Nama : Evander Octavainus Layardi

NIM : 2602075983

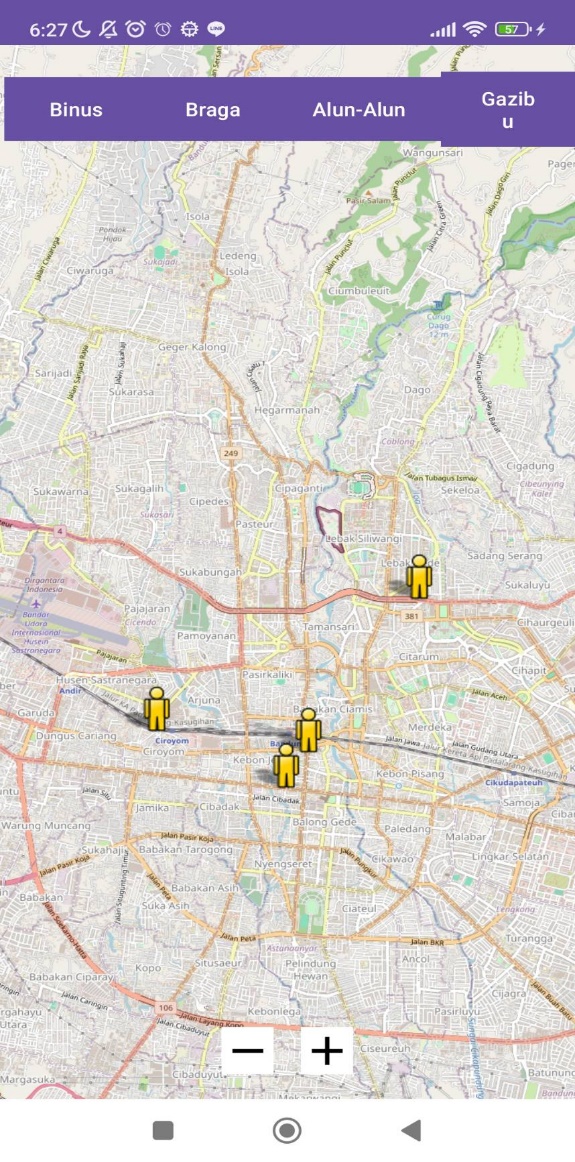
**UAS Mobile Programming**

**Berikut Link Google drive yang memuat Soruce Code aplikasi untuk Nomor 1 dan 3 + video simulasi** : <https://drive.google.com/drive/folders/1xHM_PVlltGsL776TZcn23isgoT2j2X3W?usp=drive_link>

1. Berikut adalah Output Porgram :

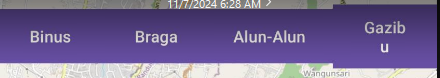
Sumber : <https://www.youtube.com/watch?v=xoFtgcOoO1I>

* Pertama tama adalah output yang menampilkan Point Of interest lokasi pada tabel yaitu ( Braga, alun-alun, Binus dan Lapangan Gasibu ) :

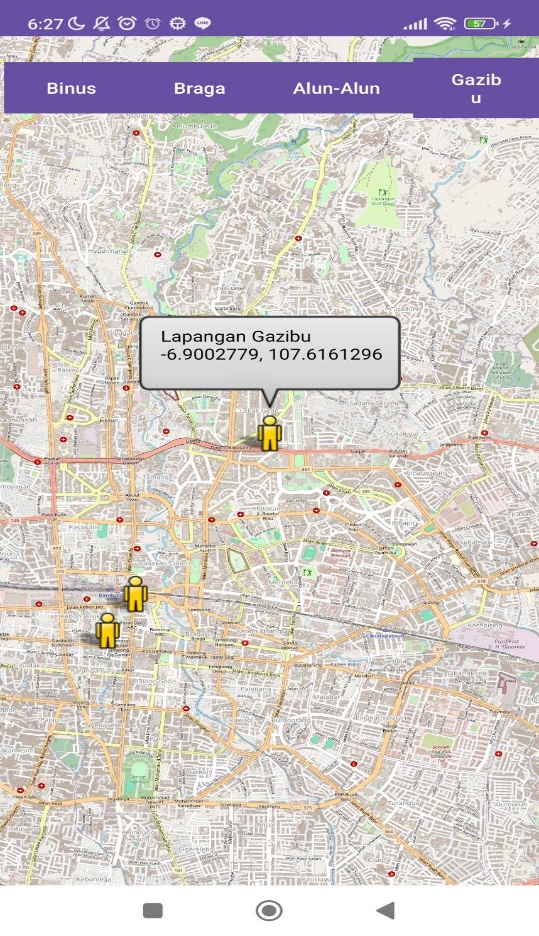


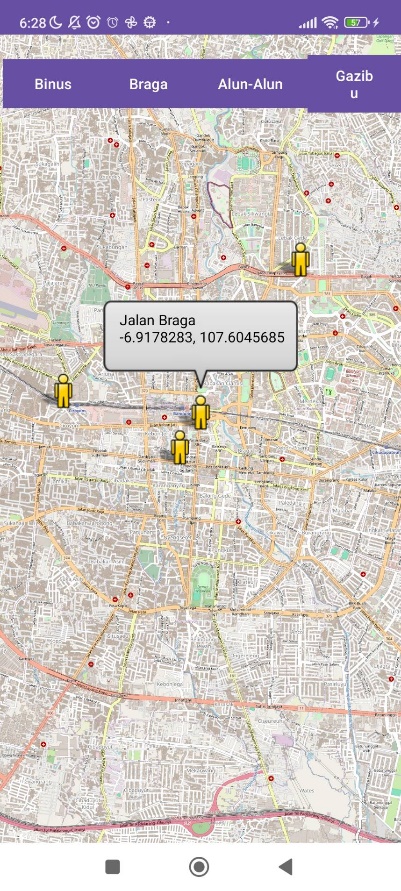
Dimana dalam output, simbol orang berwana kuning menunjukkan setiap lokasi pada tabel berdasarkan koordinat ( latitude, longtitude ).

* Kemudian dalam aplikasi juga mendukung recenter view dengan menggunakan tombol yang terletak pada bagian atas aplikasi :



Ketika Tombol tersebut dipencet untuk Binus, Braga , alun0alun dan Lapangan Gazibu, maka dalam aplikasi akan otomatis ter arah menjadi Recenter view untuk maisng masing lokasi. Berikut Gambar nya :



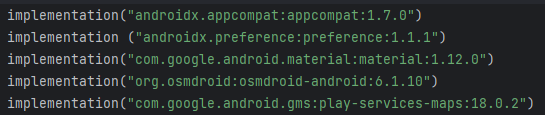


* Dalam aplikasi juga mendukung fitur zoom dan perkecil tampilan spesifik lokasi :



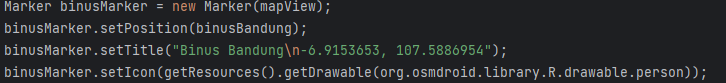
Nah dalam aplikasi terdapat simbol + untuk zoom in dan – untuk perkecil tampilan untuk lokasi.

Penjelasan Nomor 1 : jadi dalam program aplikasi ini saya menggunakan beberapa API yatu :



* Jadi untuk mendapatkan titik koordinat lokasi, saya menggunakan osmdrid versi : 6.1.10
* Kemudian untuk mendapatkan interface tampilan map saya menggnakan gms :play-service-maps dengan versi 18.0.2
* Kemudian kita perlu preference , saya menggunakan versi 1.1.1 untuk inisialisasi OSMDRoid konfigurasi pada main activity.

Berikut tahap implementasi dalam program terkiat OSMDrid ( misal untuk lokasi Binus ) :



1. A) Apa yang dimasud Dengan Thread ?

Sumber :

* <https://medium.com/@gungtia/java-thread-61b97e8dae88>
* <https://www.tutorialspoint.com/java/java_thread_synchronization.htm>

Jawab :

Thread dalam Java adalah unit eksekusi kecil yang bisa membuat program melakukan beberapa tugas secara bersamaan, yang dikenal sebagai (pemrosesan paralel / bersamaan atau saling hubung), Ini berarti program dapat lebih efisien dan responsif. Setiap program Java memiliki setidaknya satu thread utama yang disebut "main thread", yang merupakan thread pertama yang dijalankan saat program dimulai. Dengan menggunakan thread tambahan, program dapat menjalankan tugas tugas lain secara paralel, meningkatkan kinerja dan kemampuan multitaskingnya. Namun dalam beberapa kasus, perlu memperhatikan untuk mengelola thread dengan hati-hati agar tidak terjadi masalah seperti benturan data atau kebuntuan (deadlock).

b) Livelock dan Deadlock merupakan sebuah kondisi ketika memprogram menggunakan thread. Jelaskan kedua kondisi tersebut dan berikan contohnya !

Jawab :

**Livelock** terjadi ketika dua thread terus-menerus saling memberi jalan, tetapi tidak ada yang bisa maju karena mereka terus bergerak dengan cara yang sama ( buntu / terkunci ).

Contoh : misalnya dalam sebuah sistem di mana dua thread mencoba memproses transaksi bank. Setiap transaksi memerlukan validasi dari thread lain sebelum dapat diselesaikan, jika kedua thread terus-menerus memvalidasi transaksi masing-masing pada waktu yang bersamaan, maka hal itu mungkin terus memberi jalan satu sama lain tanpa pernah menyelesaikan transaksi tersebut.

Misalnya, dua thread Thread A dan Thread B bekerja dengan transaksi :

* Thread A mencoba memvalidasi transaksi dan meminta Thread B untuk memvalidasi.
* Thread B mencoba memvalidasi transaksi dan meminta Thread A untuk memvalidasi.

Jika kedua thread selalu memprioritaskan validasi dari thread lain, mereka akan terus berada dalam siklus tanpa akhir, memeriksa dan memvalidasi satu sama lain, tetapi tidak pernah menyelesaikan transaksi tersebut ( looping sehingga terjadi penumpukan ).

**Deadlock** terjadi ketika dua atau lebih thread saling menunggu sumber daya yang dimiliki oleh thread lain, sehingga tidak ada yang bisa melanjutkan eksekusi.

Contoh : ketika Thread A telah mengunci sumber daya A dan menunggu untuk mengunci sumber daya B yang dipegang oleh Thread B, sedangkan Thread B telah mengunci sumber daya B dan menunggu untuk mengunci sumber daya A yang dipegang oleh Thread A. Dalam hal ini, kedua thread saling menahan sumber daya yang diperlukan oleh yang lain untuk melanjutkan, sehingga sistem tidak bisa bergerak maju dan terjebak dalam keadaan menunggu tanpa batas.

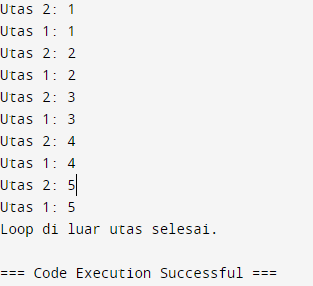
Kesimpulan perbedaan : dalam livelock, thread terus berubah posisi untuk mencoba menyelesaikan masalah tetapi tidak pernah berhasil, sementara dalam deadlock, thread tidak melakukan apapun karena masing-masing menunggu thread lain untuk melepaskan sumber daya yang dibutuhkan.

2C) Implementasikan sebuah loop sederhana menggunakan synchronous thread dalam bahasa Java. Sertakan screenshot hasil eksekusinya!

Source code :

|  |
| --- |
| public class SynchronousThread{  public static void main(String[] args) {  Thread thread1 = new Thread(new Runnable() {  @Override  public void run() {  for (int i = 1; i <= 5; i++) {  System.out.println("Utas 1: " + i);  try {  Thread.sleep(1000);  } catch (InterruptedException e) {  e.printStackTrace();  }  }  }  });  Thread thread2 = new Thread(new Runnable() {  @Override  public void run() {  for (int i = 1; i <= 5; i++) {  System.out.println("Utas 2: " + i);  try {  Thread.sleep(1000);  } catch (InterruptedException e) {  e.printStackTrace();  }  }  }  });  thread1.start();  thread2.start();  try {  thread1.join();  thread2.join();  } catch (InterruptedException e) {  e.printStackTrace();  }  System.out.println("Loop di luar utas selesai.");  }  } |

Output :



Penjelasan :

synchronous threading berarti dua atau lebih utas (threads) dieksekusi secara berurutan, di mana satu utas menunggu hingga utas sebelumnya selesai sebelum melanjutkan. Pada program ini, terdapat dua utas yang masing-masing memiliki tugas berulang sederhana dengan )jeda waktu 1 detik antar iterasi).

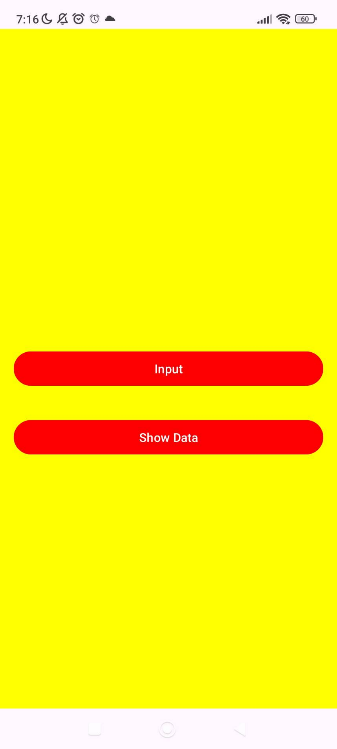
Setiap utas (thread) digunakan untuk mengeksekusi tugas terpisah (dalam hal ini, perulangan sederhana) secara sinkronisasi. Jadid dengan program ini bisa untuk melakukan beberapa tugas secara paralel tetapi dengan koordinasi yang terkontrol, yang berguna dalam situasi di mana urutan eksekusi atau hasil dari satu utas harus diketahui atau diproses sebelum melanjutkan utas berikutnya. Dengan menggunakan metode thread.join(), program menunggu kedua utas selesai sebelum mencetak pesan bahwa semua perulangan di luar utas telah selesai dieksekusi.

1. (LO 4 : 25 poin) Buatlah sebuah aplikasi Android sederhana yang menggunakan SQLite. Aplikasi tersebut mampu merekam data mahasiswa beserta mata kuliah yang diambil. Aplikasi yang diminta wajib menerapkan hal berikut :
2. (2 poin) Terdapat data mahasiswa pada tabel diatas.
3. (12 poin) Dapat menghapus dan menambahkan data mahasiswa.
4. (11 poin) Menampilkan semua data mahasiswa menggunakan ViewGroup seperti Recycler View atau Grid View.

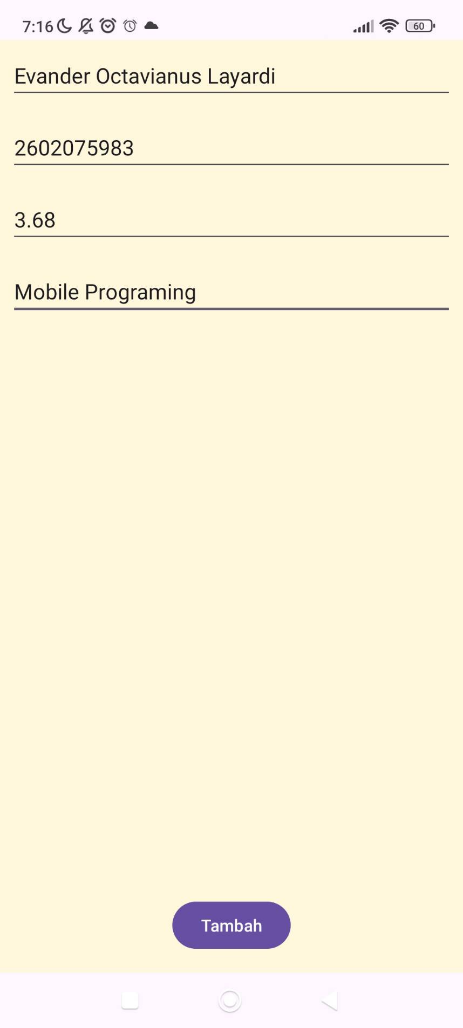
Sumber : <https://youtu.be/4k1ZMpO9Zn0?si=q2xgG2bT5fFV1_OK>

Output Project :

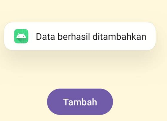
* Dalam aplikasi saya membuat Menu untuk input data Mahasiswa seperti Nama, NIM, IPK dan Mata kuliah, dan saya juga mmebuat menu untuk Show data / menampilkan data yang tersimpan saat sudah diinput dalam database SQLite ( halaman tersebut bernama MenuActivity ) :



* Ketika user menekan tombol Input maka akan menampilkan / beralih ke halaman baru (MainActivity) untuk melakukan input data :

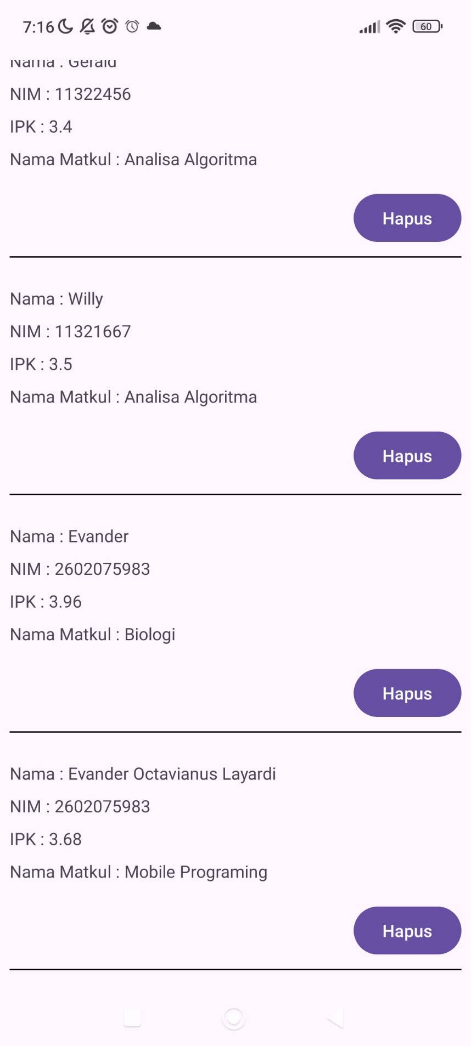


Proses input data dapat dilakukan secara berulang kali dengan user menekan tombol Tambah, ketika menekan tombol tersebut akan memunculkan notifikasi :

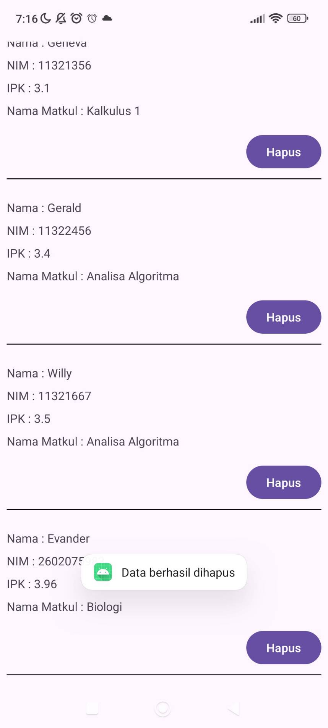


Artinya data sudah berhasil disimpan kedalam database SQLite.

* Kemudian kita akan beralih ke menu Show data, dimana data yang sebelumnya diinput user dan berhasil disimpan dalam databse pada menu Input, maka dokumentasi data akan ditampilkan dalam menu ini dengan berlih ke halaman baru bernama MenuActivity untuk menampilkan data :

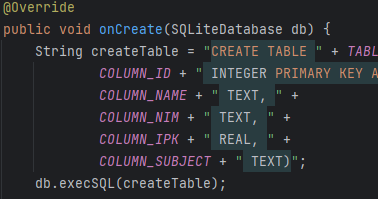


Dapat Kita lihat data yang sebelumnya diinput sudah berada pada tampilan data menu, kemudian dalam show data ini juga terdapat fitur delete untuk melakukan delete data secara permanen dari database SQLite sehingga menghindari penumpukan data, berikut simulasinya :

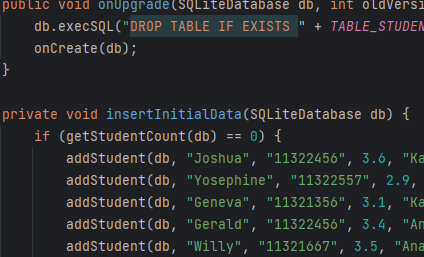


Data dengan Nama : Evander, NIM : 2602075983, IPK : 3.68, Mata Kuliah : Mobile Programming sebelumnya yang ditampilkan, ketika user pencet tombol delete maka akan menghapus dari tampilan dengan ditandai notifikasi pesan sistem aplikasi “Data berhasl dihapus”.

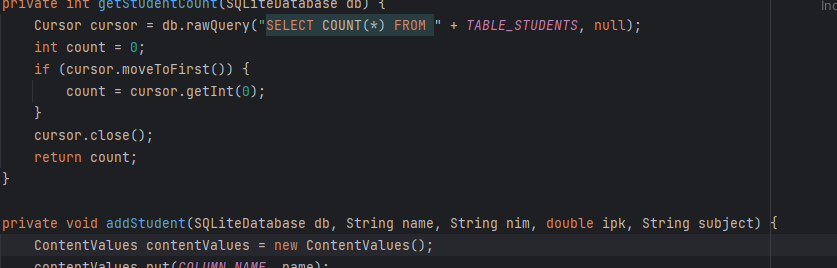
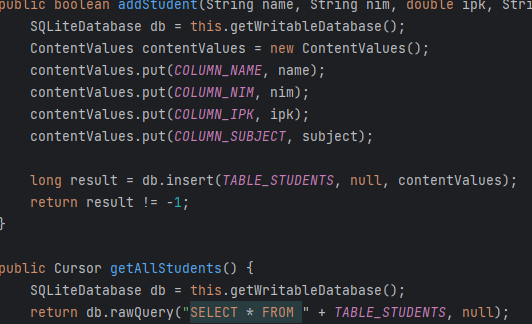
* Implementasi Code terkait aplikais ini :



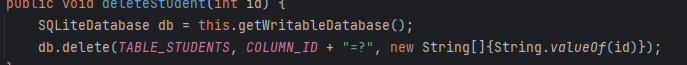
Dalam onCreate, tabel students dibuat dengan kolom-kolom seperti id (kunci utama), name (nama mahasiswa), nim (nomor induk mahasiswa), ipk (indeks prestasi kumulatif), dan subject (mata kuliah).



Metode insertInitialData dipanggil saat database pertama kali dibuat untuk menyisipkan data awal jika tabel kosong. Ini dilakukan dengan memanggil addStudent, yang menggunakan ContentValues untuk menambahkan entri baru ke tabel.



Fungsi addStudent digunakan untuk menambahkan mahasiswa baru ke database, dengan memasukkan nilai-nilai nama, nim, ipk, dan mata kuliah ke dalam ContentValues dan melakukan operasi insert ke tabel students. Sementara Fungsi getAllStudents mengembalikan sebuah Cursor yang berisi semua entri dari tabel "students", memungkinkan aplikasi untuk mengambil dan menampilkan semua data mahasiswa.



Metode deleteStudent menghapus entri mahasiswa dari tabel berdasarkan id yang diberikan.

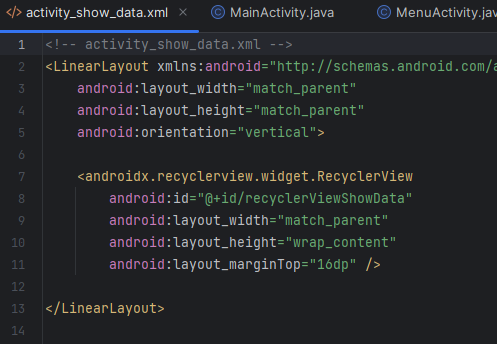
* Implementasi SQLite dalam penambahan dan delete data mahasiswa :

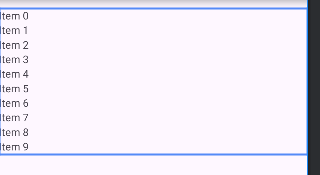
Pada Class MainActivity dalam aplikasi Android ini, implementasi SQLite digunakan pada **penambahan dan penghapusan data mahasiswa**. Ketika pengguna menekan tombol "Tambah", data yang dimasukkan seperti nama, NIM, IPK, dan mata kuliah disimpan ke dalam database SQLite menggunakan metode addStudent() dari DatabaseHelper, sementara itu, penghapusan data dilakukan ketika pengguna memilih untuk menghapus entri mahasiswa, dengan memanggil deleteStudent() berdasarkan ID mahasiswa. Ini memastikan pengelolaan data yang efisien dan responsif di dalam aplikasi.

* Implementasi SQLite dalam Menampilkan data mahasiswa :

Class ShowDataActivity dalam aplikasi ini menggunakan SQLite untuk menampilkan daftar mahasiswa dari database lokal. Data diambil dengan metode getAllStudents() dari DatabaseHelper, kemudian ditampilkan dalam RecyclerView menggunakan StudentAdapter. user dapat menghapus data langsung dari tampilan apk dengan mengklik tombol hapus pada setiap item ( elemen ) , yang otomatis merubah tampilan data mahasiswa.

Implementasi menampilkan data mahasiswa menggunakan Recycle View pada layout activity\_show\_data.xml :





1. Jelaskan dengan detil aplikasi yang menjadi tugas project kalian. Detil yang diminta adalah sebagai berikut :
2. Tujuan atau manfaat aplikasi.

Jawab :

Dalam aplikasi yang dibuat saat project yang bernama ProjectLC ( LoCam ) : 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) serta dalam aplikasi juga bisam menampilkan profile dari masing masing pengembang aplikasi, ketika format email ditekan akan mengarahkan ke halaman gmail.

**Manfaat** : yaitu user dapat mengambil foto di suatu tempat disertai detail lokasi dan waktu pengambilan yang berguna saat sewaktu waktu ingin mengenang atau mendokumentasi hasil foto yang diambil secara rinci (user dapat mengingat kapan dan dimana foto itu diambil ), kemudian foto juga akan disimpan dalam memori internal sehingga selama foto tidak dihapus, data foto akan terjaga aman di memori.

1. Karakteristik pengguna aplikasi .

Jadi karakterisik user nya yaitu lebih kepada seseorang yang senang melakukan selfie baik pribadi atau keluarga jika berkunjung ke suatu tempat tertentu, kemudian pengguna selanjutnya yaitu orang yang suka berpergian untuk travlling ke kota / wisata tertentu, karena seorang traveller tentu kebanyakan memerlukan kenangan pribadi selama liburan dengan salah satunya dokumentasi pengambilan foto dengan tempat wisata, faktor tempat dan waktu untuk user travelling ini menjadi penting karena menjadi bukti dokumentasi dan kenangan user dalam track record travelling nya atau pengalaman aktivitas selama liburan yang diekspresikan kedalam bentuk foto.

Bisa juga dimanfaatkan oleh pihak agen polisi yang melakukan olah TKP dengan dokumentasi bukti foto yang akurat terkait tempat dan waktu kejadian yang diambil pada saat itu, sehingga fitur lokasi dan waktu pengambbilan foto berdsarkan gambar diambil sangat dibutuhkan untuk dkumentasi bukti yang akurat.

1. Salah satu implementasi dari terminologi teknis berikut (boleh lebih) :

• Implementasi location-based services.

• Implementasi CRUD

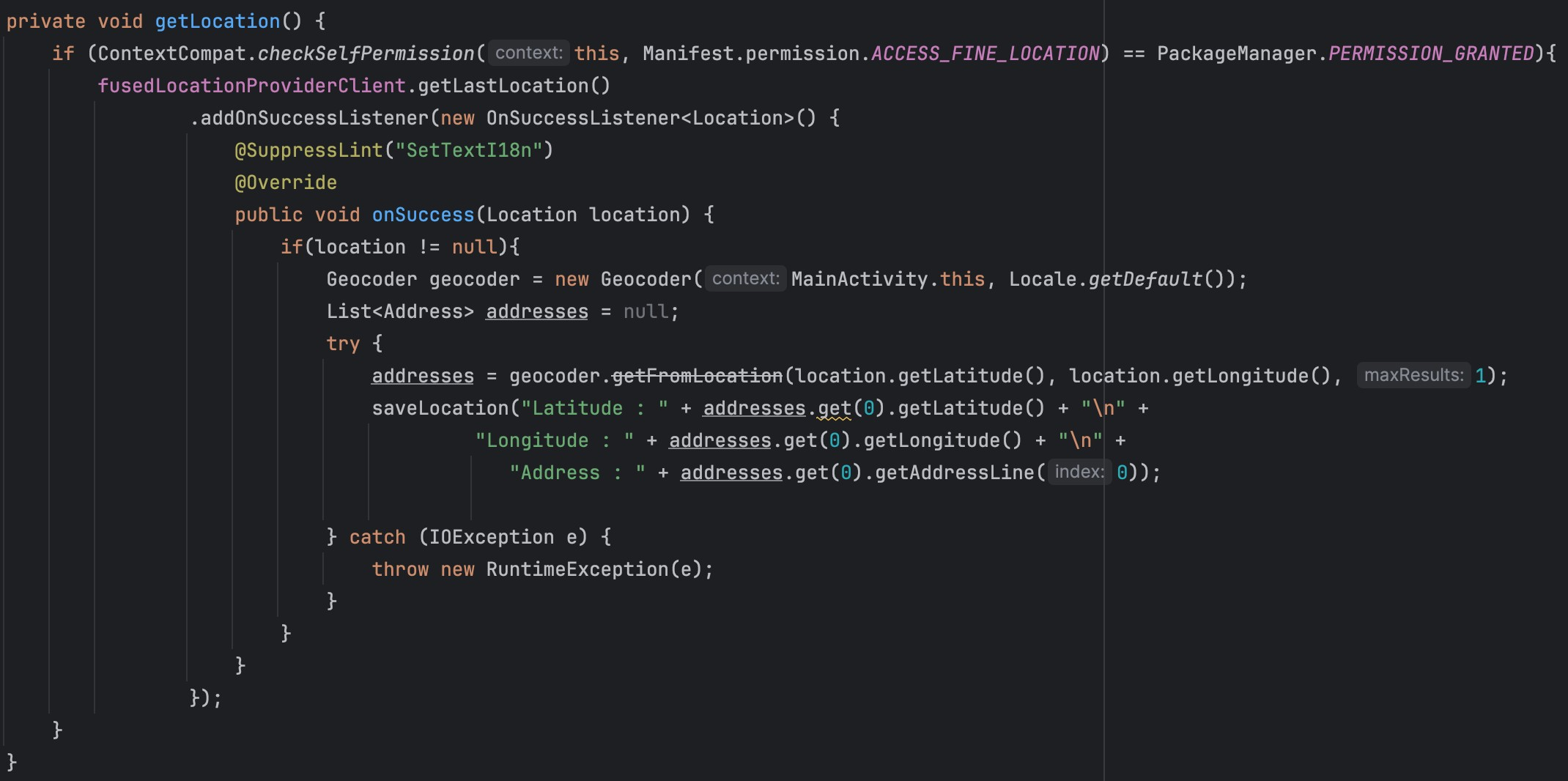
• Implementasi Thread (bukan thread bawaan dari Android)

• Penggunaan REST API

Jawab :

Dalam aplikasi projek yang kami buat yaitu menampilkan lokasi ( berdasarkan koordinat ) dan waktu kami tergolong implementasi location based services yang menggunakan implementasi **geocoder.**

**Berikut implementasi potongan source code geocoder untuk menampilkan lokasi pada fungsi getlocation() :**



**Penjelasan :** Fungsi getLocation() dalam program ini adala bagian yang digunakan untuk mengimplementasikan geocoder untuk mendapatkan lokasi userberdasarkan koordinat GPS yang diperoleh dari layanan lokasi terakhir yang diberikan oleh fusedLocationProviderClient ( untuk mendpatkan lokasi terakhir dari user berdasarkan titik koordinat ) .

Nah flow cara kerja getlocation ini adalah pertama tama meminta izin user untuk akses lokasi pada aplikasi untuk mendapatkan lokasi, Setelah mendapatkan lokasi, geocoder digunakan untuk mengonversi koordinat tersebut menjadi alamat teks yang dapat dibaca manusia seperti format alamat pada umumnya yang terdiri dari jalan, kecamatan dan nomor. Informasi lokasi ini kemudian disimpan menggunakan method saveLocation(), yang menambahkan detail lokasi ke dalam daftar imageLocations ( tujuan implementasi geocoder -> user dapat untuk melihat informasi lokasi aktual dalam format yang mudah dipahami langsung dari aplikasi Android Studio )

d) Fungsionalitas atau fitur-fitur pada aplikasi.

- Aplikasi dapat memungkinkan user untuk mengambil foto : yaitu dengan izin akses kamera

- dapat menyimpan foto ke memori internal perangkat : ketika foto diambil, maka foto akan disimpan dalam aplikasi andorid melalui penyimpanan internal.

- Dapat menampilkan detail lokasi berdasarkan koordinat dan waktu pengambilan foto tersebut diambil secara real-time saat pengambilan foto di suatu tempat.

- dapat menghapus gambar yang telah diambil, dengan menekan tombol delete di bawah foto.

- Foto yang sudah di delete akan secara permanen terhapus dari aplikasi karena adanya dukungan clean dan load image dalam program.

- menampilkan detail data diri dari pengembang aplikasi dalam halaman utama aplikasi sebagai interface UI tambahan.

- Format foto yang ditampilkan dalam bentuk Gridview adalah UI untuk menyimpan daftar foto secara berurutan dan tertata / tersusun rapi dan banyak dalam suatu baris, jadi dalam 1 halaman ( bagian foto disimpan dalam aplikasi ) dapat menampilkan banyak foto dengan dukungan layout Gridview.

Jadi, fitur fitur seperti mengambil foto dengan kamera, mencatat lokasi dan waktu, serta menyimpan data foto dalam format yang terstruktur secara langsung mendukung tujuan aplikasi untuk membantu pengguna mengelola dan mengabadikan momen user. Dengan adanya fitur ini, user umum dan travelers dapat lebih mudah mengingat dan mengatur koleksi foto mereka, sesuai dengan kebutuhan untuk mengelola foto berdasarkan lokasi dan waktu pengambilan.

Selain itu, fitur-fitur seperti menampilkan foto dalam GridView dengan informasi lengkap, dan kemampuan untuk menghapus foto dengan metadata terkait, membantu pengguna untuk secara efisien mengorganisir dan mengelola foto yang diambil. Hal ini tidak hanya memenuhi kebutuhan pengguna umum yang ingin mengelola koleksi foto pribadi mereka, tetapi juga memberikan nilai tambah bagi travelers yang menginginkan dokumentasi perjalanan yang terperinci. Profesional juga dapat memanfaatkan fitur-fitur ini untuk keperluan kerja mereka, seperti dokumentasi visual yang terstruktur dan mudah diakses. Dengan ini, fungsionalitas yang dibawakan dalam aplikasi ini secara langsung mendukung karakteristik useryang beragam ( multi fungsi dalam bidang tertentu yang hubungannnya erat dengan dokumentasi ) , serta tujuan untuk memfasilitasi pengelolaan dan pengingatan momen melalui foto dengan informasi lokasi dan waktu yang terintegrasi.