MODUL : CAMERA API

**A. Tujuan**

Pada modul ini merupakan lanjutan dari modul camera 1, perbedaanya pada modul ini akan menyimpan hasilnya pengambilan gambar pada external storage dan melakukan proses sederhana pada image yang sudah disimpan.

Tujuan modul ini, yaitu :

* Mengetahui cara untuk mengakses Camera pada perangkat android
* Dapat mengimplementasikan Camera API untuk membuat aplikasi android
* Menyimpan hasilnya pada *external storage*
* Mengetahui cara untuk melakukan prosesan image/citra

**B. DasarTeori**

**1. Camera API**

Android mendukung penggunaan kamera dan fitur kamera yang tersedia pada perangkat mobile/*handheld*, sehingga kita dapat mengambil gambar atau video melalui aplikasi yang dikembangkan. Pada modul ini akan dibahas cara untuk mengambil gambar dan video.

Ada beberapa hal yang perlu diperhatikan sebelum membuat aplikasi yang menggunakan Camera API, antara lain:

1. Ketersediaan Camera -> karena kita akan menggunakan kamera pada perangkat mobile yang menggunakan aplikasi yang kita kembangkan, maka kita pastikan aplikasi terinstal pada perangkat mobile yang memiliki kamera. Dengan cara mendeklarasikannya di android manifest, dengan cara

|  |  |  |
| --- | --- | --- |
| **Camera Permission** | | |
| <uses-permission android:name="android.permission.CAMERA" /> | | Jika aplikasi yang dikembangkan akan menggunakan Camera pada Android Device maka harus menulis permission ini pada manifest. |
| **Camera Feature** | | |
| <uses-feature android:name="android.hardware.camera" /> | | Aplikasi menggunakan kamera harus menulis camera feature pada persmission. Untuk feature dari kamera lebih detail dapat dilihat pada link (<https://developer.android.com/guide/topics/manifest/uses-feature-element.html#hw-features>). Salah satu feature camera yang dapat digunakan adalah autofocus, penggunaan flash pada kamera. |
| **Storage Permission** | | |
| <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" /> | | Jika aplikasi kita akan menyimpan gambar atau video pada SDCard (External Storage), harus menulis permission ini pada manifest |
| **Audio Recording Permission** | | |
| <uses-permission android:name="android.permission.RECORD\_AUDIO" /> | Untuk merekam audio dengan video capture, aplikasi yang dikembangkan harus menulis permission untuk merekam Audio | |

1. Hal kedua yang harus diperhatikan ketika mengembangkan aplikasi menggunakan Camera API adalah apakah aplikasi yang dikembangkan hanya berguna untuk mengambil gambar atau aplikasi yang dikembangkan menggunakan fitur atau tampilan yang telah kita *custom.* Jika akan mengcustom fitur pada android dapat dilihat pada link (https://developer.android.com/guide/topics/media/camera.html#custom-camera)
2. Hal terakhir pada penggunaan Camera API adalah apakah image atau video yang telah kita ambil akan disimpan atau hanya ditampilkan/di-*preview* pada aplikasi kita namun tidak disimpan. Jika bertujuan untuk di-*shared* maka image/video yang sudah diambil harus disimpan pada umumnya pada external storage agar space untuk sistem pada internal storage tetap terjaga/tidak penuh dan memungkinkan user mengakses image/video tersebut tanpa handphone.

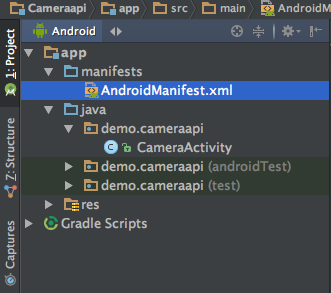
**2. Praktikum Camera API untuk Mengambil Gambar Dan di simpan hasilnya pada external storage**

Karena ini merupakan modul lanjutan dari modul camera 1 maka tidak perlu membuat project baru akan tetapi kita melanjutkan project sebelumnya. Pada modul yang sebelumnya, telah dibuat aplikasi dengan menggunakan Camera API seperti pada Gambar 1 dan 2. Namun, image yang didapat berukuran kecil, oleh karena itu kita pada modul ini akan membuat aplikasi yang dapat mengakses image/gambar dengan ukuran yang penuh.

|  |  |
| --- | --- |
| **Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-10-18-095246.png** | **Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-10-18-095352.png** |
| Gambar 1. Tampilan Proses Mengambil Image/Gambar | Gambar 2. Hasil Pengambilan Gambar |

Tahapan yang dilakukan adalah :

1. Menulis pada manifest untuk permission menulis pada external storage dengan cara sebagai berikut:

****

Gambar 3. Letak AndroidManifest Pada Project Android

|  |
| --- |
| **AndroidManifest.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="demo.cameraapi">  <!-- Akses camera-->  <uses-feature android:name="android.hardware.camera" />  <!-- Menyimpan pada 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:supportsRtl="true"  android:theme="@style/AppTheme">  <activity android:name=".CameraActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN"/>  <category android:name="android.intent.category.LAUNCHER"/>  </intent-filter>  </activity>  </application>  </manifest> |

1. Langkah berikutnya adalah mendeteksi letak dari folder pada External storage untuk menyimpan gambar dengan menggunakan code :

|  |
| --- |
| New File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_PICTURES), "CameraDemo") |
| Pada code tersebut bertujuan untuk mendapatkan letak dari folder “CameraDemo” pada external storage. Jika folder atau direktori tidak ada maka akan dibuat direktori/folder pada external srorage dengan kode sebagai berikut  // Buat directori tidak direktori tidak eksis  if (!mediaStorageDir.exists()) {  if (!mediaStorageDir.mkdirs()) {  Log.d("CameraDemo", "Gagal membuat directori"+ "CameraDemo");  return null;  }  } |

Setelah folder dipastikan ada, langkah selanjutnya adalah membuat nama file untuk image/gambar yang akan disimpan dengan menggunakan code berikut:

|  |
| --- |
| // Membuat nama file  String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss", Locale.getDefault()).format(new Date());  File mediaFile = null;  mediaFile = new File(mediaStorageDir.getPath() + File.separator + "IMG\_" + timeStamp + ".jpg"); |

Fungsi final untuk mengecek folder dan membuat nama file diberi nama fungsi getMediaFileName().

|  |
| --- |
| getMediaFileName() |
| private static File getMediaFileName() {  // Lokasi External sdcard  File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_PICTURES), "CameraDemo");  // Buat directori tidak direktori tidak eksis  if (!mediaStorageDir.exists()) {  if (!mediaStorageDir.mkdirs()) {  Log.d("CameraDemo", "Gagal membuat directori"+ "CameraDemo");  return null;  }  }  // Membuat nama file  String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss", Locale.getDefault()).format(new Date());  File mediaFile = null;  mediaFile = new File(mediaStorageDir.getPath() + File.separator + "IMG\_" + timeStamp + ".jpg");  return mediaFile;  } |

1. Pada langkah ketiga adalah mengedit fungsi captureImage(), dengan menambahkan intent putExtra informasi letak/folder gambar dan nama file yang akan disimpan pada external storage.

|  |
| --- |
| private void captureImage() {  Intent takePictureIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  if (takePictureIntent.resolveActivity(getPackageManager()) != null) {  mFileURI = getMediaFileName();  takePictureIntent.putExtra(MediaStore.EXTRA\_OUTPUT, URI.fromFile(getMediaFileName(mFileURI)));  startActivityForResult(takePictureIntent, 100);  }  } |

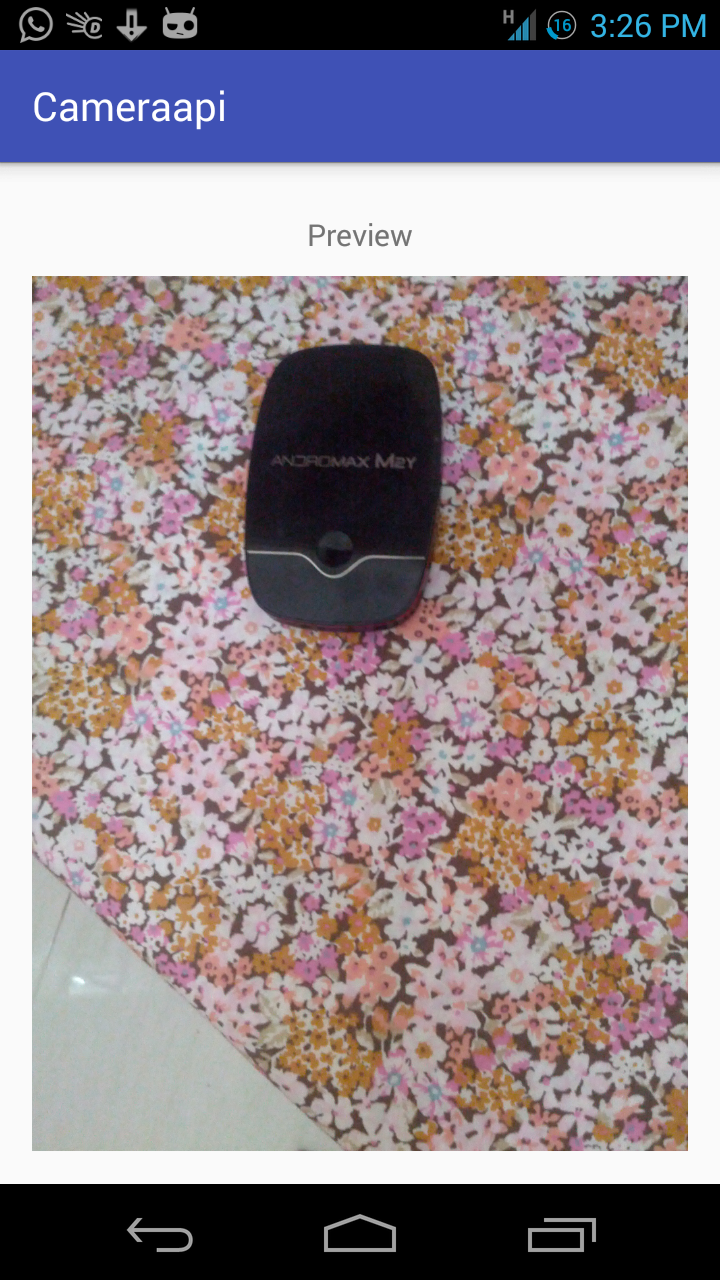
1. Aplikasi android pada tahap ini sudah dapat berjalan namun pada beberapa device/perangkat android yang memiliki memory yang kecil aplikasi akan force close hal ini dapat ditangani dengan melakukan rescale pada file image/gambar dengan cara :

|  |
| --- |
| /\*\*  \* activity result akan dipanggi setelah camera ditutup  \* \*/  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (requestCode == 100 && resultCode == RESULT\_OK) {  BitmapFactory.Options bmOptions = new BitmapFactory.Options();  // rescale bitmap jika aplikasi force close  // semakin besar ukuran rescale maka image/gambar yang ditampilkan semakin kecil  bmOptions.inSampleSize = 4;  Bitmap bitmap = BitmapFactory.decodeFile(mFileURI.getPath(), bmOptions);  mImageView.setVisibility(View.VISIBLE);  mImageView.setImageBitmap(bitmap);  }  } |

1. Final Code dari aplikasi camera yang menyimpan datanya pada file external adalah sebagai berikut:

|  |
| --- |
| package demo.cameraapi;  import android.content.Intent;  import android.content.pm.PackageManager;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.os.Bundle;  import android.os.Environment;  import android.provider.MediaStore;  import android.support.v7.app.AppCompatActivity;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.ImageView;  import android.widget.Toast;  import java.io.File;  import java.text.SimpleDateFormat;  import java.util.Date;  import java.util.Locale;  import static android.provider.CalendarContract.CalendarCache.URI;  import static demo.cameraapi.R.id.imgPreview;  public class CameraActivity extends AppCompatActivity {  Button mButtonPicture;  ImageView mImageView;  File mFileURI;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_camera);  mButtonPicture = (Button) findViewById(R.id.btnCapture);  mImageView = (ImageView) findViewById(imgPreview);  mButtonPicture.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  captureImage();  }  });  // Checking camera availability  if (!isDeviceSupportCamera()) {  Toast.makeText(getApplicationContext(), "Camera di device anda tidak tersedia", Toast.LENGTH\_LONG).show();  finish();  }  }  /\*  \* Capturing Camera Image will lauch camera app requrest image capture  \*/  private void captureImage() {  Intent takePictureIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  if (takePictureIntent.resolveActivity(getPackageManager()) != null) {  mFileURI = getMediaFileName();  takePictureIntent.putExtra(MediaStore.EXTRA\_OUTPUT, URI.fromFile(mFileURI));  startActivityForResult(takePictureIntent, 100);  }  }  /\*\*  \* activity result akan dipanggi setelah camera ditutup  \* \*/  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (requestCode == 100 && resultCode == RESULT\_OK) {  BitmapFactory.Options bmOptions = new BitmapFactory.Options();  // rescale bitmap jika aplikasi force close  // semakin besar ukuran rescale maka image/gambar yang ditampilkan semakin kecil  bmOptions.inSampleSize = 4;  Bitmap bitmap = BitmapFactory.decodeFile(mFileURI.getPath(), bmOptions);  mImageView.setVisibility(View.VISIBLE);  mImageView.setImageBitmap(bitmap);  }  }  /\*\*  \* mengecek pada perangkat mobile memiliki kamera atau tidak  \* \*/  private boolean isDeviceSupportCamera() {  if (getApplicationContext().getPackageManager().hasSystemFeature(  PackageManager.FEATURE\_CAMERA)) {  // this device has a camera  return true;  } else {  // no camera on this device  return false;  }  }  private static File getMediaFileName() {  // Lokasi External sdcard  File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_PICTURES), "CameraDemo");  // Buat directori tidak direktori tidak eksis  if (!mediaStorageDir.exists()) {  if (!mediaStorageDir.mkdirs()) {  Log.d("CameraDemo", "Gagal membuat directory"+ "CameraDemo");  return null;  }  }  // Membuat nama file  String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss", Locale.getDefault()).format(new Date());  File mediaFile = null;  mediaFile = new File(mediaStorageDir.getPath() + File.separator + "IMG\_" + timeStamp + ".jpg");  return mediaFile;  }  } |

1. Tampilan aplikasi saat menampilkan hasil pengambilan dapat dilihat pada Gambar 4.



Gambar 4. Hasil Akhir Pengambilan Gambar yang disimpan di External Storage

**3. Praktikum Load Image Menggunakan Glide dan Menambah ScroolView**

JIka tidak menggunakan library untuk load image, maka aplikasi yang dikembangkan akan sangat rawan terjadi force close oleh karena itu akan lebih baik dengan menggunakan library untuk melakukan load image. Library yang akan digunakan pada praktikum ini adalah Glide. Cara penggunakan Glide adalah dengan menambahkan pada build Gradle selanjutnya dilakukan sync, jika tidak terjadi error maka Glide siap digunakan. Contohnya dapat dilihat pada dibawah ini.

|  |
| --- |
| build.gradle |
| apply plugin: 'com.android.application'  android {  compileSdkVersion 23  buildToolsVersion "23.0.3"  defaultConfig {  applicationId "demo.cameraapi"  minSdkVersion 14  targetSdkVersion 23  versionCode 1  versionName "1.0"  testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  }  buildTypes {  release {  minifyEnabled false  proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  }  }  }  dependencies {  compile fileTree(dir: 'libs', include: ['\*.jar'])  androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {  exclude group: 'com.android.support', module: 'support-annotations'  })  compile 'com.android.support:appcompat-v7:23.4.0'  testCompile 'junit:junit:4.12'  **compile 'com.github.bumptech.glide:glide:3.7.0'**  } |

Salanjutnya ubah fungsi onActivityResult di kelas CameraActivity.java seperti berikut:

|  |
| --- |
| @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (requestCode == 100 && resultCode == RESULT\_OK) {  // BitmapFactory.Options bmOptions = new BitmapFactory.Options();  // // rescale bitmap jika aplikasi force close  // // semakin besar ukuran rescale maka image/gambar yang ditampilkan semakin kecil  // bmOptions.inSampleSize = 4;  //  // Bitmap bitmap = BitmapFactory.decodeFile(mFileURI.getPath(), bmOptions);  // mImageView.setVisibility(View.VISIBLE);  // mImageView.setImageBitmap(bitmap);  **Glide.with(getApplicationContext()).load(new File(mFileURI.getPath())).into(mImageView);**  **mImageView.setVisibility(View.VISIBLE);**  }  } |

Pada Gambar 4, terlihat bahwa button take picture berada di bawah gambar sehingga aplikasi tersebut tidak dapat melakukan capture image lagi. Jadi akan lebih baik jika layout yang sudah dibuat diberi scroll view seperti berikut.

|  |
| --- |
| activity\_camera.xml |
| <?xml version="1.0" encoding="utf-8"?>  <ScrollView  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:id="@+id/activity\_camera"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="demo.cameraapi.CameraActivity">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <TextView  android:id="@+id/tvPreview"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Preview"  android:layout\_centerHorizontal="true"  android:padding="10dp"  android:textSize="15dp"/>  <!-- Display Image yang diambil -->  <ImageView  android:id="@+id/imgPreview"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:visibility="gone"  android:layout\_below="@+id/tvPreview"/>  <!-- Untuk mengambil Image/Gambar -->  <Button  android:id="@+id/btnCapture"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Take a Picture"  android:layout\_marginBottom="10dp"  android:layout\_below="@+id/imgPreview"  android:layout\_centerHorizontal="true"/>  </RelativeLayout>  </ScrollView> |

**4. Praktikum Melakukan Pemrosesan Image**

Setelah membuat aplikasi yang menggunakan Camera API, menyimpan hasilnya di external storage dan menampilkannya pada ImageView. Pada tahap selanjutnya aplikasi tersebut akan dikembangkan sehingga dapat melakukan pemrosesan image/gambar dari gambar yang sudah disimpan. Langkah-langkahnya adalah sebagai berikut:

1. Ubah Layout tampilan menjadi seperti berikut



Gambar 5. Preview Layout Untuk Pemrosesan Image

|  |
| --- |
| activity\_camera.xml |
| <?xml version="1.0" encoding="utf-8"?>  <ScrollView  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:id="@+id/activity\_camera"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="demo.cameraapi.CameraActivity">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <TextView  android:id="@+id/tvPreview"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Preview"  android:layout\_centerHorizontal="true"  android:padding="10dp"  android:textSize="15dp"/>  <!-- Display Image yang diambil -->  <ImageView  android:id="@+id/imgPreview"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:visibility="gone"  android:layout\_below="@+id/tvPreview"/>  <!-- Untuk mengambil Image/Gambar -->  <Button  android:id="@+id/btnCapture"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Take a Picture"  android:layout\_marginBottom="10dp"  android:layout\_below="@+id/imgPreview"  android:layout\_centerHorizontal="true"/>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@+id/btnCapture"  android:gravity="center"  android:orientation="horizontal">  <Button  android:id="@+id/btnGray"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Gray"/>  <Button  android:id="@+id/btnNegatif"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Negatif"/>  </LinearLayout>  </RelativeLayout>  </ScrollView> |

1. Langkah selanjutnya, ubah Camera Activity seperti berikut

|  |
| --- |
| CameraActivity.java |
| package demo.cameraapi;  import android.content.Intent;  import android.content.pm.PackageManager;  import android.graphics.Bitmap;  import android.graphics.Color;  import android.os.Bundle;  import android.os.Environment;  import android.provider.MediaStore;  import android.support.v7.app.AppCompatActivity;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.ImageView;  import android.widget.Toast;  import com.bumptech.glide.Glide;  import java.io.File;  import java.text.SimpleDateFormat;  import java.util.Date;  import java.util.Locale;  import static android.provider.CalendarContract.CalendarCache.URI;  import static demo.cameraapi.R.id.imgPreview;  public class CameraActivity extends AppCompatActivity {  Button mButtonPicture, mButtonGray, mButtonNegatif;  ImageView mImageView;  File mFileURI;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_camera);  mButtonPicture = (Button) findViewById(R.id.btnCapture);  mButtonGray = (Button) findViewById(R.id.btnGray);  mButtonNegatif = (Button) findViewById(R.id.btnNegatif);  mImageView = (ImageView) findViewById(imgPreview);  mButtonPicture.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  captureImage();  }  });  mButtonGray.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  mImageView.buildDrawingCache();  Bitmap bitmap = mImageView.getDrawingCache();  Bitmap newBitmap = Bitmap.createBitmap(bitmap.getWidth(), bitmap.getHeight(), Bitmap.Config.ARGB\_8888);  for (int i = 0; i < bitmap.getWidth(); i++) {  for (int j = 0; j < bitmap.getHeight(); j++) {  int p = bitmap.getPixel(i,j);  int r = Color.red(p);  int g = Color.green(p);  int b = Color.blue(p);  r = g = b = (int) (r+g+b)/3;  newBitmap .setPixel(i, j, Color.rgb(r, g, b));  }  }  mImageView.setImageBitmap(newBitmap );  }  });  mButtonNegatif.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  mImageView.buildDrawingCache();  Bitmap bitmap = mImageView.getDrawingCache();  Bitmap newBitmap = Bitmap.createBitmap(bitmap.getWidth(), bitmap.getHeight(), Bitmap.Config.ARGB\_8888);  for (int i = 0; i < bitmap.getWidth(); i++) {  for (int j = 0; j < bitmap.getHeight(); j++) {  int p = bitmap.getPixel(i,j);  int r = 255 - Color.red(p);  int g = 255 - Color.green(p);  int b = 255 - Color.blue(p);  newBitmap .setPixel(i, j, Color.argb(255, r, g, b));  }  }  mImageView.setImageBitmap(newBitmap);  }  });  // Checking camera availability  if (!isDeviceSupportCamera()) {  Toast.makeText(getApplicationContext(), "Camera di device anda tidak tersedia", Toast.LENGTH\_LONG).show();  finish();  }  }  /\*  \* Capturing Camera Image will lauch camera app requrest image capture  \*/  private void captureImage() {  Intent takePictureIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  if (takePictureIntent.resolveActivity(getPackageManager()) != null) {  mFileURI = getMediaFileName();  takePictureIntent.putExtra(MediaStore.EXTRA\_OUTPUT, URI.fromFile(mFileURI));  startActivityForResult(takePictureIntent, 100);  }  }  /\*\*  \* activity result akan dipanggi setelah camera ditutup  \* \*/  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (requestCode == 100 && resultCode == RESULT\_OK) {  // BitmapFactory.Options bmOptions = new BitmapFactory.Options();  // // rescale bitmap jika aplikasi force close  // // semakin besar ukuran rescale maka image/gambar yang ditampilkan semakin kecil  // bmOptions.inSampleSize = 4;  //  // Bitmap bitmap = BitmapFactory.decodeFile(mFileURI.getPath(), bmOptions);  // mImageView.setVisibility(View.VISIBLE);  // mImageView.setImageBitmap(bitmap);  Glide.with(getApplicationContext()).load(new File(mFileURI.getPath())).into(mImageView);  mImageView.setVisibility(View.VISIBLE);  mButtonNegatif.setVisibility(View.VISIBLE);  mButtonGray.setVisibility(View.VISIBLE);  }  }  /\*\*  \* mengecek pada perangkat mobile memiliki kamera atau tidak  \* \*/  private boolean isDeviceSupportCamera() {  if (getApplicationContext().getPackageManager().hasSystemFeature(  PackageManager.FEATURE\_CAMERA)) {  // this device has a camera  return true;  } else {  // no camera on this device  return false;  }  }  private static File getMediaFileName() {  // Lokasi External sdcard  File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_PICTURES), "CameraDemo");  // Buat directori tidak direktori tidak eksis  if (!mediaStorageDir.exists()) {  if (!mediaStorageDir.mkdirs()) {  Log.d("CameraDemo", "Gagal membuat directory"+ "CameraDemo");  return null;  }  }  // Membuat nama file  String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss", Locale.getDefault()).format(new Date());  File mediaFile = null;  mediaFile = new File(mediaStorageDir.getPath() + File.separator + "IMG\_" + timeStamp + ".jpg");  return mediaFile;  }  } |

Hasil Akhirnya dapat dilihat pada Gambar 6, Gambar 7, Gambar 8 dan Gambar 9.

|  |  |
| --- | --- |
| Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-11-09-234604.png | Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-11-09-234621.png |
| Gambar 6. Color Image | Gambar 7. Hasil Grayscale dari Color Image |

|  |  |
| --- | --- |
| **Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-11-09-234522.png** | **Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-11-09-234537.png** |
| Gambar 8. Color Image | Gambar 9. Hasil Negatid Image dari Color Image |