CODING JAVA MAIN ACTIVITY

ANDROID STUDIO

```
package com.app.lightcreative;

import android.Manifest;

import android.annotation.SuppressLint;

import android.app.Activity;

import android.os.Bundle;

import android.util.Log;

import android.view.KeyEvent;

import android.view.View;

import android.webkit.ValueCallback;

import android.webkit.WebChromeClient;

import android.webkit.WebResourceError;

import android.webkit.WebResourceRequest;

import android.webkit.WebSettings;

import android.webkit.WebView; // memasukan class WebView

import android.webkit.WebViewClient; //memasukan class WebViewClient

import android.widget.ProgressBar;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import com.light.creative.R;


public class MainActivity extends AppCompatActivity {

   private WebView view; //membuat variabel view agar bisa akses method

   private String DEFAULT_ERROR_PAGE_PATH =
"file:///android_asset/error1.html";
```

```
    //private static String webview_url   = "file:///android_asset/error1.html";    // web
address or local file location you want to open in webview
    private static String file_type     = "image/*";    // file types to be allowed for
upload
    private boolean multiple_files      = true;         // allowing multiple file upload
    WebView webView;
    private String mCM;
    private ValueCallback<Uri> mUM;
    private ValueCallback<Uri[]> mUMA;
    private ValueCallback<Uri> mUploadMessage;
    private Uri mCapturedImageURI = null;
    private ValueCallback<Uri[]> mFilePathCallback;
    private String mCameraPhotoPath;
    private static final int INPUT_FILE_REQUEST_CODE = 1;
    private static final int FILECHOOSER_RESULTCODE = 1;
    private static final String TAG = MainActivity.class.getSimpleName();
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            if (requestCode != INPUT_FILE_REQUEST_CODE || mFilePathCallback
== null) {
                super.onActivityResult(requestCode, resultCode, data);
                return;
            }
            Uri[] results = null;
            // Check that the response is a good one
            if (resultCode == Activity.RESULT_OK) {
                if (data == null) {
                    // If there is not data, then we may have taken a photo
                    if (mCameraPhotoPath != null) {
                        results = new Uri[]{Uri.parse(mCameraPhotoPath)};
                    }
                }   else {
                    String dataString = data.getDataString();
                    if (dataString != null) {
                        results = new Uri[]{Uri.parse(dataString)};
                    }
                }
            }
            mFilePathCallback.onReceiveValue(results);
            mFilePathCallback = null;
        } else if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.KITKAT) {
            if (requestCode != FILECHOOSER_RESULTCODE || mUploadMessage ==
null) {
                super.onActivityResult(requestCode, resultCode, data);
                return;
            }
            if (requestCode == FILECHOOSER_RESULTCODE) {
                if (null == this.mUploadMessage) {
                    return;
                }
                Uri result = null;
                try {
                    if (resultCode != RESULT_OK) {
                        result = null;
                    } else {
                        // retrieve from the private variable if the intent is null
                        result = data == null ? mCapturedImageURI : data.getData();
                    }
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(), "activity :" + e,
                      Toast.LENGTH_LONG).show();
                }
                mUploadMessage.onReceiveValue(result);
                mUploadMessage = null;
            }
        }
        return;
    }

    private class MyBrowser extends WebViewClient {

        @SuppressLint("NewApi")
        @Override
        public void onReceivedError(WebView view, WebResourceRequest request,
WebResourceError error) {
            if (request.isForMainFrame() && error != null) {
                view.loadUrl(DEFAULT_ERROR_PAGE_PATH);
            }
        }

        @Override
        public void onReceivedError(WebView view, int errorCode, String description,
String failingUrl) {
            if (errorCode != WebViewClient.ERROR_UNSUPPORTED_SCHEME &&
errorCode != WebViewClient.ERROR_HOST_LOOKUP) {
                view.loadUrl(DEFAULT_ERROR_PAGE_PATH);
            }
        }

        @Override
        public void onPageStarted(WebView view, String url, Bitmap favicon) {
            super.onPageStarted(view, url, favicon);


        }
        @Override
        public boolean shouldOverrideUrlLoading(WebView view, String url) {
            view.loadUrl(url);
            return super.shouldOverrideUrlLoading(view, url);
        }
        public void onPageFinished(WebView view, String url) {
            super.onPageFinished(view, url);
            bar.setVisibility(View.GONE);
        }
    };



    ProgressBar bar;
    @SuppressLint({"SetJavaScriptEnabled", "WrongViewCast"})
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        view = (WebView) this.findViewById(R.id.webView);
        webView = findViewById(R.id.webView);
        if (Build.VERSION.SDK_INT >= 23 &&
(ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED ||
ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED)) {

            ActivityCompat.requestPermissions(MainActivity.this, new String[]
{Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.CAMERA}, 1);

        }
        assert webView != null;

        WebSettings webSettings = webView.getSettings();

        webSettings.setJavaScriptEnabled(true);

        webSettings.setAllowFileAccess(true);

        webView.setWebChromeClient(new WebChromeClient() {

            private File createImageFile() throws IOException {

              String timeStamp = new
SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());

              String imageFileName = "JPEG_" + timeStamp + "_";

              File storageDir = Environment.getExternalStoragePublicDirectory(

                  Environment.DIRECTORY_PICTURES);

              File imageFile = File.createTempFile(

                  imageFileName,  /* prefix */

                  ".jpg",         /* suffix */

                  storageDir      /* directory */

              );

              return imageFile;

            }

            public boolean onShowFileChooser(WebView view, ValueCallback<Uri[]>
filePath, WebChromeClient.FileChooserParams fileChooserParams) {

              // Double check that we don't have any existing callbacks

              if (mFilePathCallback != null) {

                mFilePathCallback.onReceiveValue(null);

              }

              mFilePathCallback = filePath;

              Intent takePictureIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);

              if (takePictureIntent.resolveActivity(getPackageManager()) != null) {

                // Create the File where the photo should go

                File photoFile = null;

                try {
                  photoFile = createImageFile();
                  takePictureIntent.putExtra("PhotoPath", mCameraPhotoPath);
                } catch (IOException ex) {
                  // Error occurred while creating the File
                  Log.e(TAG, "Unable to create Image File", ex);
                }
                // Continue only if the File was successfully created
                if (photoFile != null) {
                  mCameraPhotoPath = "file:" + photoFile.getAbsolutePath();
                  takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
                      Uri.fromFile(photoFile));
                } else {
                  takePictureIntent = null;
                }
              }
              Intent contentSelectionIntent = new
Intent(Intent.ACTION_GET_CONTENT);
              contentSelectionIntent.addCategory(Intent.CATEGORY_OPENABLE);
              contentSelectionIntent.setType("image/*");
              Intent[] intentArray;
              if (takePictureIntent != null) {
                intentArray = new Intent[]{takePictureIntent};
              } else {
                intentArray = new Intent[0];
              }
              Intent chooserIntent = new Intent(Intent.ACTION_CHOOSER);
              chooserIntent.putExtra(Intent.EXTRA_INTENT, contentSelectionIntent);
              chooserIntent.putExtra(Intent.EXTRA_TITLE, "Image Chooser");
              chooserIntent.putExtra(Intent.EXTRA_INITIAL_INTENTS,
intentArray);
              startActivityForResult(chooserIntent, INPUT_FILE_REQUEST_CODE);
              return true;

            }

            public void openFileChooser(ValueCallback<Uri> uploadMsg, String
acceptType) {

              mUploadMessage = uploadMsg;

              // Create AndroidExampleFolder at sdcard

              // Create AndroidExampleFolder at sdcard

              File imageStorageDir = new File(

                  Environment.getExternalStoragePublicDirectory(

                      Environment.DIRECTORY_PICTURES)

                  , "AndroidExampleFolder");

              if (!imageStorageDir.exists()) {

                // Create AndroidExampleFolder at sdcard

                imageStorageDir.mkdirs();

              }

              // Create camera captured image file path and name

              File file = new File(

                  imageStorageDir + File.separator + "IMG_"

                      + String.valueOf(System.currentTimeMillis())

                      + ".jpg");

              mCapturedImageURI = Uri.fromFile(file);

              // Camera capture image intent

              final Intent captureIntent = new Intent(

                  android.provider.MediaStore.ACTION_IMAGE_CAPTURE);

              captureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
mCapturedImageURI);

              Intent i = new Intent(Intent.ACTION_GET_CONTENT);

              i.addCategory(Intent.CATEGORY_OPENABLE);

              i.setType("image/*");

              // Create file chooser intent

              Intent chooserIntent = Intent.createChooser(i, "Image Chooser");
```

```
            // Set camera intent to file chooser
            chooserIntent.putExtra(Intent.EXTRA_INITIAL_INTENTS
                    , new Parcelable[] { captureIntent });
            // On select image call onActivityResult method of activity
            startActivityForResult(chooserIntent, FILECHOOSER_RESULTCODE);
        }
        public void openFileChooser(ValueCallback<Uri> uploadMsg,
                                    String acceptType,
                                    String capture) {
            openFileChooser(uploadMsg, acceptType);
        }
    });

    bar=(ProgressBar) findViewById(R.id.progressBar2);
    view.getSettings().setJavaScriptEnabled(true);
    view.setWebViewClient(new MyBrowser());
    view.loadUrl("https://www.wayanrida.com/");
}

public boolean  onKeyDown(int keyCode, KeyEvent event) {
    //ketika disentuh tombol back
    if ((keyCode == KeyEvent.KEYCODE_BACK) && view.canGoBack()) {
        view.goBack(); //method goback(),untuk kembali ke halaman sebelumnya
        return true;
    }
    // Jika tidak ada halaman yang pernah dibuka
    // maka akan keluar dari activity (tutup aplikasi)
    return super.onKeyDown(keyCode, event);
 }
//public class Callback extends WebViewClient {
```

```
//}
```