

# ProjectLC

Project Aplikasi  
Berbasis Camera &  
Location

LB75  
Kelompok 2

2602056851		BENEDICT ZEVARNO CHRISTABEL
2602075983		EVANDER OCTAVIANUS LAYARDI
2502016090		GIAN GUIDO H.
2602198900		HOSANNA MEGAN PUTRA W.
2602198900		NATHANAEL HANSEL YAPUTRA
2602139103		RICHARD YOHANES

# PEMBAGIAN KERJA

## BENEDICT Z.

- Mengimplementasikan *geocoder* untuk menampilkan detail lokasi gambar dengan akurat seperti *latitude*, *longitude*, dan *address*, dan
- memperbaiki kesalahan *geocoder* yang awalnya menampilkan *current location* menjadi *last location*.

## EVANDER O. L.

- Membuat fungsi *delete* untuk gambar pada memori eksternal,
- menambahkan algoritma *intent* untuk flow pada aplikasi,
- membuat slide presentasi, dan
- melakukan *trial & error* aplikasi.

## GIAN GUIDO

- Mengimplementasikan *Timestamp* untuk menampilkan detail waktu gambar, seperti 2024-06-10 09:41:00, dan
- membuat perancangan *code* yang dapat menyambungkan aplikasi dengan kamera.

## HOSANNA M.

- Membuat algoritma sistematis kamera pada aplikasi, dan
- membuat perancangan *code* yang dapat menyambungkan aplikasi dengan kamera.

## NATHANAEL H.

- Membuat tampilan menu aplikasi yang berisi kata sambutan “Hello, BINUSIAN”, anggota kelompok dengan *text email* yang *direct* ke aplikasi email, tampilan *gallery* yang berisi foto-foto,
- membuat slide presentasi, dan
- mencoba dan mendokumentasi aplikasi.

## RICHARD Y.

- Menentukan dan mengimplementasikan database *sharedPreferences*, dan
- membuat perancangan *code* yang efisien.



## “LC”?

*Nama "LC" pada ProjectLC berasal dari dua kata kunci utama yaitu Location dan Camera. Kedua elemen ini menjadi inti dari aplikasi, menggabungkan kemampuan pengambilan gambar dengan pelacakan lokasi yang akurat.*

# Introduction ProjectLC

ProjectLC adalah aplikasi Android yang dirancang untuk memudahkan pengguna dalam mengambil dan menyimpan foto secara otomatis. Dengan menggunakan penyimpanan internal perangkat, ProjectLC tidak hanya menyimpan hasil gambar, tetapi juga menampilkan detail informasi penting seperti lokasi dan waktu pengambilan (time stamp).

# Cara Kerja Aplikasi

## Open ProjectLC app on device



Setelah membuka aplikasi, maka aplikasi akan menampilkan halaman utama

## Permission Camera & Location



Untuk kondisi awal, jika user baru menggunakan aplikasi untuk pertama kalinya, maka sistem aplikasi menampilkan opsi izin untuk akses lokasi dan kamera

## Take picture



Setelah user memberi izin untuk lokasi & kamera, maka aplikasi dapat membuka kamera dan dapat mengambil gambar



## Show Detail Image



Ketika gambar yang masih tersimpan dalam aplikasi, maka user dapat memilih suatu gambar secara spesifik dan aplikasi akan menampilkan detail dari gambar yaitu berupa koordinat, waktu pengambilan gambar dan lokasi.

## Delete picture



Aplikasi juga mendukung fitur delete gambar, yang memungkinkan user dapat menghapus gambar dari aplikasi

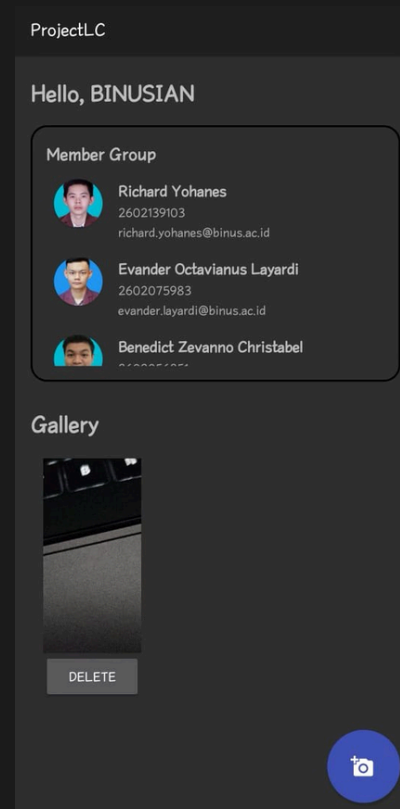
## Save picture



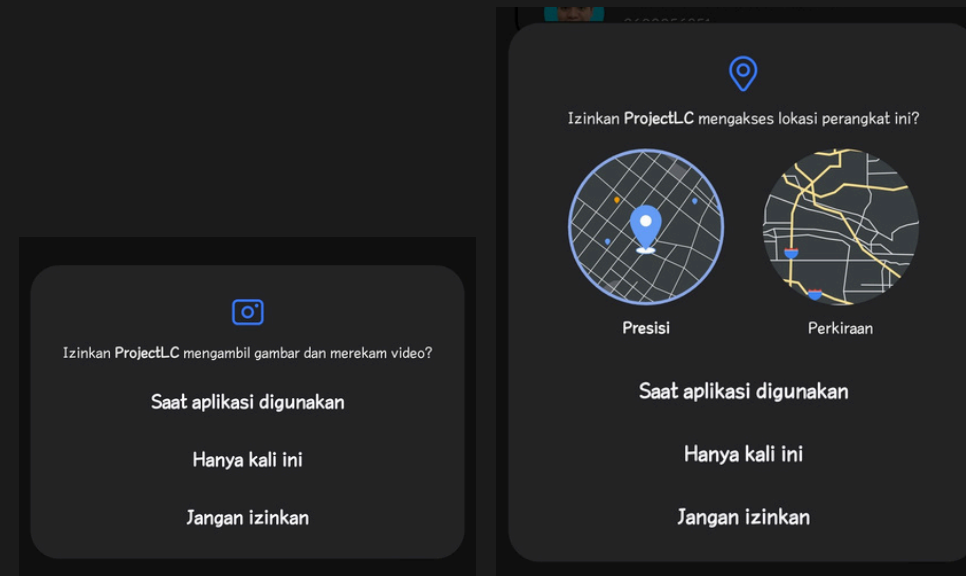
Setelah gambar diambil, maka aplikasi akan secara otomatis menyimpan kamera dengan menggunakan penyimpanan eksternal dalam aplikasi

# Cara Kerja Aplikasi

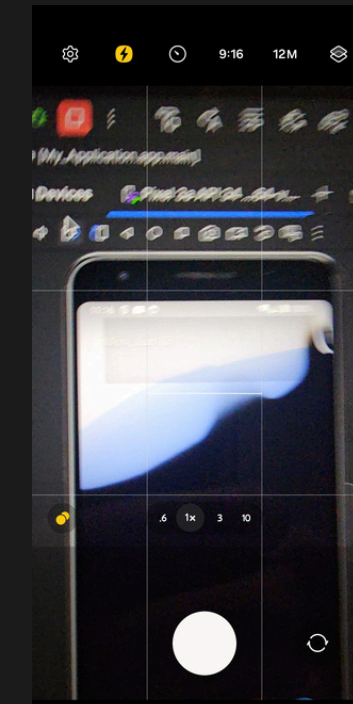
Open ProjectLC app  
on device



Permission Camera & Location



Take picture



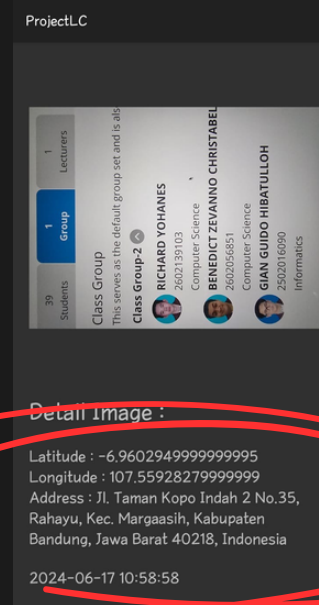
Save picture



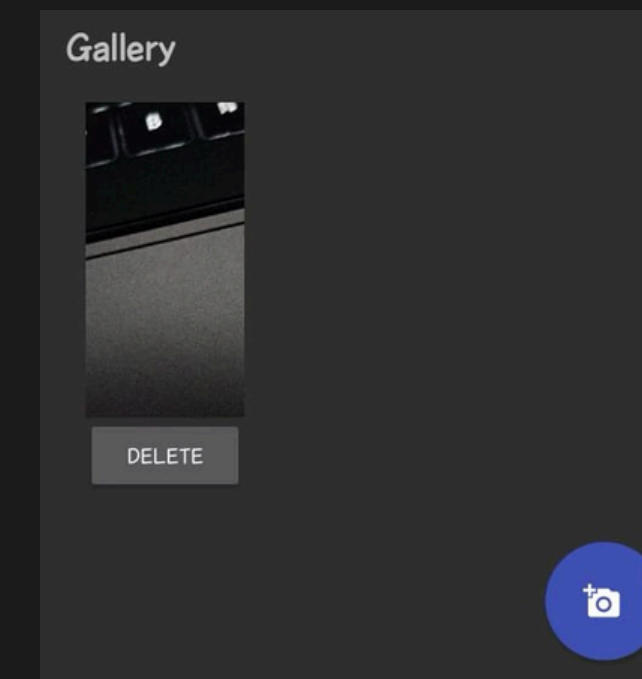
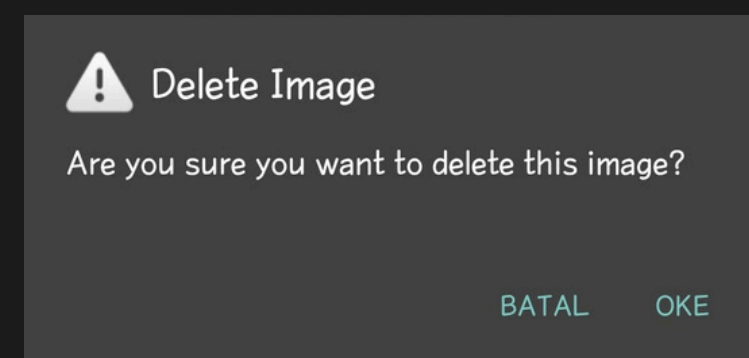
Delete picture

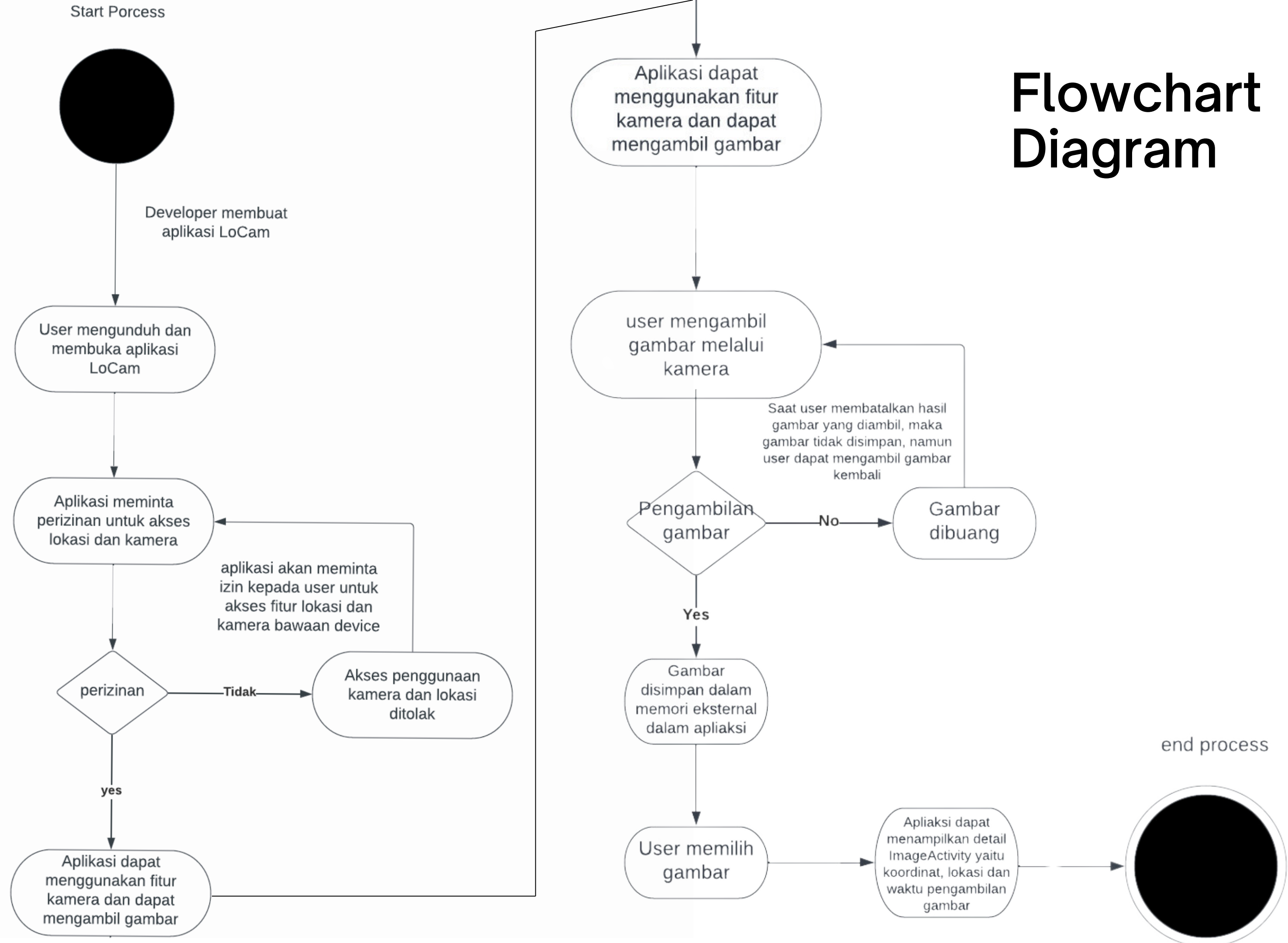


Show Detail Image



Project  
Aplikasi  
Berbasis  
Camera &  
Location





# CLASS YANG DIGUNAKAN DALAM APK

## MainActivity

Class MainActivity, sebuah aktivitas Android yang berfungsi agar pengguna untuk mengambil foto menggunakan kamera perangkat, menyimpan jalur gambar, lokasi, dan timestamp ke **SharedPreferences**, dan menampilkan gambar-gambar tersebut dalam tampilan grid. Aktivitas ini juga menangani permintaan izin untuk kamera dan lokasi, serta memastikan file gambar yang tidak valid dibersihkan saat aplikasi dijalankan.

## ImageAdapter

Class ImageAdapter, sebuah adaptor untuk tampilan grid yang mengelola dan menampilkan gambar dari daftar jalur gambar. Adaptor ini juga menangani klik pada gambar untuk membuka detail gambar di ImageActivity, serta menyediakan fungsi untuk menghapus gambar dengan konfirmasi dialog.

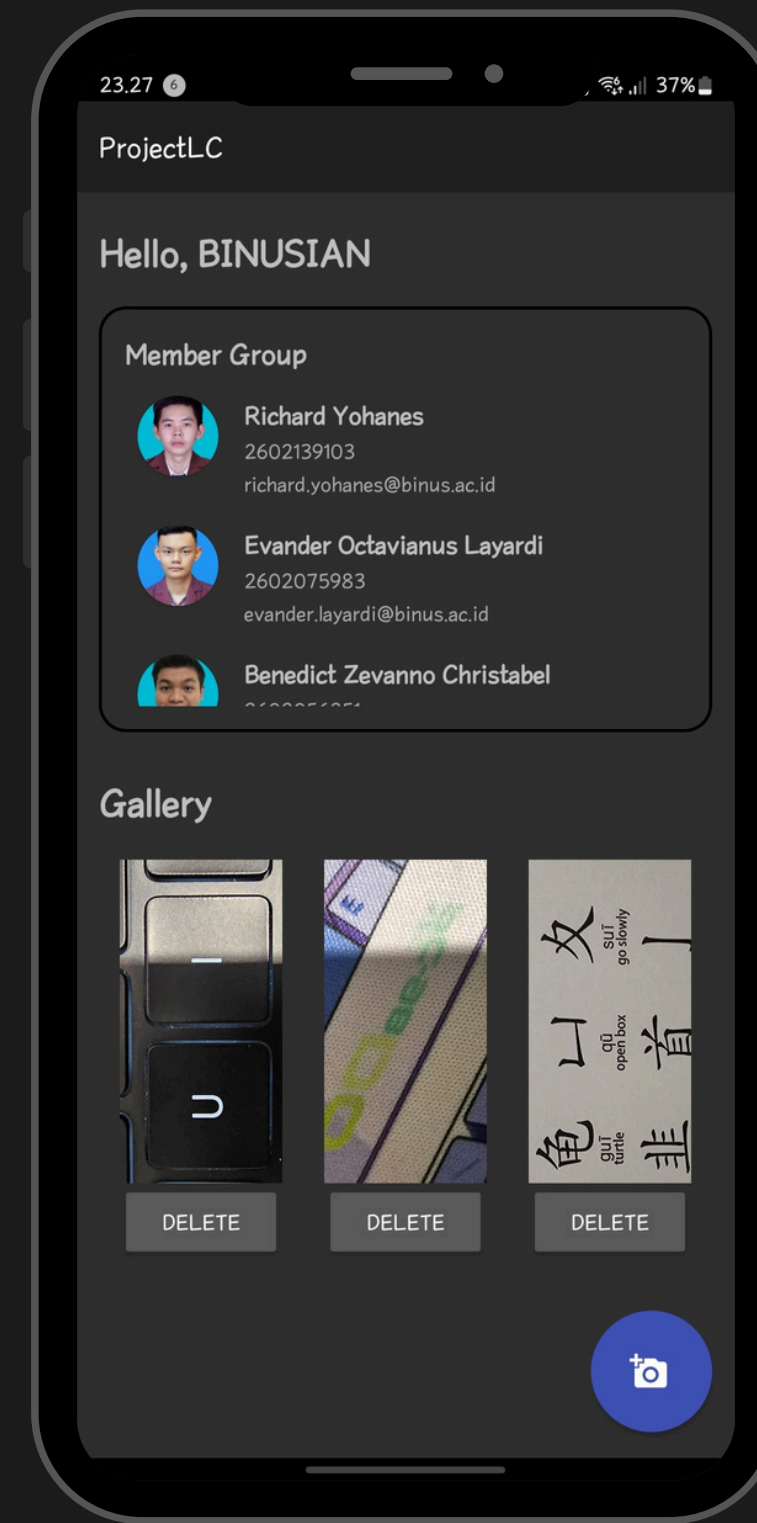
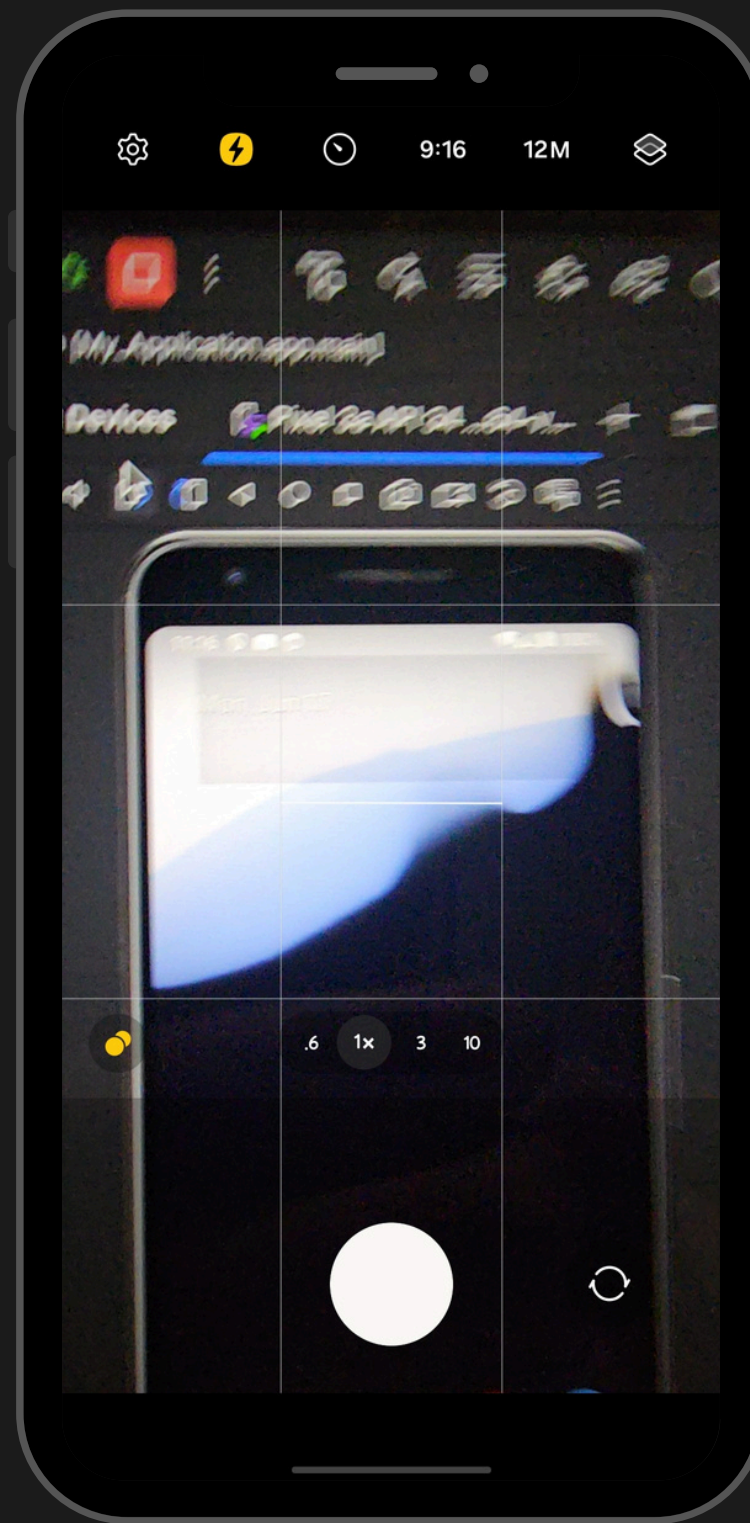
## ImageActivity

Class ImageActivity yang berfungsi untuk menampilkan gambar, lokasi, dan timestamp dari gambar yang diberikan melalui `Intent`. Aktivitas ini memuat gambar dalam ukuran yang sesuai dengan menggunakan teknik decoding bitmap yang efisien, serta menampilkan informasi lokasi dan waktu dari gambar tersebut jika tersedia.



# Mengambil dan Menyimpan Gambar

Aplikasi ini memiliki fitur untuk mengambil gambar melalui **Camera Intent** (kamera bawaan yang ada pada handphone). Kemudian gambar disimpan pada memori Internal melalui **Shared Preference**.



## METHOD UNTUK CAMERA

Kode diatas adalah bagianmethod dalam aplikasi Android untuk menangkap gambar menggunakan kamera perangkat. Pertama, metode membuat sebuah intent untuk memulai aplikasi kamera dengan **MediaStore.ACTION\_IMAGE\_CAPTURE**. Kemudian, kode memeriksa apakah ada aplikasi kamera yang tersedia untuk menangani intent tersebut. Jika ya, metode mencoba membuat file tempat gambar akan disimpan. Jika pembuatan file berhasil, URI dari file tersebut diperoleh dengan menggunakan **FileProvider**. URI ini kemudian dimasukkan ke intent sebagai tempat penyimpanan gambar yang akan diambil. Setelah itu, aplikasi kamera dipanggil dengan intent tersebut.

```
private void CaptureImageIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this,
                "com.example.android.fileprovider",
                photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent,
                REQUEST_IMAGE_CAPTURE);
            getLocation();
        }
    }
}
```

## Implementasi Code pada Program

Project Aplikasi  
Berbasis Camera  
& Location

## METHOD UNTUK MENYIMPAN GAMBAR

Metode **saveImagePath()** digunakan untuk menyimpan daftar path gambar ke dalam SharedPreferences. Pertama, SharedPreferences diakses dengan nama **SHARED\_PREFS** dalam mode **MODE\_PRIVATE**. Kemudian, sebuah editor SharedPreferences dibuat untuk mengedit data. Daftar path gambar dikonversi menjadi **Set<String>**, karena SharedPreferences mendukung **Set** tetapi tidak mendukung **List**. Setelah itu, **Set** ini disimpan ke SharedPreferences menggunakan kunci **IMAGE\_PATHS\_KEY**.

```
private void saveImagePath() {  
    SharedPreferences sharedPreferences = getSharedPreferences(SHARED_PREFS,  
MODE_PRIVATE);  
    SharedPreferences.Editor editor = sharedPreferences.edit();  
    Set<String> set = new HashSet<>(imagePaths);  
    editor.putStringSet(IMAGE_PATHS_KEY, set);  
    editor.apply();  
}
```

## Implementasi Code pada Program

Project Aplikasi  
Berbasis Camera  
& Location

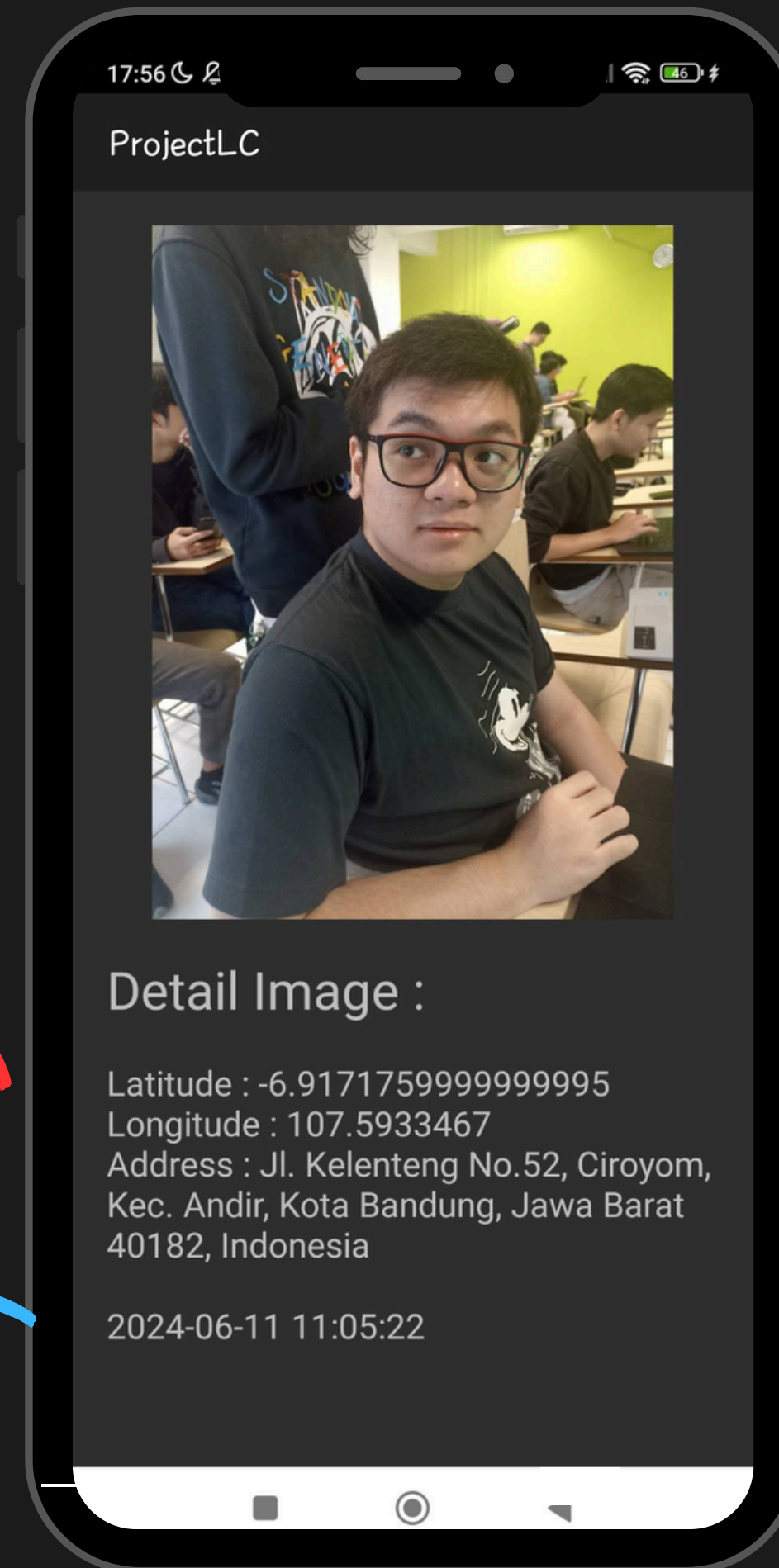
# Menampilkan Detail Gambar

## Lokasi

Aplikasi ini dapat mengetahui lokasi user menggunakan *Geocoder* ketika mengambil gambar dan menyimpannya ke dalam memori internal.

## Time Stamp

Aplikasi ini dapat mengetahui kapan gambar diambil dengan menyimpan waktu pengambilan gambar ke dalam memori internal melalui **SharedPrefence**.



ImageActivity

Bitmap decodeSampledBitmapFromFile

calculateInSampleSize

## FITUR APLIKASI

Project Aplikasi  
Berbasis Camera &  
Location



**Geocoder** dalam kode ini digunakan untuk mengonversi koordinat lokasi (*latitude* dan *longitude*) menjadi alamat yang dapat dibaca manusia. Dalam program berikut terdapat algoritma untuk menerapkan **GeoCoder** :

1. Program akan memastikan aplikasi memiliki izin untuk mengakses lokasi.
2. Menggunakan **fusedLocationProviderClient.getLastLocation()** untuk mendapatkan lokasi terakhir perangkat.
3. Dengan **Geocoder**, koordinat lokasi diubah menjadi alamat lengkap.
4. Detail lokasi (*latitude*, *longitude*, *address*) disimpan dalam daftar **imageLocations** untuk ditampilkan bersama dengan gambar.
5. Kemudian kita panggil method **getLocation()** pada **CaptureImageIntent()**, agar menampilkan lokasi pada saat suatu gambar yang tersimpan dan dipilih secara spesifik di dalam apk.

IMPLEMENTASI CODE PADA PROGRAM

# Location.

```
(photoFile != null) {  
    Uri photoURI = FileProvider.getUriForFile(context, this,  
        authority: "com.example.android.fileprovider",  
        photoFile);  
    takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);  
    startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);  
    getLocation();  
}
```

Note\* : untuk menggunakan GeoCoder dibutuhkan API :  
`implementation("com.google.android.gms:play-services-location:21.0.1")`

```

if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
    File imgFile = new File(currentPhotoPath);
    if (imgFile.exists()) {
        imagePaths.add(currentPhotoPath);
        String timeStamp = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date());
        imageTimestamps.add(timeStamp);
        imageAdapter.notifyDataSetChanged();
        saveImagePaths();
        saveImageLocations();
        saveImageTimestamps();
    }
}

```

Timestamp dalam kode diimplementasikan untuk mencatat waktu pengambilan gambar dan menampilkannya dalam interface apk. Saat gambar berhasil diambil, timestamp saat itu dibuat menggunakan SimpleDateFormat dan ditambahkan ke daftar imageTimestamps. Daftar ini kemudian disimpan ke dalam SharedPreferences (agar data timestamp tetap ada meskipun aplikasi ditutup dan dibuka Kembali).

Kode ini menyimpan daftar timestamp gambar ke dalam SharedPreferences. imageTimestamps dikonversi menjadi HashSet untuk memastikan tidak ada duplikasi sebelum disimpan. Kemudian, HashSet tersebut disimpan menggunakan kunci IMAGE\_TIMESTAMPS\_KEY, dan perubahan diterapkan dengan editor.apply() untuk memastikan bahwa timestamp gambar tersimpan secara aman di penyimpanan internal aplikasi.

```

private void saveImageTimestamps() {
    SharedPreferences sharedPreferences = getSharedPreferences("MyApp", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    Set<String> set = new HashSet<>(imageTimestamps);
    editor.putStringSet(IMAGE_TIMESTAMPS_KEY, set);
    editor.apply();
}

```

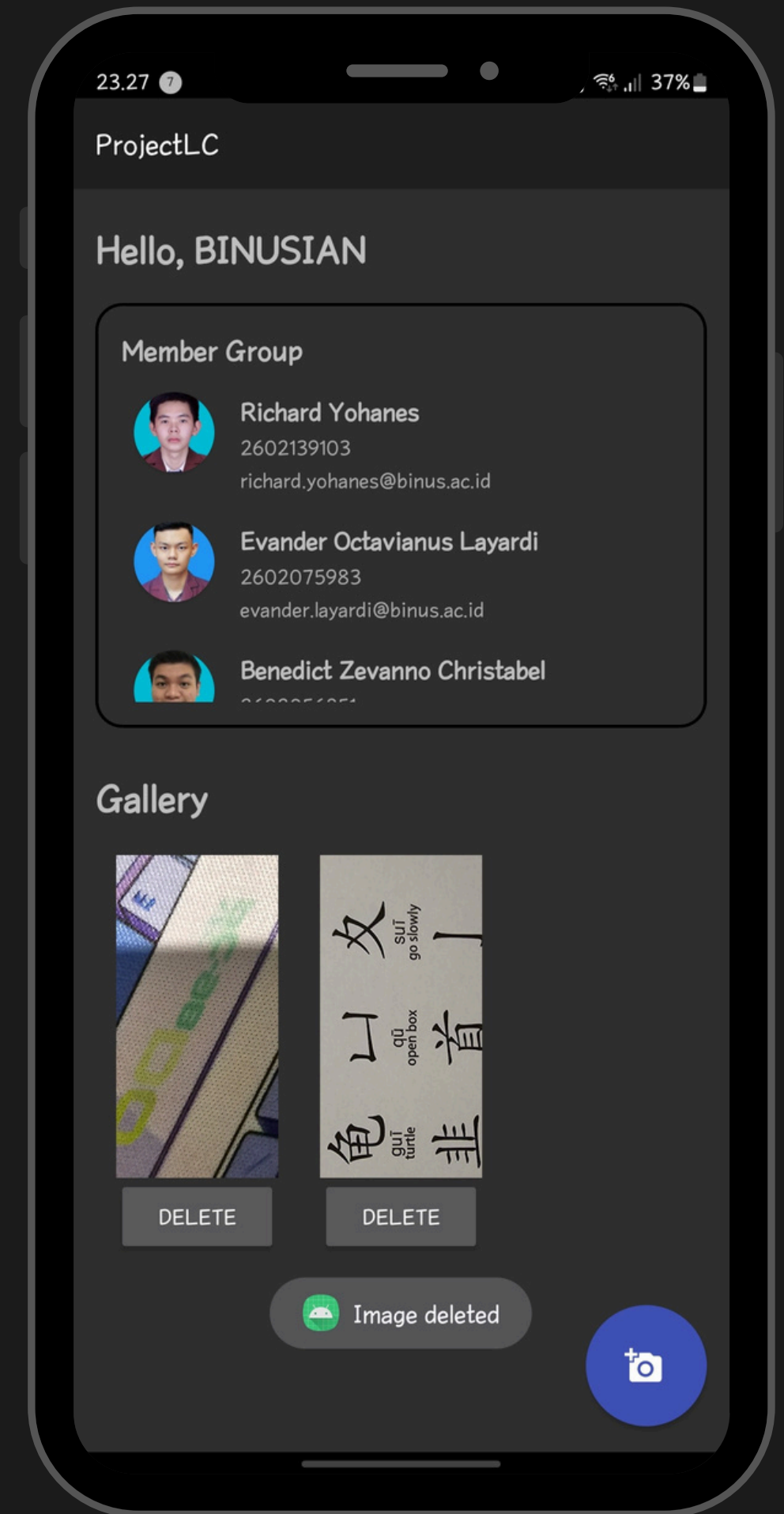
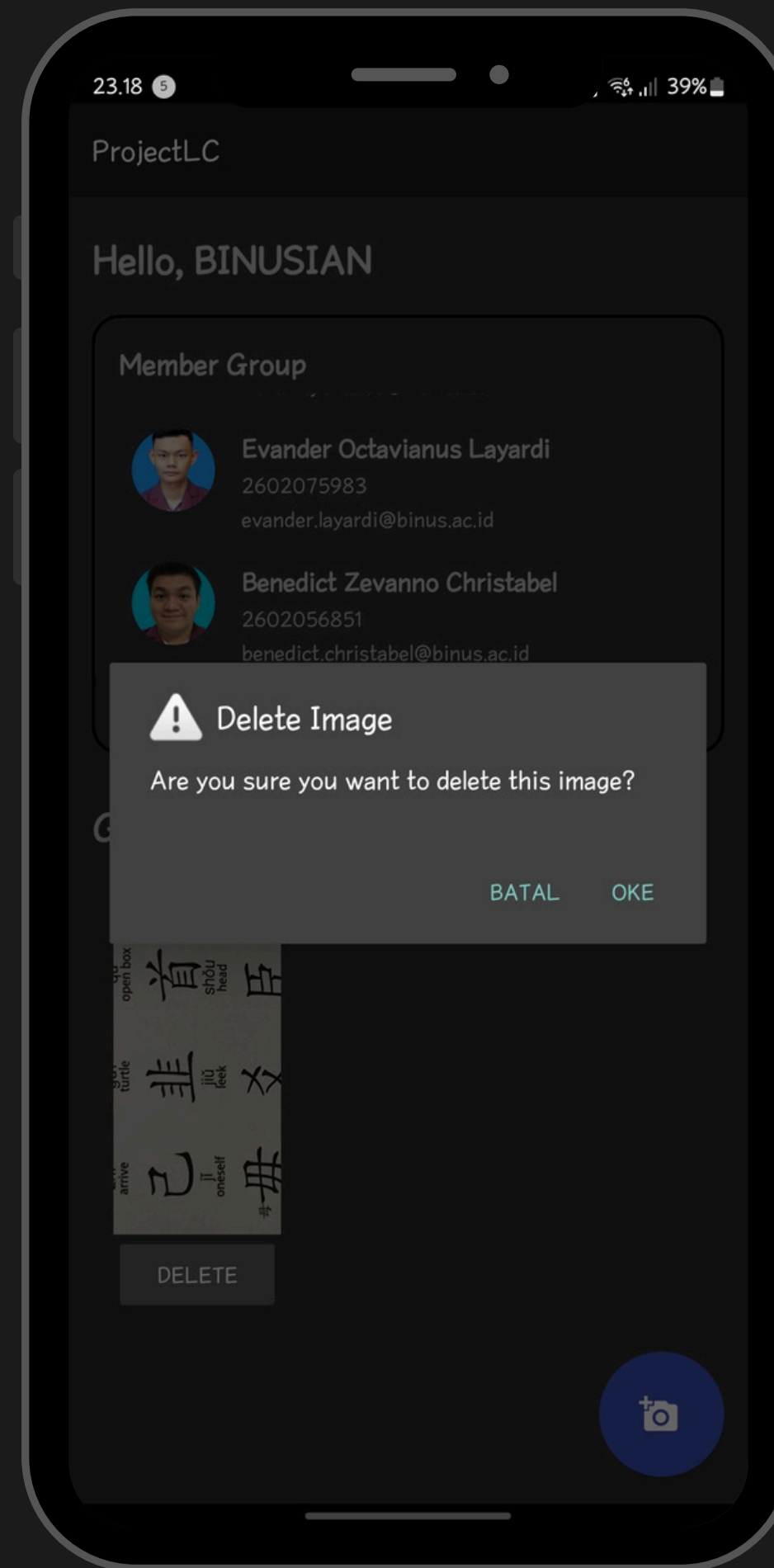
**IMPLEMENTASI CODE PADA PROGRAM**

# Timestamp.

# Delete Gambar

Aplikasi ini juga dapat menghapus gambar yang telah diambil dan disimpan pada memori internal dengan meng-klik tombol *delete* yang nantinya akan ada confirmation untuk men-delete gambar.

## FITUR APLIKASI



## ImageAdapter



```
deleteButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        new AlertDialog.Builder(mContext)  
            .setTitle("Delete Image")  
            .setMessage("Are you sure you want to delete this image?")  
            .setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog, int which) {  
                    // Remove item from list and storage  
                    ((MainActivity) mContext).removeImagePath(position);  
                    ((MainActivity) mContext).removeImageLocation(position);  
                    ((MainActivity) mContext).removeImageTimestamp(position);  
                    notifyDataSetChanged();  
  
                    Toast.makeText(mContext, text: "Image deleted", Toast.LENGTH_SHORT).show();  
                }  
            })  
            .setNegativeButton(android.R.string.no, null)  
            .show();  
    }  
});
```

## Delete Gambar Code

IMPLEMENTASI CODE PADA PROGRAM

Program delete gambar pada aplikasi terdapat di kelas **ImageAdapter** pada metode **getView**. Bagian kode ini mengatur perilaku tombol delete pada setiap item gambar yang ditampilkan dalam grid view. Ketika tombol delete ditekan, sebuah dialog konfirmasi akan muncul menanyakan apakah user akan menghapus gambar tersebut. Jika pengguna memilih "**yes**", maka item gambar akan dihapus dari daftar **imagePaths**, **imageLocations** dan **imageTimestamps**, serta pembaharuan daftar ditampilkan dengan memanggil fungsi **notifyDataSetChanged()**. Metode **removeImagePath**, **removeImageLocation** dan **removeImageTimestamp** di **MainActivity** digunakan untuk menghapus informasi terkait gambar dari penyimpanan. Setelah itu, pesan Toast akan muncul memberitahu pengguna bahwa gambar telah dihapus.



# Link Demo Aplikasi

[https://drive.google.com/file/d/1jDUnWk-X60e\\_Lfhi4lNyVbl\\_1NLcs31Z/view?usp=drive\\_link](https://drive.google.com/file/d/1jDUnWk-X60e_Lfhi4lNyVbl_1NLcs31Z/view?usp=drive_link)



\*\*\*[Link Keseluruhan Drive:  
https://drive.google.com/drive/folders/13\\_OzYcWTgOeRl4xt7RFnre51TDOUObth?usp=sharing](https://drive.google.com/drive/folders/13_OzYcWTgOeRl4xt7RFnre51TDOUObth?usp=sharing)

Project Aplikasi  
Berbasis Camera  
& Location

# Link APK Android

[https://drive.google.com/file/d/1\\_SCv2uoYQaHkIXsiOQnj\\_g29EKv-r1uL/view?usp=drive\\_link](https://drive.google.com/file/d/1_SCv2uoYQaHkIXsiOQnj_g29EKv-r1uL/view?usp=drive_link)



\*\*\*[Link Keseluruhan Drive:  
https://drive.google.com/drive/folders/13\\_OzYcWTgOeRl4xt7RFnre51TDOUObth?usp=sharing](https://drive.google.com/drive/folders/13_OzYcWTgOeRl4xt7RFnre51TDOUObth?usp=sharing)

Project Aplikasi  
Berbasis Camera  
& Location

Project Aplikasi  
Berbasis Camera  
& Location

KELOMPOK 2

ขอบคุณ Terima Kasih  
СПАСИБО Hatur Nuhun Matur Nuwun  
Danke  
תודה Thank You شكراً لك  
ありがとう 谢谢 감사합니다  
धन्यवाद