

Nama : Liskania Aprilia  
Kelas : TI.22.A4  
NIM : 312210383  
Mata Kuliah : Pemrograman Mobile 1

---

## LAPORAN HASIL UJIAN AKHIR SEMESTER (TYRELL APPS)

### 1. Laporan Aktivitas Activity: MainHello

Kelas: MainHello

- Tujuan: Menampilkan antarmuka pengguna melalui layout `activity\_hello`.
- Metode Utama:
  - `onCreate`: Menginisialisasi dan menetapkan tata letak (layout) dengan menggunakan `setContentView`.

Fungsi Utama:

Menampilkan Layout:

- Kode ini bertujuan untuk menampilkan layout dengan nama `activity\_hello` ke pengguna.

Catatan:

- Kode ini adalah contoh sederhana dari sebuah aktivitas di Android yang menggunakan layout untuk menampilkan antarmuka pengguna.
- Aktivitas ini tidak memiliki interaksi pengguna tambahan, karena hanya menampilkan layout yang telah ditentukan (`activity\_hello`).
- Sangat penting untuk memastikan bahwa file layout yang disebutkan dalam `setContentView` (`activity\_hello`) tersedia dan telah dikonfigurasi sesuai kebutuhan aplikasi.

Pembaruan dan Rekomendasi:

- Menambahkan logika bisnis atau interaksi pengguna sesuai kebutuhan aplikasi.
- Melakukan penanganan kesalahan dan validasi input jika diperlukan.
- Menambahkan komentar untuk menjelaskan bagian-bagian kunci dalam metode `onCreate` agar memudahkan pemahaman dan pemeliharaan kode.
- Memastikan bahwa layout yang digunakan (`activity\_hello`) sesuai dengan kebutuhan desain dan UI aplikasi.

Pesan Tambahan:

- Aktivitas ini mungkin merupakan bagian dari struktur aplikasi yang lebih besar, dan perlu dipastikan bahwa navigasi dan interaksi antar aktivitas diatur dengan benar.

### 2. Laporan Aktivitas Activity: MainCount

Kelas: MainCount

- Tujuan: Mengimplementasikan fungsionalitas penjumlahan dan tampilan pesan Toast dalam aplikasi Android.

- Metode Utama:
  - ``onCreate``: Menginisialisasi tata letak (layout) dan komponen-komponen yang diperlukan.
  - ``countUp``: Menaikkan nilai hitungan dan menampilkan pada ``TextView``.
  - ``showToast``: Menampilkan pesan Toast saat tombol diklik.

Fungsi Utama:

- Inisialisasi:
  - ``onCreate`` digunakan untuk menginisialisasi aktivitas dan menetapkan layout (``activity_count``) serta mendapatkan referensi ke ``TextView`` (``show_count``).
- Penjumlahan dan Tampilan:
  - ``countUp`` digunakan untuk menambah nilai hitungan (``currentCount``) dan menampilkan pada ``TextView`` (``show_count``).
- Pesan Toast:
  - ``showToast`` menampilkan pesan Toast dengan pesan yang diambil dari sumber daya string (``R.string.toast_message``).

Catatan:

- ``currentCount`` digunakan untuk menyimpan nilai hitungan saat ini.
- Fungsi ``countUp`` digunakan untuk meningkatkan nilai hitungan dan memperbarui tampilan pada ``TextView``.
- Fungsi ``showToast`` digunakan untuk menampilkan pesan Toast yang berasal dari sumber daya string.

Pembaruan dan Rekomendasi:

- Menambahkan logika bisnis tambahan sesuai kebutuhan aplikasi.
- Melakukan penanganan kesalahan atau validasi input jika diperlukan.
- Memperkaya antarmuka pengguna dengan elemen-elemen UI tambahan atau desain yang lebih kompleks.

Pesan Tambahan:

- Aktivitas ini dapat digunakan sebagai contoh sederhana untuk mengimplementasikan penjumlahan dan menampilkan pesan Toast dalam pengembangan aplikasi Android.

### 3. Laporan Aktivitas Activity: MainIceCold

Kelas: MainIceCold

- Tujuan: Menampilkan tampilan berita dari sumber daya layout ``activity_icecold``.
- Metode Utama:
  - ``onCreate``: Menginisialisasi aktivitas dan menetapkan tata letak (layout) yang diberikan oleh ``activity_icecold``.

Fungsi Utama:

- Inisialisasi:

- ``onCreate`` digunakan untuk menginisialisasi aktivitas dan menetapkan layout (``activity_icecold``) yang berisi tampilan berita.

Tampilan Berita:

- Aktivitas ini bertujuan untuk menampilkan tampilan berita yang diatur dalam layout ``activity_icecold``.

Pesan Tambahan:

- Aktivitas ini menyajikan contoh sederhana dari penggunaan layout untuk menampilkan tampilan berita di aplikasi Android.

Pembaruan dan Rekomendasi:

- Menambahkan komponen-komponen UI atau fungsionalitas tambahan sesuai kebutuhan aplikasi.
- Menggunakan komponen Android seperti ``TextView``, ``ImageView``, atau ``WebView`` untuk menampilkan konten berita yang lebih dinamis.
- Memperkaya tampilan dengan elemen desain yang sesuai dengan tema aplikasi.

#### **4. Laporan Aktivitas Alarm**

Kelas: `AlarmActivity`

- Tujuan: Membuat dan mengelola alarm menggunakan ``AlarmManager``.
- Metode Utama:
  - ``onCreate``: Menginisialisasi ``TimePicker`` dan ``AlarmManager``.
  - ``OnToggleClicked``: Mengelola fungsi alarm saat tombol toggle diubah.
- Fungsi Utama:
  - Setting Alarm:
    - Membuat objek ``Calendar`` untuk mendapatkan waktu saat ini dalam jam dan menit.
    - Menetapkan jam dan menit pada objek ``Calendar`` berdasarkan ``TimePicker``.
    - Membuat objek ``Intent`` untuk memanggil ``AlarmReceiver`` saat alarm berbunyi.
    - Membuat objek ``PendingIntent`` untuk memanggil siaran tertunda saat alarm berbunyi.
    - Mengatur waktu alarm menggunakan ``AlarmManager.setRepeating``.
    - Jika waktu saat ini lebih besar dari waktu alarm yang diatur, penyesuaian waktu dilakukan untuk alarm keesokan harinya.
  - Alarm berbunyi terus-menerus sampai tombol toggle dimatikan.
  - Mematikan Alarm:
    - Membatalkan alarm menggunakan ``AlarmManager.cancel`` saat tombol toggle dimatikan.

Pesan Log:

- Menampilkan pesan toast "ALARM ON" saat tombol toggle dihidupkan.
- Menampilkan pesan toast "ALARM OFF" saat tombol toggle dimatikan.

Pembaruan dan Rekomendasi:

- Perlu memperhatikan bahwa kode saat ini memiliki alarm yang diatur untuk berbunyi setiap 10 detik (`alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000, pendingIntent)`). Jika niatnya adalah memicu alarm setiap hari pada waktu tertentu, waktu pengulangan perlu disesuaikan.
- Menambahkan komentar untuk menjelaskan bagian-bagian penting dalam metode `OnToggleClicked` agar memudahkan pemahaman dan pemeliharaan kode.
- Periksa keamanan aplikasi dan pastikan bahwa alarm hanya diatur untuk tugas yang diperlukan. Jangan lupa untuk mendapatkan izin yang diperlukan jika diperlukan oleh aplikasi.

Catatan:

- Pastikan kelas `AlarmReceiver` telah diimplementasikan dan dikonfigurasi dengan benar agar berfungsi sesuai harapan.

## 5. Laporan Aktivitas Activity: MainActivity (Fibonacci)

Kelas: MainActivity

- Tujuan: Menyediakan fungsionalitas untuk menampilkan dan mengelola bilangan Fibonacci dengan tampilan pengguna yang sesuai.
- Metode Utama:
  - `onCreate`: Menginisialisasi aktivitas dan menetapkan tata letak (layout) yang diberikan oleh `activity_toast`.
  - `updateCountDisplay`: Memperbarui tampilan pengguna dengan nilai terkini dari bilangan Fibonacci dan menyesuaikan warna teks berdasarkan aturan tertentu.
  - `showToast`: Menampilkan pesan Toast yang memberi tahu pengguna bahwa mereka berurusan dengan bilangan Fibonacci.
  - `countUp`: Menambahkan nilai bilangan Fibonacci dan memperbarui tampilan pengguna sesuai.
  - `back1`: Mengembalikan pengaturan ke awal dengan mengatur ulang nilai-nilai untuk menghitung bilangan Fibonacci.

Fungsi Utama:

- Inisialisasi:
  - `onCreate` digunakan untuk menginisialisasi aktivitas, menetapkan layout (`activity_toast`), dan mengakses elemen antarmuka pengguna seperti `TextView` dan `EditText`.
- Perhitungan Bilangan Fibonacci:
  - `countUp` digunakan untuk menghitung dan menampilkan bilangan Fibonacci selanjutnya berdasarkan input pengguna.
- Pengaturan Ulang:
  - `back1` memungkinkan pengguna mengatur ulang hitungan bilangan Fibonacci ke kondisi awal.

Antarmuka Pengguna:

- `TextView`: Menampilkan nilai bilangan Fibonacci dan berubah warna sesuai aturan tertentu.

- EditText: Memungkinkan pengguna memasukkan nilai maksimum untuk perhitungan bilangan Fibonacci.

Pesan Toast:

- Pesan Toast diperlihatkan saat pengguna menekan tombol untuk menyoroti penggunaan bilangan Fibonacci dalam aktivitas ini.

Catatan:

- Implementasi ini memberikan pengalaman interaktif kepada pengguna dengan memperbarui tampilan setiap kali bilangan Fibonacci bertambah.
- Warna teks berubah sesuai aturan tertentu untuk memberikan variasi visual.

Pembaruan dan Rekomendasi:

- Menambahkan kontrol UI tambahan seperti ProgressBar atau animasi untuk meningkatkan interaktivitas.
- Melakukan validasi input pengguna untuk memastikan bahwa input yang dimasukkan sesuai dan tidak menyebabkan kesalahan.
- Menyesuaikan desain tampilan untuk meningkatkan estetika antarmuka pengguna.

## 6. Laporan Kode: MainActivityOne

Deskripsi Kode:

Kode ini adalah implementasi dari kelas `MainActivityOne` dalam aplikasi Android. MainActivityOne bertindak sebagai aktivitas utama yang memungkinkan pengguna mengirim pesan ke SecondActivity dan menerima balasan dari aktivitas kedua tersebut.

Struktur Kode:

### 1. Variabel Kelas:

- `LOG\_TAG`: Variabel konstan yang digunakan untuk tag log.
- `EXTRA\_MESSAGE`: Variabel konstan yang berfungsi sebagai kunci untuk menyimpan pesan ekstra dalam Intent.
- `TEXT\_REQUEST`: Konstanta yang menentukan kode permintaan untuk startActivityResult.
- `mMessageEditText`: EditText untuk memasukkan pesan.
- `mReplyHeadTextView`: TextView untuk menampilkan header balasan.
- `mReplyTextView`: TextView untuk menampilkan balasan.

### 2. Metode onCreate:

- Menginisialisasi tampilan dan variabel kelas.

### 3. Metode launchSecondActivity:

- Dipanggil ketika tombol di MainActivityOne diklik.

- Membuat Intent untuk memulai SecondActivity dengan membawa pesan dari `mMessageEditText`.
- Memulai aktivitas kedua dengan `startActivityForResult` untuk mendapatkan hasil balasan.

#### 4. Metode onActivityResult:

- Dipanggil ketika aktivitas kedua selesai.
- Memeriksa apakah permintaan adalah `TEXT_REQUEST` dan hasilnya adalah `RESULT_OK`.
- Menampilkan header balasan dan menetapkan teks balasan ke `mReplyTextView`.

#### Catatan Tambahan:

- Log: Penggunaan `Log.d` untuk mencatat pesan ke konsol log.
- Intent Ekstra: Menggunakan intent ekstra untuk mengirim pesan antara aktivitas.
- `startActivityForResult`: Menggunakan `startActivityForResult` untuk memulai aktivitas kedua dan mendapatkan hasil balasan.

## 7. Laporan Kode: MainMaps

#### Deskripsi Kode:

Kode ini mengimplementasikan kelas `MainMaps` dalam aplikasi Android. `MainMaps` bertindak sebagai aktivitas yang menampilkan lokasi geografis pada peta menggunakan aplikasi peta default di perangkat.

#### Struktur Kode:

##### 1. Variabel Kelas:

- Tidak ada variabel kelas yang dideklarasikan secara eksplisit.

##### 2. Metode onCreate:

- Dipanggil saat aktivitas dibuat.
- Mengatur tata letak tampilan dan menyiapkan data lokasi geografis.
- Membuat Uri dengan data lokasi latitude dan longitude.
- Memanggil `showMap` untuk menampilkan peta dengan lokasi yang telah ditentukan.

##### 3. Metode showMap:

- Membuat Intent dengan tindakan `ACTION_VIEW`.
- Mengatur data intent ke Uri yang mewakili lokasi geografis.
- Memeriksa keberadaan aplikasi yang dapat menangani intent untuk menampilkan peta.
- Jika ada aplikasi yang dapat menangani intent, aktivitas baru dimulai untuk menampilkan lokasi pada peta.

#### Catatan Tambahan:

- Intent `ACTION_VIEW`: Digunakan untuk meminta sistem untuk menampilkan data.
- `Uri.parse`: Membuat objek Uri yang mewakili lokasi geografis.

## 8. Laporan Kode: MainFragment (Movie)

### Deskripsi Kode

Kode ini adalah implementasi dari aktivitas utama (`MainFragment`) dalam aplikasi Android yang menggunakan `ViewPager` dan `TabLayout` untuk menampilkan tiga fragmen yang berbeda: Fantasy, Comedy, dan Romance.

### Komponen Utama

#### 1. TabLayout (`tabLayout`):

- Digunakan untuk menampilkan tab yang mewakili setiap fragmen.
- Didefinisikan dalam layout dengan ID `tablayout`.

#### 2. ViewPager (`viewPager`):

- Digunakan untuk menampilkan dan mengelola fragmen.
- Didefinisikan dalam layout dengan ID `viewpager`.

#### 3. VPAdapter (`vpAdapter`):

- Kelas yang meng extends `FragmentPagerAdapter`.
- Digunakan untuk menghubungkan ViewPager dengan fragmen dan menangani navigasi antar fragmen.

#### 4. Fragment1, Fragment2, dan Fragment3:

- Fragmen yang masing-masing mewakili kategori Fantasy, Comedy, dan Romance.

### Langkah-langkah Utama

#### 1. Inisialisasi Komponen:

- Menginisialisasi `tabLayout` dan `viewPager` dari tata letak.

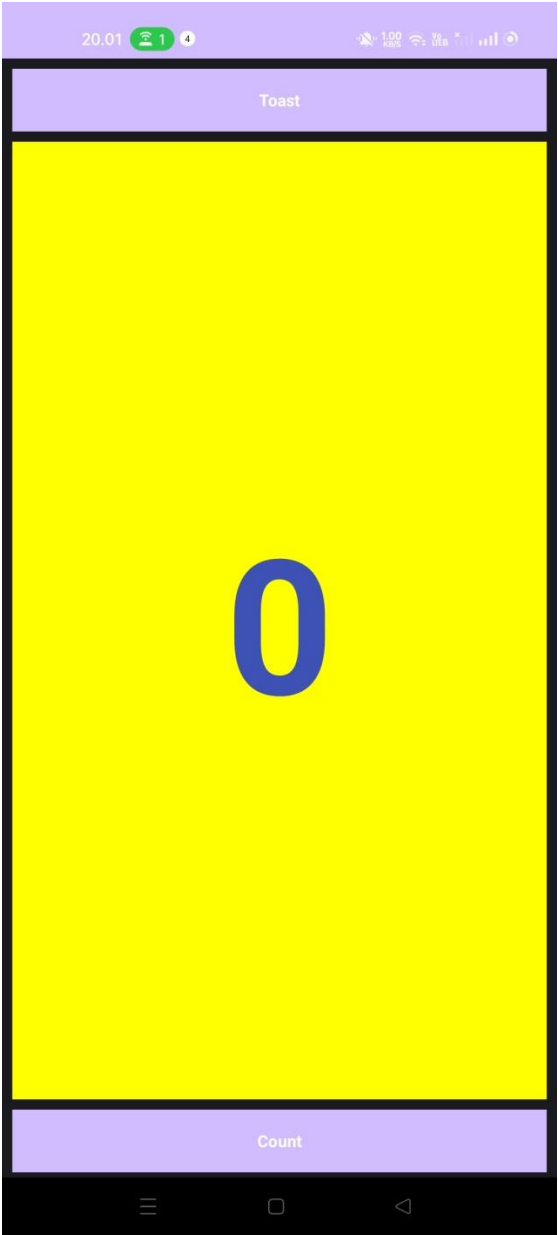
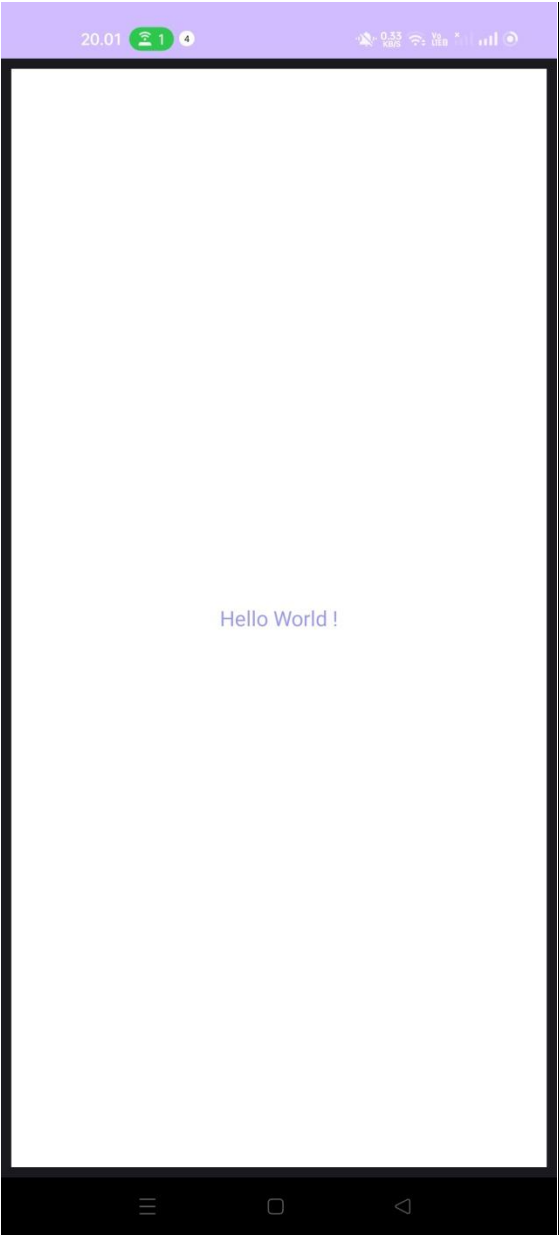
#### 2. Pengaturan Adapter dan Fragmen:

- Membuat instance dari `VPAdapter`.
- Menambahkan fragmen (Fantasy, Comedy, dan Romance) ke adapter.
- Menghubungkan `viewPager` dengan adapter.

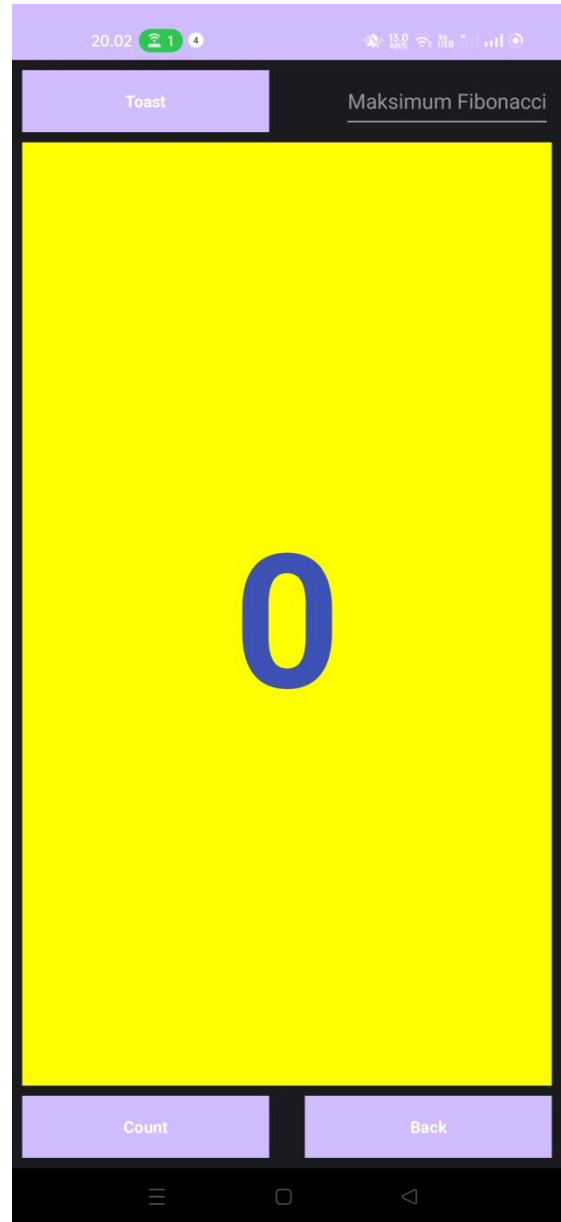
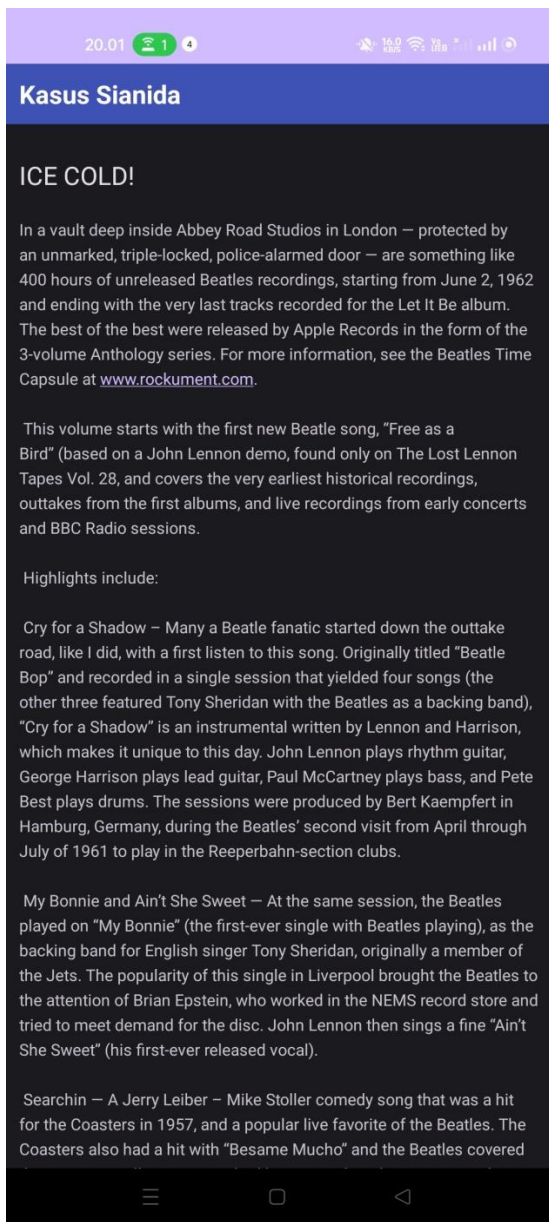
#### 3. TabLayout dan ViewPager Synchronization:

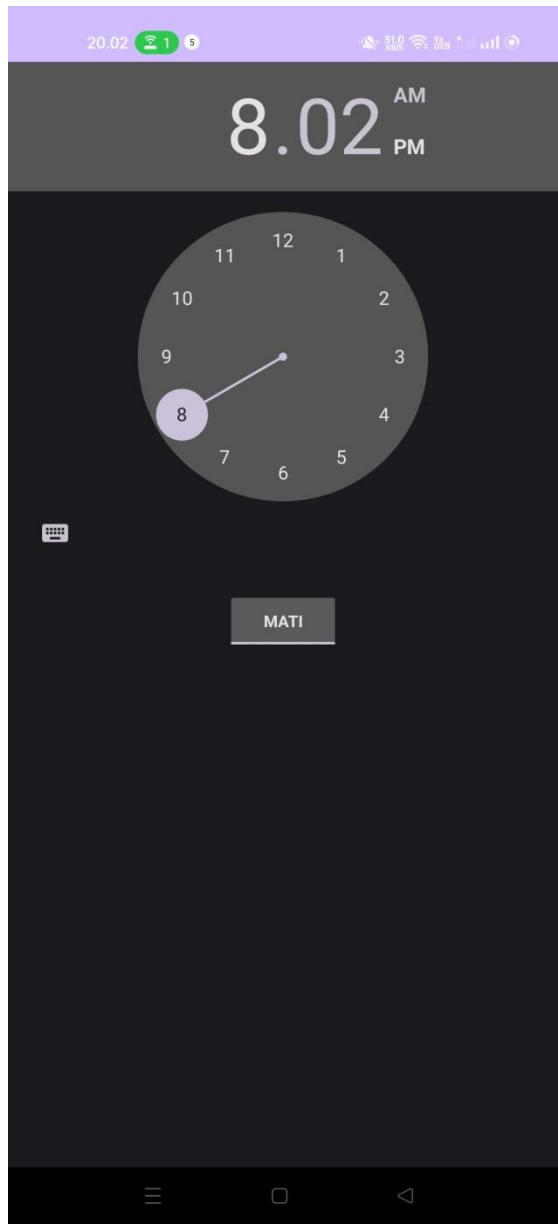
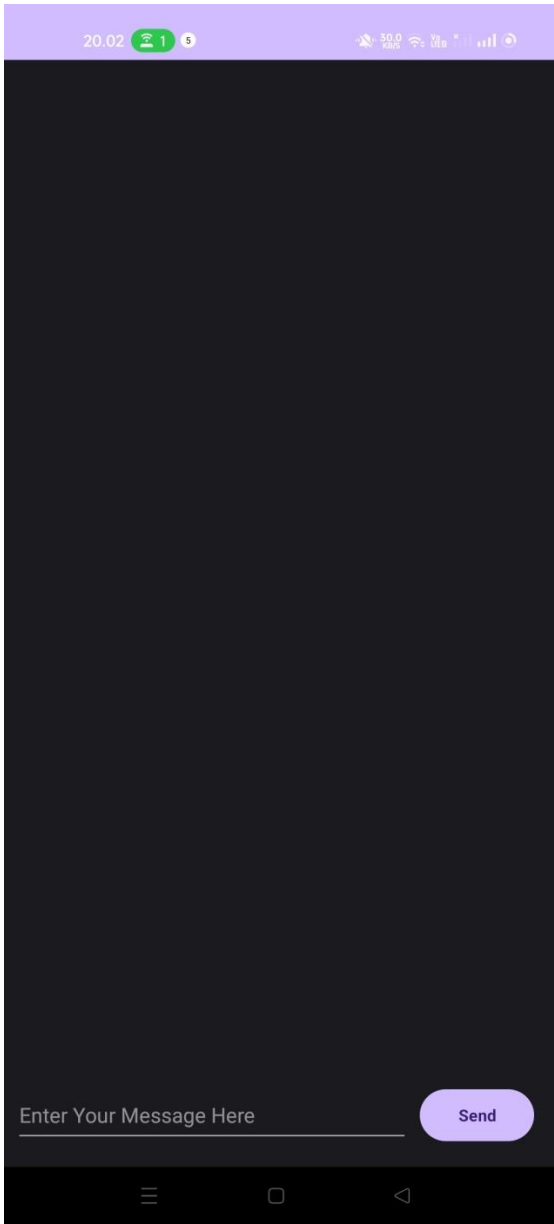
- Menggunakan metode `setupWithViewPager(viewPager)` pada `tabLayout` untuk mensinkronkan tab dengan ViewPager.

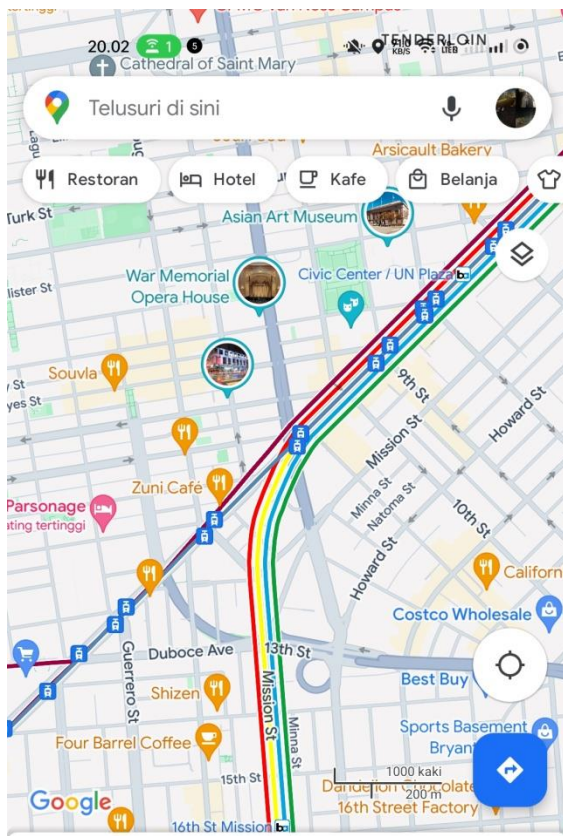
CAPTURE PROJECT











Terbaru di Civic Center

Tempat-tempat terkenal



Jelajahi Berangkat Tersimpan Kontribusi Info Terbaru

