservis dengan metode yang kita buat bernama buat_dir.

Metode buat_dir pada dasarnya akan mengecek keberadaan direktori

```
exists = CheckIfDirectoryExists(Dirname);  
    if (exists == false) {  
        new File(Dirname).mkdirs();  
    }
```

Jika direktori tidak ditemukan maka direktori baru akan dibuat.

Metode CheckIfDirectoryExists, intinya akan mengecek keberadaan direktori yang diinisialisasi pada variabel path bertipe String.

```
public boolean CheckIfDirectoryExists(String path) {  
    boolean exists = false;  
    try {  
        File dir = new File(path);  
        exists = dir.exists();  
        return exists;  
    }  
    catch (Exception e) {  
        Log.e("error",e.toString());  
        return false;  
    }  
}
```

Pengecekan keberadaan direktori berada pada rutin ini :

```
File dir = new File(path);  
exists = dir.exists();
```

Pertama tama objek berupa File dengan nama dir akan diinisialisasi dengan objek baru dari kelas File.

Selanjutnya variabel boolean exists akan digunakan menyimpan nilai kembali dari hasil pemanggilan metode exists pada kelas File.

Jika file ditemukan maka boolean exists bernilai true, jika file tidak ditemukan, maka boolean bernilai false.

L. Broadcast Receiver di Android

Broadcast Receiver merupakan salah satu komponen Android yang bertugas dalam penerima pesan dari sistem atau sebuah aplikasi, bentuk pesan yang dikirim ini bisa berupa event atau invent.

Untuk menguji Broadcast Receiver, buat projek baru dengan nama BroadRec. Cara kerja aplikasi kali ini adalah, pertama tama saat MainActivity dijalankan, activity ini akan meregister satu broadcast receiver untuk menangkap pesan, selanjutnya activity ini akan menjalankan servis di background di mana servis di background ini setelah selesai dieksekusi akan melakukan pesan ke MainActivity yang akan ditangkap oleh broadcast receiver yang sebelumnya telah diregister.

Bagian XML

Pada bagian kali ini, yang perlu diedit hanyalah AndroidManifest.xml. Berikut ini isi file AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.broadrec">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <service android:name=".Servis" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Bagian Source Code

Sama seperti proyek sebelumnya, kali ini kita memerlukan kelas Util yang digunakan sebagai rutin untuk merequest permissi untuk baca tulis file system. Buat kelas baru dengan nama Util. Berikut ini isi file Util.java

```

package com.ringlayer.broadrec;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
/**
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        public void askForPermission(String permission, Integer requestCode) {
            if (ContextCompat.checkSelfPermission(mContext, permission) !=
                PackageManager.PERMISSION_GRANTED) {
                if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                    permission)) {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                } else {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                }
            }
        }
    }
}

```

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```

if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {

```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```

ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);

```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

Selanjutnya kita memerlukan satu kelas lagi sebagai servis. Buat kelas baru dengan nama Servis. Berikut ini isi file Servis.java :

```

package com.ringlayer.broadrec;
import android.app.Service;
import android.content.Intent;

```

```

import android.os.IBinder;
import android.util.Log;
import java.io.File;
/**
 * Created by ringlayer on 15/03/19.
 */
public class Servis extends Service {
    public static final String PARAM_IN_MSG = "";
    protected void onHandleIntent(Intent intent) {
        try {
            String pesan = "";
            Log.e("[ - ] ohi", PARAM_IN_MSG);
            buat_dir("intenservis");
            boolean exists = CheckIfDirectoryExists("/sdcard/intenservis");
            if (exists) {
                pesan = "berhasil";
            }
            else {
                pesan = "gagal";
            }
            publishResults(pesan);
        }
        catch (Exception e) {
            Log.e("onHandleIntent ServisCekSesi", "Exception: " +
Log.getStackTraceString(e));
        }
    }
    public int buat_dir(String Dirname) {
        int retme = 0;
        try {
            boolean exists = false;
            exists = CheckIfDirectoryExists(Dirname);
            if (exists == false) {
                new File(Dirname).mkdirs();
            }
        }
        catch (Exception e) {
            Log.e("error",e.toString());
        }
        return retme;
    }
    public boolean CheckIfDirectoryExists(String path) {
        boolean exists = false;
        try {
            File dir = new File(path);
            exists = dir.exists();
            return exists;
        }
        catch (Exception e) {
            Log.e("error",e.toString());
            return false;
        }
    }
    private void publishResults(String pesan) {
        try {
            Log.e("[ + ] publishResults", pesan);
            Intent intent = new Intent("1");
            intent.putExtra("BROADCAST", pesan);
            sendBroadcast(intent);
        }
        catch (Exception e) {
            Log.e("publishResults", e.toString());
        }
    }
}

```

PENJELASAN

Perhatikan baris ini :

```
public class Servis extends Service
```

pada rutin di atas kita membuat class baru dengan nama Servis yang merupakan inherit / turunan dari parent class Service.

```
protected void onHandleIntent(Intent intent) {
    try {
        String pesan = "";
        Log.e("[-] ohi", PARAM_IN_MSG);
        buat_dir("intenservis");
        boolean exists = CheckIfDirectoryExists("/sdcard/intenservis");
        if (exists) {
            pesan = "berhasil";
        }
        else {
            pesan = "gagal";
        }
        publishResults(pesan);
    }
}
```

pada bagian callback method onHandleIntent, kita membuat direktori baru dengan nama intenservis. Selanjutnya variabel bertipe boolean (bisa berisi nilai logika true atau false) diinisialisasi dari hasil nilai kembali metode CheckIfDirectoryExists.

Metode CheckIfDirectoryExists, intinya akan mengecek keberadaan direktori yang diinisialisasi pada variabel path bertipe String.

```
public boolean CheckIfDirectoryExists(String path) {
    boolean exists = false;
    try {
        File dir = new File(path);
        exists = dir.exists();
        return exists;
    }
    catch (Exception e) {
        Log.e("error", e.toString());
        return false;
    }
}
```

Pengecekan keberadaan direktori berada pada rutin ini :

```
File dir = new File(path);
exists = dir.exists();
```

Pertama tama objek berupa File dengan nama dir akan diinisialisasi dengan objek baru dari kelas File.

Selanjutnya variabel boolean exists akan digunakan menyimpan nilai kembali dari hasil pemanggilan metode exists pada kelas File.

Jika file ditemukan maka boolean exists bernilai true, jika file tidak ditemukan, maka boolean bernilai false.

Metode selanjutnya yang berada paling bawah adalah metode `publishResults`, metode ini akan melakukan broadcast hasil eksekusi servis.

Pada `MainActivity` kita mendaftarkan satu broadcast receiver untuk menangkap pesan dari Servis yang kita jalankan di background. Berikut ini source code `MainActivity.java` :

```
package com.ringlayer.broadrec;
import android.Manifest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
public class MainActivity extends AppCompatActivity {
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE,
1);
            util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);
        }
        Intent intent = new Intent(MainActivity.this, Servis.class);
        startService(intent);
        stopService(intent);
    }
    private BroadcastReceiver br = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String act = "";
            final Context konteks = context;
            try {
                Bundle bundle = intent.getExtras();
                if (bundle != null) {
                    act = bundle.getString("BROADCAST").trim();
                    Log.e("[+] pgonReceive", act);
                }
            }
            catch (Exception e) {
                Log.e("onReceive", e.toString());
            }
        }
    };
    @Override
    protected void onResume() {
        super.onResume();
        registerReceiver(br, new IntentFilter("1"));
    }
    @Override
    protected void onPause() {
        super.onPause();
        unregisterReceiver(br);
    }
}
```


PENJELASAN

Pada contoh di atas, saat metode onCreate, jika versi android adalah marshmallow ke atas maka aplikasi akan melakukan request permisi untuk baca tulis

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    util.askForPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE, 1);  
    util.askForPermission(Manifest.permission.READ_EXTERNAL_STORAGE, 2);  
}
```

Activity MainActivity ini akan menjalankan service dengan nama Servis.

```
Intent intent = new Intent(MainActivity.this, Servis.class);  
startService(intent);  
stopService(intent);
```

Selanjutnya suatu objek BroadcastReceiver akan diinisialisasi di mana metode callback onReceive akan di override

```
private BroadcastReceiver br = new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        String act = "";  
        final Context konteks = context;  
        try {  
            Bundle bundle = intent.getExtras();  
            if (bundle != null) {  
                act = bundle.getString("BROADCAST").trim();  
                Log.e("[+] pgonReceive", act);  
            }  
        }  
        catch (Exception e) {  
            Log.e("onReceive", e.toString());  
        }  
    }  
};
```

Pada metode onReceiver ini akan diambil pesan dari putExtras melalui metode getExtras yang akan disimpan dalam objek Bundle :

```
Bundle bundle = intent.getExtras();
```

Jika didapati bundle diinisialisasi dari pesan hasil getExtras maka variable berjenis String dengan nama variable act akan diisi dengan hasil pesan broadcast dari servis.

```
if (bundle != null) {  
    act = bundle.getString("BROADCAST").trim();
```

Perhatikan baris ini :

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

```
        registerReceiver(br, new IntentFilter("1"));  
    }
```

pada baris

```
super.onResume();
```

karena kita akan melakukan override pada onResume, kita menggunakan super. Dengan rutin ini maka onResume yang merupakan metode asli dari kelas parent class akan dieksekusi terlebih dahulu barulah di bawahnya akan dieksekusi rutin rutin yang kita tambahkan untuk onResume.

Pada metode callback onResume, kita menambahkan rutin tambahan untuk melakukan register broadcast receiver br yang tadi kita inisialisasi. Selain pada metode callback onResume, broadcast receiver bisa juga kita register pada metode onStart.

M. Bluetooth di Android

Bluetooth adalah teknologi wireless yang digunakan untuk tukar menukar data antar perangkat pada jarak yang relatif dekat. Bluetooth merupakan frekuensi radio dengan band 2.400 hingga 2.485 GHz. Pada kesempatan kali ini kita akan mencoba membuat aplikasi android dengan fitur komunikasi melalui bluetooth.

Bluetooth API pada Android

Android menyediakan API (application programming interface) untuk konektivitas bluetooth. Untuk melihat detail daftar kelas dan interface yang ada pada API untuk bluetooth, bisa dicek di <https://developer.android.com/reference/android/bluetooth/package-summary>

Pada contoh aplikasi kali ini, kita akan menggunakan Bluetooth Adapter Class. Source code java untuk kelas ini bisa dilihat di <https://android.googlesource.com/platform/frameworks/base/+/-/master/core/java/android/bluetooth/BluetoothAdapter.java>

Berikut ini adalah beberapa konstan yang terdapat pada Bluetooth Adapter Class:

- ACTION_REQUEST_ENABLE

Menunjukkan activity dari sistem yang menunjukkan user untuk menyalakan bluetooth.

- ACTION_REQUEST_DISCOVERABLE

Digunakan untuk menyalakan bluetooth

- ACTION_DISCOVERY_STARTED

Merupakan broadcast action yang menunjukkan adapter bluetooth sudah memulai proses pencarian device bluetooth

- ACTION_DISCOVERY_FINISHED

Merupakan broadcast action yang menunjukkan adapter bluetooth selesai melakukan proses pencarian device bluetooth.

- ACTION_FOUND

Digunakan untuk menerima informasi masing masing device bluetooth yang ditemukan

Untuk daftar lengkapnya bisa dicek di

<https://developer.android.com/reference/android/bluetooth/BluetoothAdapter>

Pada contoh ini kita akan membuat dua macam aplikasi dengan konektivitas bluetooth, yang pertama adalah aplikasi dasar untuk konektivitas bluetooth dan aplikasi kedua untuk melakukan koneksi bluetooth ke printer dengan konektivitas bluetooth.

Aplikasi Pertama untuk Menguji Dasar Konektivitas dengan Bluetooth

Buat proyek android baru dengan nama BlueTooth.

Bagian XML

Pada proyek kali ini kita akan menggunakan 1 layout yaitu activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.ringlayer.bluetooth.MainActivity">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:id="@+id/background"
android:gravity="center">
<TextView
android:id="@+id/desc"
android:background="@color/colorAccent"
android:text="No Status"
android:textColor="#ffffff"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingRight="10dp"
android:paddingLeft="10dp"
/>
<Space
android:layout_width="1dp"
android:layout_height="30dp"/>
<Button
android:id="@+id/on"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:text="Turn On"
/>
<Space
android:layout_width="1dp"
android:layout_height="30dp"/>
<Button
android:id="@+id/discover"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Discoverable"
/>
<Space
android:layout_width="1dp"
android:layout_height="30dp"/>
<Button
android:id="@+id/off"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Turn Off"
    />
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

Pertama tama tambahkan permisi bluetooth pada bagian AndroidManifest.xml

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

```

Berikut ini isi lengkap AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ringlayer.bluetooth">
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Bagian Source Code

Di proyek ini kita membutuhkan kelas Util sama seperti proyek proyek sebelumnya. Tambahkan kelas baru dengan nama kelas Util.

Berikut ini isi Util.java :

```

package com.ringlayer.bluetooth;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
/**

```

```

* Created by ringlayer on 24/11/18.
*/
public class Util {
    Activity mContext;

    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);}

        public void askForPermission(String permission, Integer requestCode) {
            if (ContextCompat.checkSelfPermission(mContext, permission) !=
                PackageManager.PERMISSION_GRANTED) {

                if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
                    permission)) {

                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                } else {
                    ActivityCompat.requestPermissions(mContext, new String[]{permission},
                        requestCode);
                }
            }
        }
    }
}

```

PENJELASAN

Kelas Util di atas digunakan untuk melakukan request permissi dari user berupa pop up. Pertama tama kita cek terlebih dahulu apakah permissi sudah diizinkan :

```

if (ContextCompat.checkSelfPermission(mContext, permission) !=
    PackageManager.PERMISSION_GRANTED) {

```

Pada rutin di atas, jika permissi bukanlah **PERMISSION_GRANTED** , maka source code di bawahnya akan dieksekusi.

Source code di bawahnya berupa request permissi ke user :

```

ActivityCompat.requestPermissions(mContext, new String[]{permission}, requestCode);

```

Pada baris di atas metode requestPermissions dipanggil untuk menampilkan popup merequest suatu permissi.

MainActivity.java :

```

package com.ringlayer.bluetooth;
import android.Manifest;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;

```

```

import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Set;
import static android.Manifest.permission.CAMERA;
public class MainActivity extends AppCompatActivity {
    final BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();
    TextView desc;
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                util.askForPermission(Manifest.permission.BLUETOOTH, 1);
                util.askForPermission(Manifest.permission.BLUETOOTH_ADMIN, 2);
            }
            desc = (TextView) findViewById(R.id.desc);
            final Button on = (Button) findViewById(R.id.on);
            final Button discover = (Button) findViewById(R.id.discover);
            final Button off = (Button) findViewById(R.id.off);
            if (bluetooth == null) {
                Toast.makeText(this, "Bluetooth not supported on this device",
Toast.LENGTH_LONG).show();
            }
            on.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    On();
                }
            });
            off.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Off();
                }
            });
            discover.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    Discover();
                }
            });
        } catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    void On() {
        try {
            desc.setText("Bluetooth on");
            Log.e("on", "Turn on");
            Intent turnOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(turnOn, 0);
            Toast.makeText(getApplicationContext(), "Bluetooth
On", Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            Toast.makeText(getApplicationContext(), "Failed to turn on
bluetooth", Toast.LENGTH_LONG).show();
            Log.e("error", e.toString());
        }
    }
    void Off() {

```

```

        try {
            desc.setText("Bluetooth off");
            bluetooth.disable();
            Toast.makeText(getApplicationContext(), "Bluetooth
Off", Toast.LENGTH_LONG).show();
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    void Discover() {
        try {
            if (!bluetooth.isDiscovering()) {
                Toast.makeText(getApplicationContext(), "MAKING YOUR DEVICE
DISCOVERABLE",
                    Toast.LENGTH_LONG);
                Intent EnableDiscoverable = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
                startActivityForResult(EnableDiscoverable, 0);
                desc.setText("This device is discoverable");
            }
        }
        catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
}

```

PENJELASAN

Pertama tama kita siapkan terlebih dahulu objek berupa BluetoothAdapter yang nantinya akan kita gunakan untuk koneksi bluetooth.

```
final BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();
```

Selanjutnya :

```

final Button on = (Button) findViewById(R.id.on);
final Button discover = (Button) findViewById(R.id.discover);
final Button off = (Button) findViewById(R.id.off);

```

Kita ambil referensi referensi ke tombol tombol di layout dengan id on, discover dan off.

Langkah selanjutnya adalah memasang callback ketika user melakukan klik tombol, di mana masing masing tombol ketika diklik akan menjalankan “metode” yang telah dipersiapkan untuk menangani perangkat bluetooth di android.

```

on.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        On();
    }
});
off.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Off();
    }
});
discover.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Discover();
    }
});

```



```
    }  
});
```

Pada metode On, terdapat rutin untuk mengaktifkan bluetooth dengan Intent.

```
Intent turnOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
startActivityForResult(turnOn, 0);
```

Pada metode Discover, terdapat rutin untuk membuat android kita menjadi discoverable (dapat terlihat) oleh perangkat bluetooth lainnya di sekitar

```
Intent EnableDiscoverable = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);  
startActivityForResult(EnableDiscoverable, 0);
```

Pada metode Off, terdapat rutin untuk menonaktifkan bluetooth

```
bluetooth.disable();
```

Aplikasi Kedua : Koneksi ke Printer Bluetooth

Untuk aplikasi kedua kita akan membuat aplikasi android yang terkoneksi ke printer bluetooth.

Buat proyek baru dengan nama PrinterBluetooth

BAGIAN XML

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.ringlayer.printerbluetooth">  
    <uses-permission android:name="android.permission.BLUETOOTH" />  
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportsRtl="true"  
        android:theme="@style/AppTheme">  
        <activity android:name=".MainActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.ringlayer.printerbluetooth.MainActivity">
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:layout_margin="10dp">
<TextView
android:id="@+id/label"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Type here:" />
<EditText
android:id="@+id/entry"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_below="@id/label" />
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_below="@id/entry">
<Button
android:id="@+id/open"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="10dip"
android:text="Open" />
<Button
android:id="@+id/send"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Send" />
<Button
android:id="@+id/close"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Close" />
</LinearLayout>
</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

BAGIAN SOURCE CODE

Util.java :

```
package com.ringlayer.bluetooth;
```

```

import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.StrictMode;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
/**
 * Created by ringlayer on 24/11/18.
 */
public class Util {
    Activity mContext;
    public Util (Activity context)
    {
        mContext = context;
        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);}
    }
    public void askForPermission(String permission, Integer requestCode) {
        if (ContextCompat.checkSelfPermission(mContext, permission) !=
        PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(mContext,
            permission)) {
                ActivityCompat.requestPermissions(mContext, new String[]
            {permission}, requestCode);
            } else {
                ActivityCompat.requestPermissions(mContext, new String[]
            {permission}, requestCode);
            }
        }
    }
}

```

MainActivity.java :

```

package com.ringlayer.printerbluetooth;
import android.Manifest;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Build;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Set;
import java.util.UUID;
/*
 * source code was modified from internet sample
 * of bluetooth printer

```

```

*
* */
public class MainActivity extends AppCompatActivity {
    BluetoothAdapter mBluetoothAdapter;
    BluetoothSocket mmSocket;
    BluetoothDevice mmDevice;
    OutputStream mmOutputStream;
    InputStream mmInputStream;
    Thread workerThread;
    byte[] readBuffer;
    int readBufferPosition;
    volatile boolean stopWorker;
    TextView myLabel;
    EditText myTextbox;
    public Util util = new Util(MainActivity.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                util.askForPermission(Manifest.permission.BLUETOOTH, 1);
                util.askForPermission(Manifest.permission.BLUETOOTH_ADMIN, 2);
            }
            Button openButton = (Button) findViewById(R.id.open);
            Button sendButton = (Button) findViewById(R.id.send);
            Button closeButton = (Button) findViewById(R.id.close);
            myLabel = (TextView) findViewById(R.id.label);
            myTextbox = (EditText) findViewById(R.id.entry);
            openButton.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    findBT();
                    openBT();
                }
            });
            sendButton.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    sendData();
                }
            });
            closeButton.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    closeBT();
                }
            });
        } catch (Exception e) {
            Log.e("error", e.toString());
        }
    }
    void findBT() {
        try {
            mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
            if (mBluetoothAdapter == null) {
                myLabel.setText("No bluetooth adapter available");
            }
            if (!mBluetoothAdapter.isEnabled()) {
                Intent enableBluetooth = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBluetooth, 0);
            }
        }
    }
}

```

```

Set<BluetoothDevice> pairedDevices =
mBluetoothAdapter.getBondedDevices();
    if(pairedDevices.size() > 0) {
        for (BluetoothDevice device : pairedDevices) {
            /* GANTI RPP300 DENGAN NAMA DEVICE PRINTER */
            if (device.getName().equals("RPP300")) {
                mmDevice = device;
                break;
            }
        }
        myLabel.setText("Bluetooth device found.");
    }
    catch(Exception e){
        e.printStackTrace();
    }
}

void openBT() {
    try {
        // Standard SerialPortService ID
        UUID uuid = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
        mmSocket = mmDevice.createRfcommSocketToServiceRecord(uuid);
        mmSocket.connect();
        mmOutputStream = mmSocket.getOutputStream();
        mmInputStream = mmSocket.getInputStream();
        beginListenForData();
        myLabel.setText("Bluetooth Opened");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

void beginListenForData() {
    try {
        final Handler handler = new Handler();
        // this is the ASCII code for a newline character
        final byte delimiter = 10;
        stopWorker = false;
        readBufferPosition = 0;
        readBuffer = new byte[1024];
        workerThread = new Thread(new Runnable() {
            public void run() {
                while (!Thread.currentThread().isInterrupted() && !
stopWorker) {
                    try {
                        int bytesAvailable = mmInputStream.available();
                        if (bytesAvailable > 0) {
                            byte[] packetBytes = new byte[bytesAvailable];
                            mmInputStream.read(packetBytes);
                            for (int i = 0; i < bytesAvailable; i++) {
                                byte b = packetBytes[i];
                                if (b == delimiter) {
                                    byte[] encodedBytes = new

byte[readBufferPosition];

                                    System.arraycopy(
                                        readBuffer, 0,
                                        encodedBytes, 0,
                                        encodedBytes.length
                                    );
                                    // specify US-ASCII encoding
                                    final String data = new

String(encodedBytes, "US-ASCII");
                                    readBufferPosition = 0;
                                }
                            }
                        }
                    }
                }
            }
        });
    }
}

```

```

bluetooth printer device

// tell the user data were sent to
handler.post(new Runnable() {
    public void run() {
        myLabel.setText(data);
    }
});
} else {
    readBuffer[readBufferPosition++] = b;
}
}
}
} catch (IOException ex) {
    stopWorker = true;
}
}
});
workerThread.start();
}
catch(Exception e) {
    Log.e("error", e.toString());
}
}
void sendData() throws IOException {
    try {
        String msg = myTextbox.getText().toString();
        msg += "\n";
        mmOutputStream.write(msg.getBytes());
        myLabel.setText("Data sent.");
    }
    catch(Exception e) {
        Log.e("error", e.toString());
    }
}
void closeBT() throws IOException {
    try {
        stopWorker = true;
        mmOutputStream.close();
        mmInputStream.close();
        mmSocket.close();
        myLabel.setText("Bluetooth Closed");
    }
    catch(Exception e) {
        Log.e("error", e.toString());
    }
}
}
}

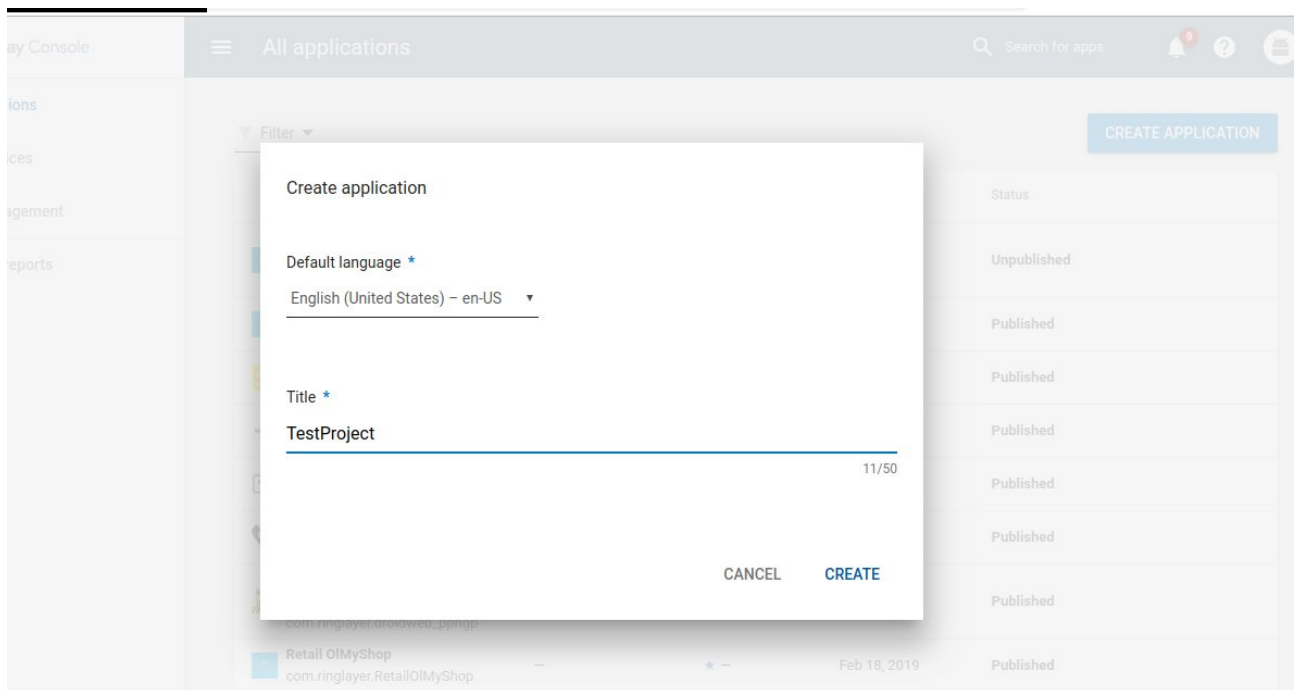
```

H. Mengupload ke Google Playstore

Pada contoh kali ini, kita akan melakukan publishing aplikasi ke google playstore.

Untuk mengaktifkan akun google play developer console. Anda memerlukan data kartu kredit.

Setelah akun Anda aktif, Untuk mempublish aplikasi baru klik tombol “CREATE APPLICATION”.



Selanjutnya isikan deskripsi aplikasi

ay Console

Store listing

TestProject Draft

ENGLISH (UNITED STATES) - en-US Manage translations

Fields marked with * need to be filled before publishing.

Product details

Title *
English (United States) - en-US TestProject 11/50

Short description *
English (United States) - en-US keterangan singkat 18/80

Full description *
English (United States) - en-US keterangan penuh

SAVE DRAFT

Selanjutnya isikan image untuk phone dan tablet.

ay Console

Store listing

TestProject Draft

ENGLISH (UNITED STATES) - en-US Manage translations

Add at least one phone screenshot here to help phone users see how your app will look on their device.

BROWSE FILES

Hi-res icon *
Default - English (United States) - en-US
512 x 512
32-bit PNG (with alpha)

Feature Graphic *
Default - English (United States) - en-US
1024 w x 500 h
JPG or 24-bit PNG (no alpha)

Promo Graphic
Default - English (United States) - en-US
180 w x 120 h
JPG or 24-bit PNG (no alpha)

+ Add high-res icon
Drop image here

+ Add feature graphic
Drop image here

+ Add promo graphic
Drop image here

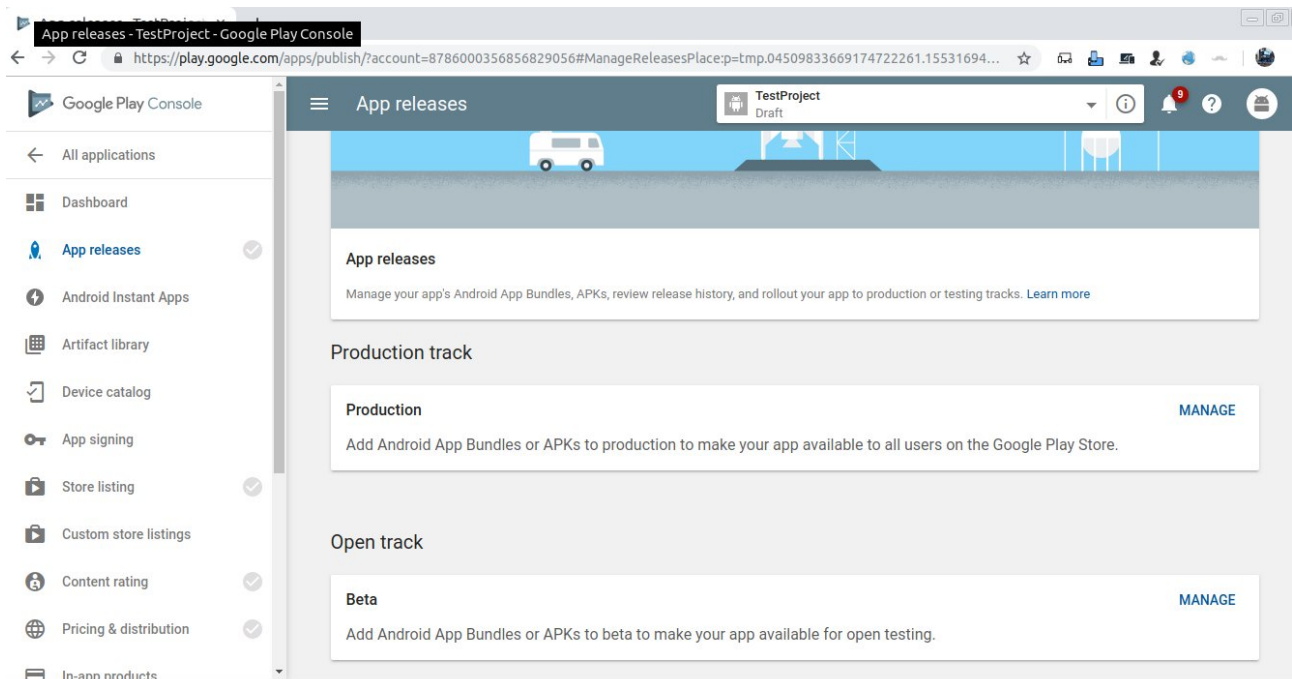
SAVE DRAFT

Semua ukuran image dalam pixel

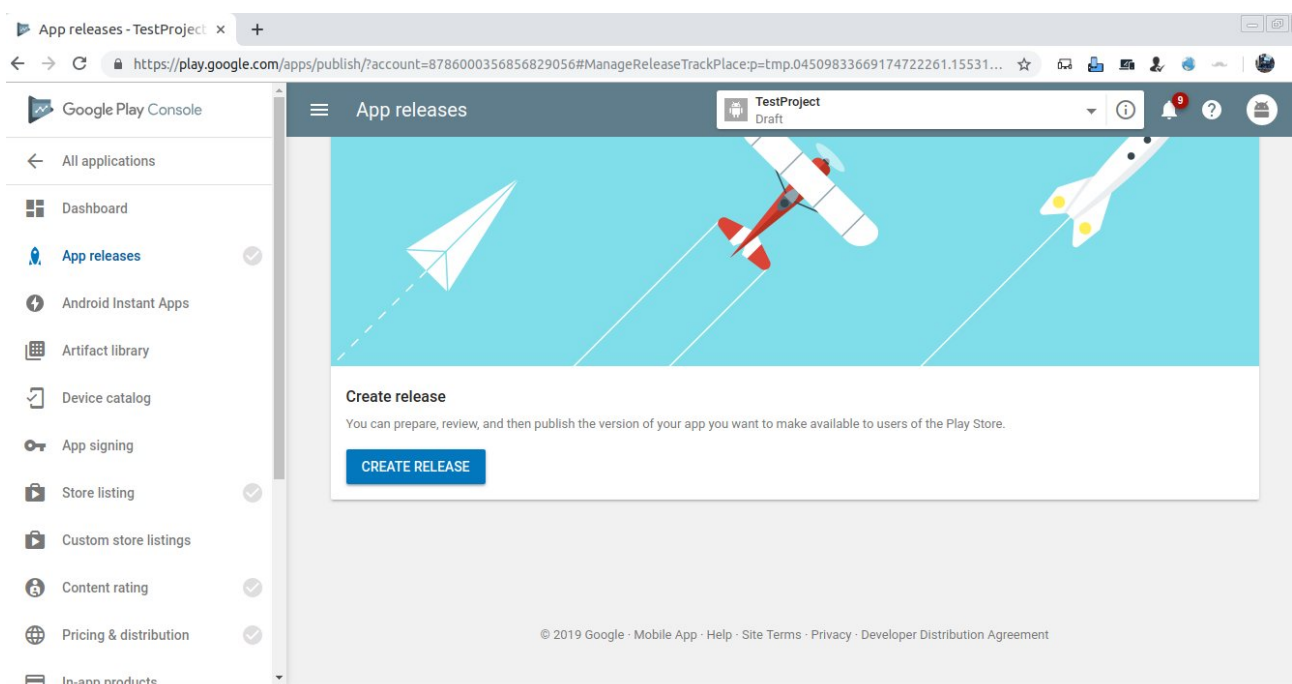
- Upload 2 screenshot aplikasi pada android
- Upload 2 screenshot aplikasi pada tablet
- High res icon gunakan ikon png berukuran 512 x 512
- Feature Graphic gunakan image png ukuran 1024 x 500
- Promo Graphic, gunakan image ukuran 180 x 120

- Selanjutnya lengkapi data isian di bawahnya.
- Setelah selesai, klik tombol “Save Draft”

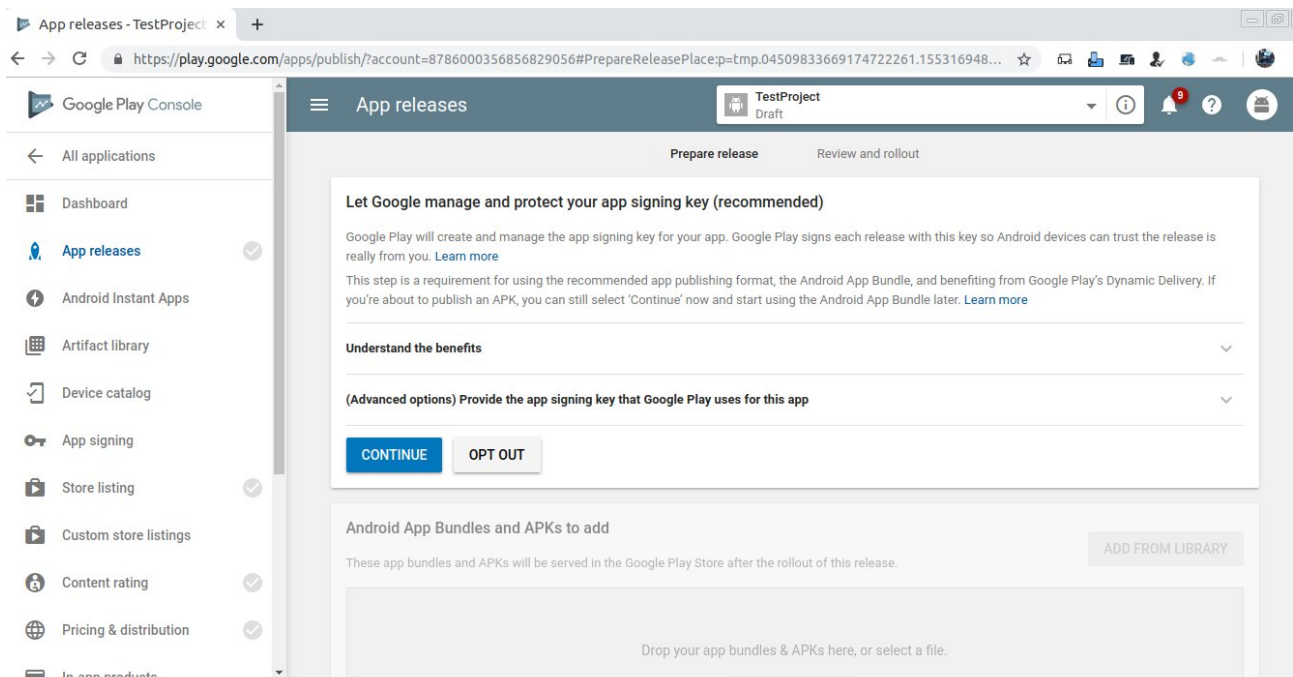
Selanjutnya klik menu “App Releases”, pada bagian production klik “Manage”



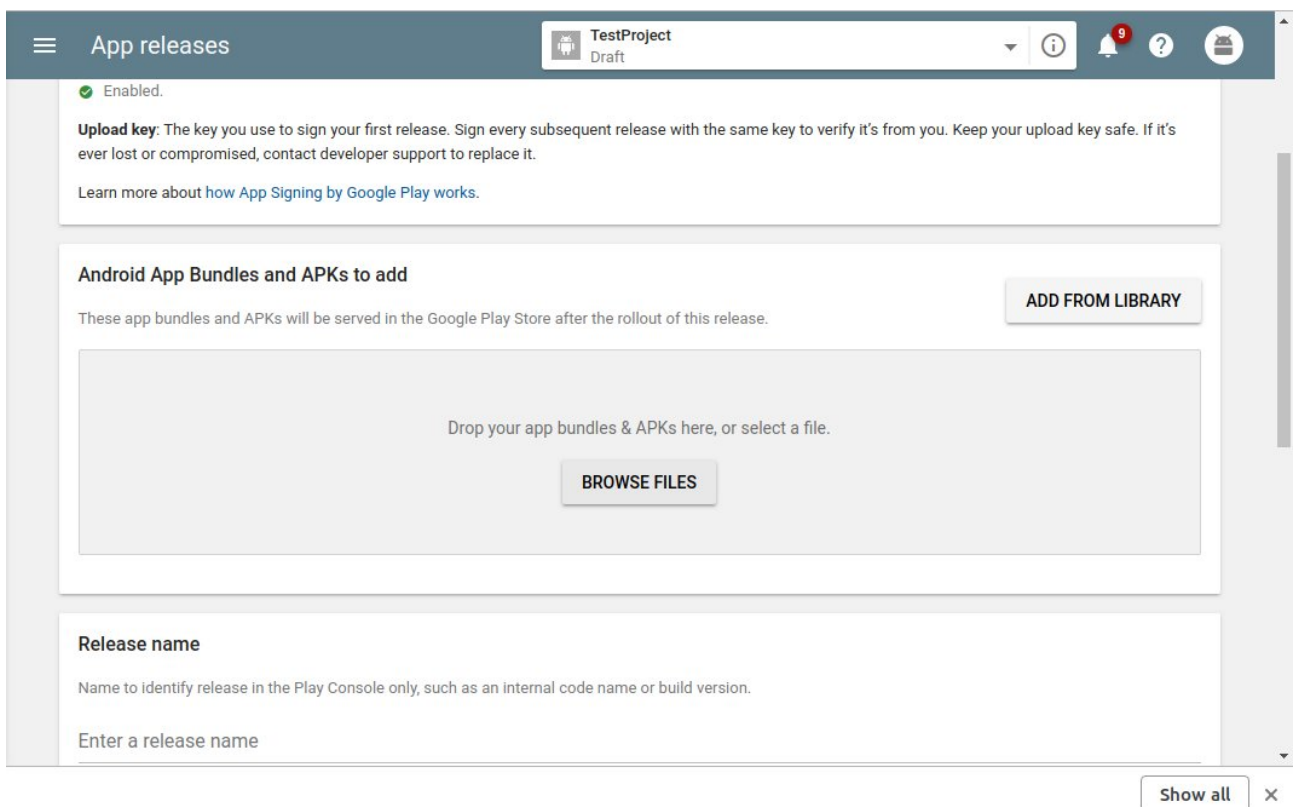
Selanjutnya klik tombol “Create Release”



Selanjutnya klik tombol “Continue”



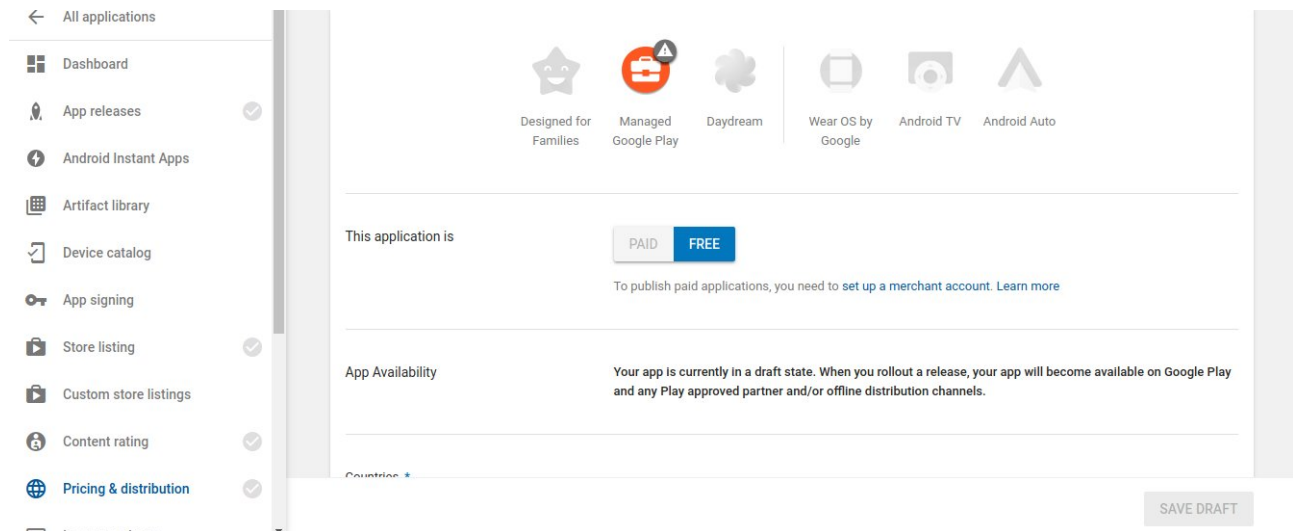
Selanjutnya upload apk dari komputer



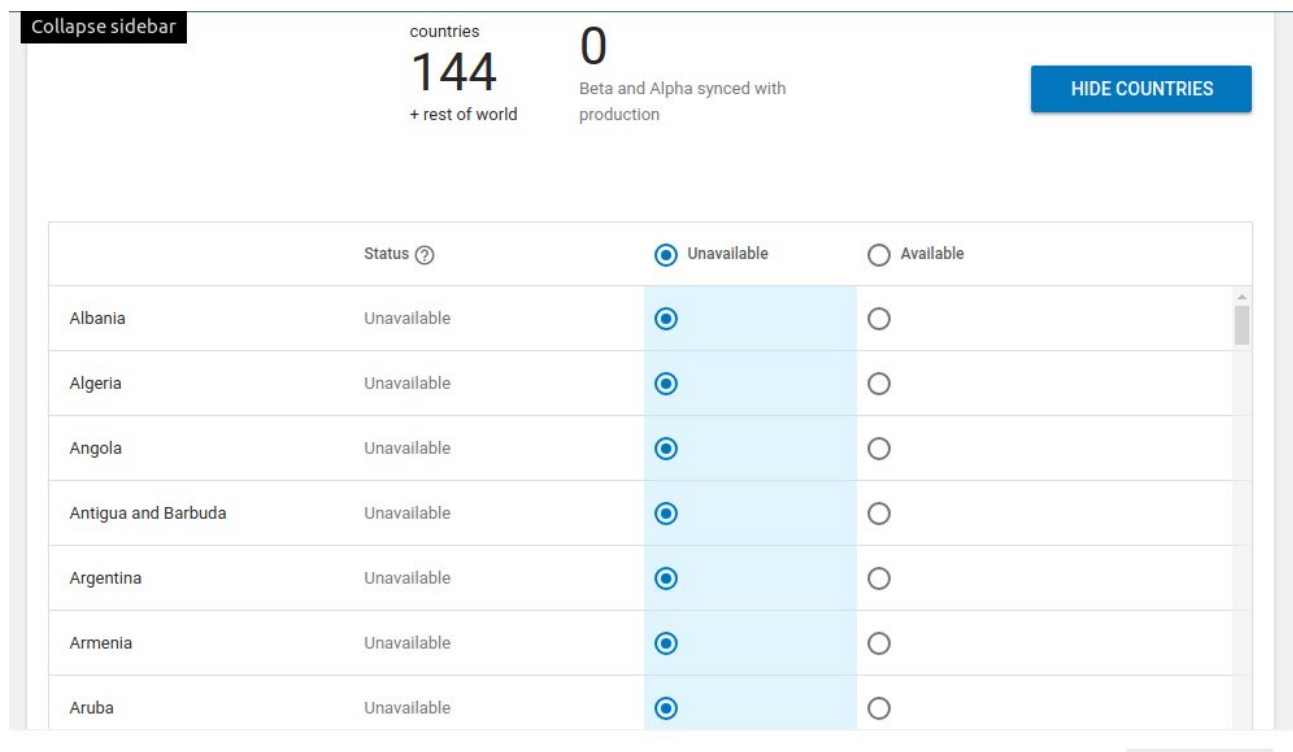
- Isikan release name
- Isikan What's new in this release

Sebelum mengklik tombol review, kita perlu mengisi content rating. Klik content rating setelah mengupload apk

Setelah mengapply rating, Klik pricing & distribution



Dari 144 negara, pilih negara yang available



Setelah klik tombol save draft. Kembali lagi ke menu “App Releases”, lalu klik “Edit Release”

Seret page ke bawah. Setelah Anda pastikan semua data terisi dengan benar. Klik tombol review

DISCARD

SAVED

REVIEW

© 2019 Google · Mobile App · Help · Site Terms · Privacy · Developer Distribution Agreement