MODUL : CAMERA API

**A. Tujuan**

Pada modul ini akan membahas tentang camera api, dengan tujuan agar mahasiswa:

* Mengetahui cara untuk mengakses Camera pada perangkat android
* Dapat mengimplementasikan Camera API untuk membuat aplikasi android

**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

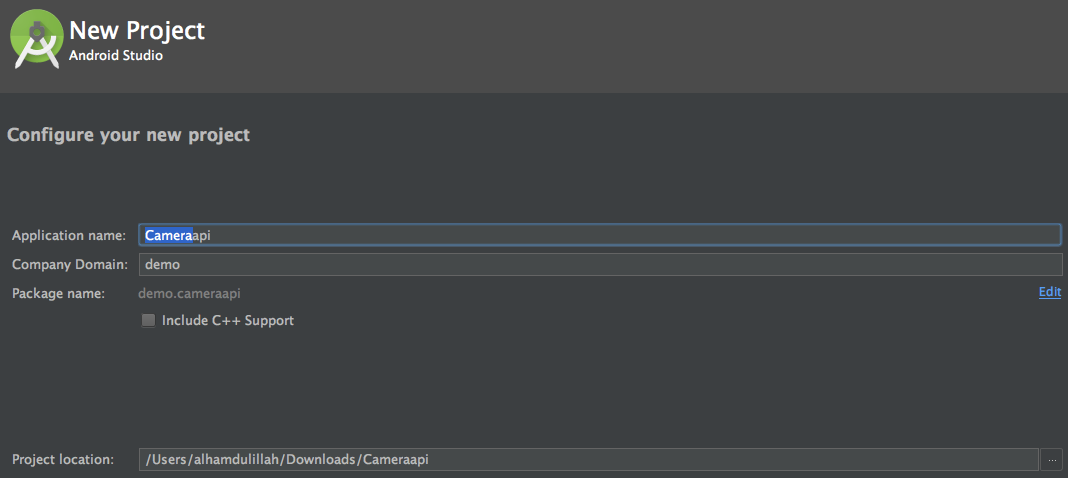
|  |  |  |
| --- | --- | --- |
| **Camera Permission** | | |
| <uses-permission android:name="android.permission.CAMERA" /> | | Jika aplikasi yang dikembangkan akan menggunakan Camera pada Android Device maka harus menulis permission ini pada manifest. |
| **Camera Feature** | | |
| <uses-feature android:name="android.hardware.camera" /> | | 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 untutk sistem pada internal storage tetap terjaga/tidak penuh dan memungkinkan user mengakses image/video tersebut tanpa handphone.

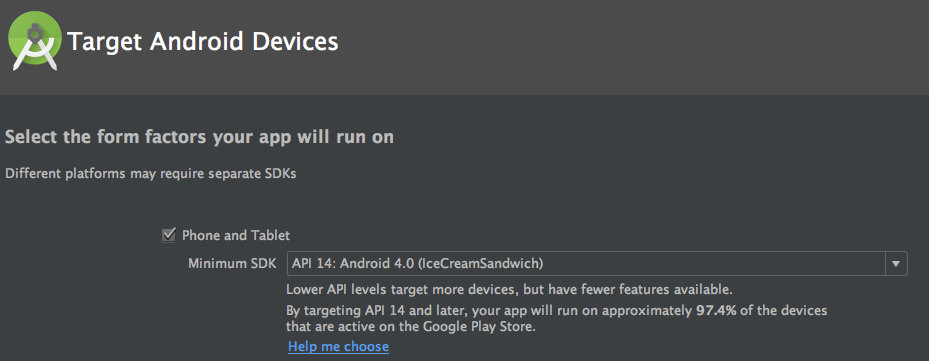
**2. Praktikum Camera API untuk Mengambil Gambar**

1. Pada langkah pertama buat project baru dengan empty activity, seperti berikut :

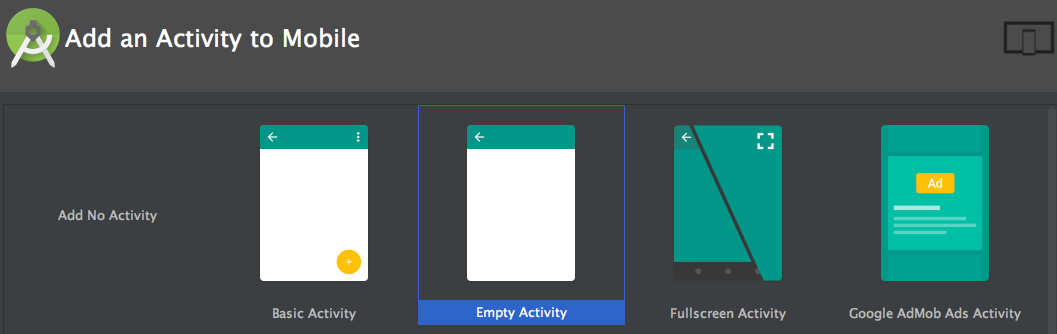
**StartNewProject->Configure Your New Project-> Setting Target Device -> Pilih Empty Activity-> Setting Activiy Name -> Finish**

****

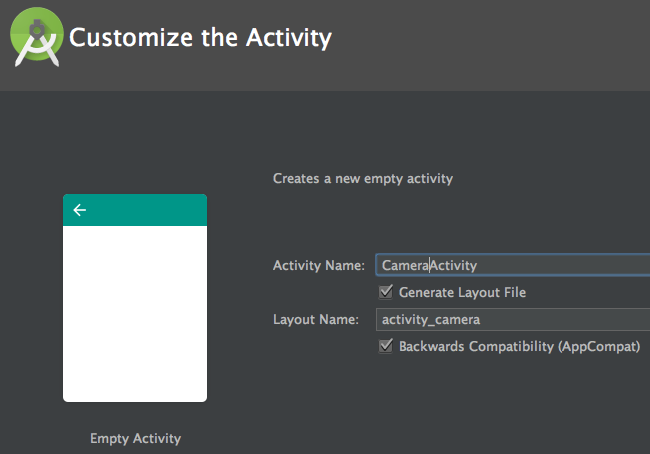
Gambar 1. Configure Your New Project



Gambar 2. Setting Target Device

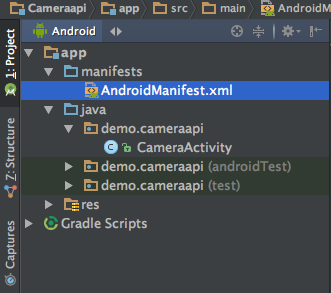


Gambar 3. Pilih Empty Activity



Gambar 4. Setting Activity Name

1. Pada langkah berikutnya tambahkan permission pada android manifest, seperti berikut :

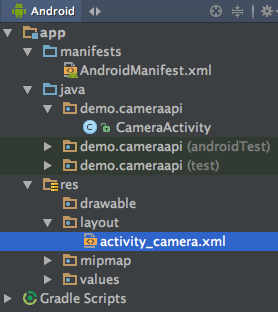
****

Gambar 5. Letak AndroidManifest Pada Project Android

Tabel 1. AndroidManifest.xml

|  |
| --- |
| **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>** |

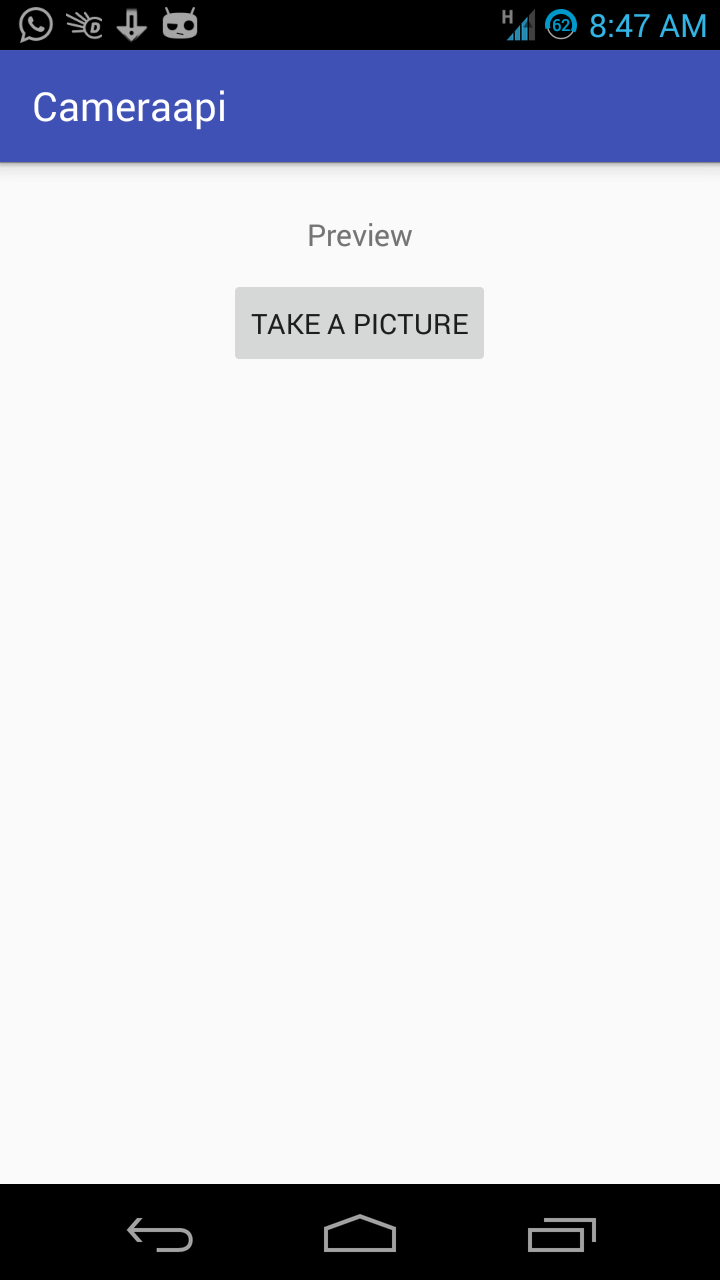
1. Setelah melakukan setting user permission pada manifest langkah selanjutnya adalah membuat layout aplikasi. Letak layout aplikasi dapat dilihat pada Gambar 6.



Gambar 6. Letak Layout Aplikasi Pada Project Android

Tabel 2. Layout activity\_camera.xml

|  |
| --- |
| activity\_camera.xml |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android: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">  <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> |



Gambar 7. Tampilan Layout activity\_camera.xml

1. Setelah membuat layout atau tampilan aplikasi pada langkah selanjutnya adalah membuat code untuk mendefinisikan cara kerja aplikasi pada CameraActivity.java. Terdapat beberapa tahapan untuk membuat code pada CameraActivity.java, yaitu:
2. Mendefinisikan setiap komponen/widget pada layout yang akan dipakai seperti berikut:

|  |
| --- |
| CameraActivity.java |
| package demo.cameraapi;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.widget.Button;  import android.widget.ImageView;  public class CameraActivity extends AppCompatActivity {  Button mButtonPicture;  ImageView mImageView;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_camera);  mButtonPicture = (Button) findViewById(R.id.btnCapture);  mImageView = (ImageView) findViewById(R.id.imgPreview);  }  } |

1. Mengecek apakah device yang akan menggunakan aplikasi yang dikembangkan memiliki camera. Hal ini dapat dilakukan secara otomatis dan manual:
   1. Secara Otomatis hal ini telah di handle oleh <uses-feature android:name="android.hardware.camera" /> pada manifest. Dengan menulis ini pada manifest maka google play akan secara otomatis mencegah aplikasi ini untuk diinstal pada handphone yang tidak memiliki kamera.
   2. Untuk mengecek ketersediaan kamera juga dapat dilakukan secara manual dengan cara:

|  |
| --- |
| /\*\*  \* 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;  }  } |

1. Setelah memastikan device memiliki kamera langkah selanjutnya adalah membuat fungsi untuk mengambil image/gambar.

|  |
| --- |
| /\*  \* Capturing Camera Image will lauch camera app request image capture  \*/  private void captureImage() {  Intent takePictureIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  if (takePictureIntent.resolveActivity(getPackageManager()) != null) {  startActivityForResult(takePictureIntent, 100);  }  } |

1. Setalah kita mengambil gambar maka pada langkah selanjutnya adalah menampilkan hasilnya pada imageView.

|  |
| --- |
| /\*\*  \* Receiving activity result method will be called after closing the camera  \* \*/  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (requestCode == 100 && resultCode == RESULT\_OK) {  Bundle extras = data.getExtras();  Bitmap imageBitmap = (Bitmap) extras.get("data");  mImageView.setImageBitmap(imageBitmap);  }  } |

Aplikasi camera mengencode image ketika mengembalikan intent yang dikirim ke onActivityResult() sebagai Bitmap yang berukuran kecil dengan key “data”. Bitmap tersebut lebih cocok untuk icon. Untuk mendapat image dengan ukuran yang lebih besar akan dipelajari pada modul selanjutnya.

1. Final Code untuk mengambil gambar tanpa disimpan pada external storage adalah

|  |
| --- |
| package demo.cameraapi;  import android.content.Intent;  import android.content.pm.PackageManager;  import android.graphics.Bitmap;  import android.os.Bundle;  import android.provider.MediaStore;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;  import android.widget.ImageView;  import android.widget.Toast;  import static demo.cameraapi.R.id.imgPreview;  public class CameraActivity extends AppCompatActivity {  Button mButtonPicture;  ImageView mImageView;  @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) {  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) {  Bundle extras = data.getExtras();  Bitmap imageBitmap = (Bitmap) extras.get("data");  mImageView.setVisibility(View.VISIBLE);  mImageView.setImageBitmap(imageBitmap);  }  }  /\*\*  \* 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;  }  }  } |

1. **Proses pengambilan gambar dan proses untuk menampilkan gambar dapat dilihat pada Gambar**

|  |  |
| --- | --- |
| **Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-10-18-095246.png** | **Alhamdulillah:Users:alhamdulillah:Desktop:device-2016-10-18-095352.png** |
| Gambar 8. Tampilan Proses Mengambil Image/Gambar | Gambar 9. Hasil Pengambilan Gambar |