

ULANGAN AKHIR SEMESTER - PEMROGRAMAN MOBILE 1

Mata Kuliah : Pemrograman Mobile 1

Dosen : Donny Maulana, S.Kom., M.M.S.I.

Nama : Rizjky Dito Ridwansyah

NIM : 312210405

Mata Kuliah Pemrograman mobile menggunakan aplikasi Android Studio.

1. Hello

Activity pertama yang dibuat adalah teks greeting, contoh Hello!

```
```Java
```

```
<TextView
```

```
 android:layout_width="wrap_content"
```

```
 android:layout_height="wrap_content"
```

```
 android:text="Halo Selamat Datang Di Aplikasi Saya.Perkenalkan Nama Saya Rizjky Dito
Ridwansyah"
```

```
 android:textColor="#FF0000"
```

```
 android:textAlignment="center"
```

```
 app:layout_constraintBottom_toBottomOf="parent"
```

```
 app:layout_constraintEnd_toEndOf="parent"
```

```
 app:layout_constraintStart_toStartOf="parent"
```

```
 app:layout_constraintTop_toTopOf="parent" />
```

```
```
```

2. Toast

```
```Java
```

```
 public void showToast(View view) {
```

```

 Toast toast = Toast.makeText(this, R.string.toast_message, Toast.LENGTH_SHORT);

 toast.show();
 }
}

```

Kode diatas adalah metode suatu kelas di Android yang bertujuan untuk menampilkan pesan toast. Penjelasannya:

- Metode `showToast(View view)`: metode yang akan dipanggil saat suatu tindakan terjadi, seperti menekan tombol yang terhubung dengan metode ini.
- `Toast toast = Toast.makeText(this, R.string.toast_message, Toast.LENGTH_SHORT);`:
- `Toast`: Objek dari kelas `Toast` yang digunakan untuk menampilkan pesan sementara.
- `makeText()`: Metode statis dari kelas `Toast` untuk membuat instans `Toast` baru.
- `this`: Merujuk pada konteks saat ini, yaitu aktivitas atau objek konteks tempat metode ini dipanggil.
- `R.string.toast_message`: Menggunakan string dari sumber daya string (biasanya didefinisikan di file `res/values/strings.xml`). Ini memungkinkan untuk lebih mudah mengelola pesan yang ditampilkan.
- `Toast.LENGTH_SHORT`: Menunjukkan durasi tampilan pesan toast, dalam hal ini, pesan akan ditampilkan dalam waktu singkat.
- `toast.show();`: Memanggil metode `show()` untuk menampilkan pesan toast yang telah disiapkan sebelumnya.

Jadi `Toast` itu tujuannya untuk menampilkan pesan singkat kepada pengguna (sesuai dengan nilai yang didefinisikan dalam `R.string.toast_message`) melalui elemen UI, seperti tombol, yang terhubung dengan metode `showToast`.

### 3. Scrolling (Ice Cold)

Menampilkan teks berita yang dapat di scroll. Menggunakan metode `scrollView` yang didalamnya diisi dengan `textView` berita.

```

` ` ` JAVA

```

```

<ScrollView

```

```

 android:layout_width="wrap_content"

```

```

 android:layout_height="wrap_content">

```

```

 <TextView ... />
 </ScrollView>
 ...

```

## ## 4. Fibonacci

### ### 1. Metode countUp(View view)

```
``JAVA
```

```

public void countUp(View view) {
 EditText inputLimit = findViewById(R.id.input_limit);
 if(!inputLimit.getText().toString().isEmpty()) {
 int limit = Integer.parseInt(inputLimit.getText().toString());
 if(mCount >= limit) {
 Toast.makeText(this, "Telah mencapai limit", Toast.LENGTH_LONG).show();
 return;
 }
 }
 int next = mCount;
 mCount = secondCount;
 secondCount = next + mCount;
 if (mShowCount != null)
 mShowCount.setTextColor(setColor());
 mShowCount.setText(Integer.toString(next));
}
...

```

Metode ini terfokus pada perhitungan deret Fibonacci dan pembaruan tampilan pada suatu aplikasi. Pertama, nilai batas diambil dari komponen EditText dengan ID "input\_limit". Proses ini melibatkan pengecekan apakah nilai batas tidak kosong. Jika tidak kosong, nilai tersebut dikonversi ke tipe data int. Selanjutnya, dilakukan pengecekan apakah nilai saat ini (mCount) lebih besar atau sama dengan batas yang telah ditetapkan. Jika kondisi ini terpenuhi, metode akan menampilkan pesan Toast dan menghentikan eksekusi lebih lanjut.

Jika batas tidak tercapai, metode akan melanjutkan dengan melakukan perhitungan deret Fibonacci dan mengupdate tampilan aplikasi sesuai dengan nilai yang dihitung. Selain itu, metode ini juga mengatur warna teks pada tampilan menggunakan metode setColor(). Dengan pendekatan ini, aplikasi dapat memberikan respons yang sesuai terhadap input pengguna dan memastikan bahwa perhitungan deret Fibonacci dan pembaruan tampilan berjalan dengan tepat.

### Poin-poin:

- Metode ini terkait dengan penanganan perhitungan deret Fibonacci dan pembaruan tampilan.
- Mengambil nilai batas dari EditText dengan ID input\_limit.
- Jika batas tidak kosong, maka konversi nilai tersebut ke tipe data int.
- Jika nilai saat ini (mCount) lebih besar atau sama dengan batas, tampilkan pesan Toast dan hentikan eksekusi metode.
- Selanjutnya, lakukan perhitungan deret Fibonacci dan perbarui tampilan sesuai dengan nilai yang dihitung.
- Set juga warna teks pada tampilan menggunakan metode setColor().

## 2. Metode reset(View view)

```
```JAVA
public void reset(View view) {
    mCount = 0;
    secondCount = 1;
    if (mShowCount != null)
        mShowCount.setText(Integer.toString(mCount));
}
```
```

Penjelasan

Metode ini digunakan untuk mereset nilai deret Fibonacci ke awal, di mana variabel mCount diatur kembali menjadi 0 dan secondCount menjadi 1. Setelah reset dilakukan, tampilan kemudian diperbarui dengan nilai-nilai yang telah direset. Proses ini membantu memulai ulang deret Fibonacci dari awal, memberikan basis yang bersih untuk menghitung nilai-nilai selanjutnya dalam deret tersebut.

Poin-poin:

- Metode ini digunakan untuk mereset nilai deret Fibonacci ke awal (mCount = 0 dan secondCount = 1).
- Kemudian, perbarui tampilan dengan nilai yang direset.

### 3. Metode setColor()

```
```JAVA
```

```
public int setColor() {  
    currentFib++;  
    if(currentFib % 2 == 0) {  
        return ContextCompat.getColor(this, R.color.black);  
    } else {  
        return ContextCompat.getColor(this, R.color.white);  
    }  
}  
}```
```

Penjelasan

Metode ini menawarkan fungsionalitas perubahan warna yang sesuai dengan setiap pemanggilannya. Untuk melacak urutan pemanggilan, digunakan variabel currentFib. Ketika currentFib memiliki nilai genap, metode ini mengembalikan warna hitam; sebaliknya, jika nilai currentFib adalah ganjil, warna yang dikembalikan adalah warna putih. Implementasi warna ini memanfaatkan fungsi ContextCompat.getColor() dengan merujuk pada sumber daya warna dari file resource (R.color). Dengan demikian, metode ini memberikan dinamika visual yang menarik, menciptakan variasi warna yang berubah sesuai dengan karakteristik bilangan dalam deret Fibonacci.

Poin-poin:

- Metode ini memberikan warna yang berubah sesuai dengan setiap pemanggilannya.
- Menggunakan variabel `currentFib` untuk melacak pemanggilan.
- Jika `currentFib` adalah bilangan genap, maka warna yang dikembalikan adalah warna hitam; jika ganjil, warna yang dikembalikan adalah warna putih.
- Warna didapatkan menggunakan `ContextCompat.getColor()` dengan referensi warna dari resource (`R.color`).

5. Pesan

Kode ini adalah bagian dari sebuah aplikasi Android yang terdiri dari dua aktivitas (`MainActivityOne` dan `MainActivitySecond`) dan digunakan untuk berinteraksi antaraktivitas. Berikut adalah penjelasan singkatnya:

Deklarasi Variabel:

- `LOG_TAG`: Variabel untuk tag log, digunakan untuk memudahkan identifikasi log.
- `EXTRA_MESSAGE`: String konstanta yang digunakan sebagai kunci untuk mengirim pesan antar aktivitas.
- `TEXT_REQUEST`: Integer konstanta yang digunakan sebagai kode permintaan untuk `startActivityForResult`.

Inisialisasi UI:

- `onCreate`: Metode yang dipanggil saat aktivitas pertama kali dibuat.
- Mengatur tata letak dengan menggunakan layout dari file `R.layout.activity_one`.
- Mendapatkan referensi ke elemen UI seperti `EditText` dan dua `TextView`.

Button Click Event:

- `launchSecondActivity`: Dipanggil saat tombol diaktivitas ini diklik.
- Membuat objek `Intent` untuk memulai aktivitas kedua (`MainActivitySecond`).
- Mengirim pesan yang dimasukkan oleh pengguna melalui `EditText` ke aktivitas kedua menggunakan `putExtra`.
- Memulai aktivitas kedua dengan `startActivityForResult` untuk menerima hasil dari aktivitas kedua.

Handle Result:

- `onActivityResult`: Metode yang dipanggil ketika aktivitas kedua selesai dan memberikan hasil kembali.
- Memeriksa apakah hasil diterima sesuai dengan kode permintaan `(TEXT_REQUEST)`.
- Jika hasil OK, menampilkan header dan isi pesan yang dikirim kembali dari aktivitas kedua pada dua `TextView`.

Kode ini menciptakan pengalaman pengguna di mana pengguna dapat memasukkan pesan di `MainActivityOne`, mengirimkannya ke `MainActivitySecond` melalui tombol, dan menampilkan balasan di `MainActivityOne` setelah aktivitas kedua selesai.

6. Menu

Menu ini adalah bagian dari sebuah aplikasi Android yang mengimplementasikan aktivitas menu dengan menggunakan `CardView` untuk setiap opsi menu. Kita dapat memilih aplikasi-aplikasi yang kita buat, dengan menggunakan metode `intent`. Berikut adalah penjelasannya:

Deklarasi Variabel:

`tombolSatu`, `tombolDua`, ..., `tombolDelapan`: Variabel yang merepresentasikan `CardView` untuk setiap opsi menu.

Inisialisasi UI:

- `onCreate`: Metode yang dipanggil saat aktivitas pertama kali dibuat.
- Mengatur tata letak dengan menggunakan layout dari file `R.layout.activity_menuaplikasi`.
- Menginisialisasi setiap `CardView` dan menetapkan `OnClickListener` untuk setiap tombol.

Pengaturan Tombol:

- Setiap tombol (`tombolSatu` hingga `tombolDelapan`) memiliki `OnClickListener` yang memulai aktivitas tertentu ketika tombol tersebut diklik. Aktivitas yang dimulai bervariasi dari `MainActivity` hingga `ViewPagerActivity`, dan bahkan membuka aplikasi Google Maps untuk menavigasi ke "Universitas Pelita Bangsa" pada tombol tujuh.

Kode ini memberikan pengguna akses ke berbagai fitur dan aktivitas dalam aplikasi melalui menu dengan menggunakan `CardView` sebagai elemen UI yang responsif terhadap sentuhan.

7. Maps

- Men-setting dependency maps pada `build.gradle`.

```
```JAVA
implementation("com.google.android.gms:play-services-maps:17.0.1")
```
```

Membuka aplikasi Google Maps dengan memanggil aplikasi sistem (Implicit intent).

```
```JAVA
Intent mapIntent = new Intent(Intent.ACTION_VIEW);
 mapIntent.setPackage("com.google.android.apps.maps");
```
```

Lalu mengarahkan kita ke target lokasi yang telah di setting.

```
```JAVA
if (mapIntent.resolveActivity(getPackageManager()) != null) {
 Uri gmmIntentUri = Uri.parse("geo:0,0?q=" + Uri.encode("Universitas Pelita Bangsa"));
}
```
```

8. Splash

Implementasi dari sebuah Splash Screen pada aplikasi, berguna sebagai intro aplikasi. Berikut adalah deskripsi singkat bagian-bagian kode:

- Men-setting Splash pada `AndroidManifest` sebagai aktivitas pertama yang dijalankan

```
```Java
<activity android:name=".Splash"
 android:exported="true">
 <intent-filter>
```
```

- Memberikan waktu splash selama 3 detik.

```
```Java
private static final int SPLASH_DELAY = 3000;
```
```


- Menggunakan Handler untuk menunda eksekusi kode selama 3 detik (sesuai dengan SPLASH_DELAY). Setelah delay berakhir, akan dijalankan Runnable yang berisi kode untuk membuat dan menjalankan Intent ke Menu, dan kemudian menyelesaikan aktivitas Splash Screen.

```
```Java
new Handler().postDelayed(new Runnable() {
 @Override
 public void run() {
 Intent intent = new Intent(Splash.this, Menu.class);
 startActivity(intent);
 finish();
 }
}, SPLASH_DELAY);
```
```

9. Fragment

- ViewPagerAdapter

`ViewPagerAdapter` adalah sebuah kelas yang mengimplementasikan `FragmentPagerAdapter` pada aplikasi Android. Fungsi utamanya adalah mengelola dan menyediakan tampilan fragmen untuk ditampilkan dalam suatu `ViewPager`.

- ViewPagerActivity

`ViewPagerActivity` mengelola daftar fragmen yang akan ditampilkan dalam `ViewPager`. Ini bisa termasuk membuat instance fragmen, menambahkan fragmen ke dalam `ViewPagerAdapter`, dan mengatur judul/judul halaman untuk setiap fragmen.

```
```Java
@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.fragment_menu);

 ViewPager viewPager = findViewById(R.id.viewPager);
 TabLayout tabLayout = findViewById(R.id.tab_layout);
}
```

```

ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());
adapter.addFragment(new Tab1(), "Thriller");
adapter.addFragment(new Tab2(), "Action");
adapter.addFragment(new Tab3(), "Romance");

viewPager.setAdapter(adapter);
tabLayout.setupWithViewPager(viewPager);
}
}
...

```

## 10. Sinopsis dan Trailer

```

```Java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.sinopsis_(judulfilm));

    WebView webView1 = findViewById(R.id.webView1);

    String video1 = "<iframe width=\"100%\" height=\"100%\"
src=\"https://www.youtube.com/embed/iaD2f2O9wGk\" title=\"YouTube video player\"
frameborder=\"0\" allow=\"accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
picture-in-picture; web-share\" allowfullscreen></iframe>";

    webView1.loadData(video1, "text/html", "utf-8");
    webView1.getSettings().setJavaScriptEnabled(true);
    webView1.setWebChromeClient(new WebChromeClient());
}
}
...

```

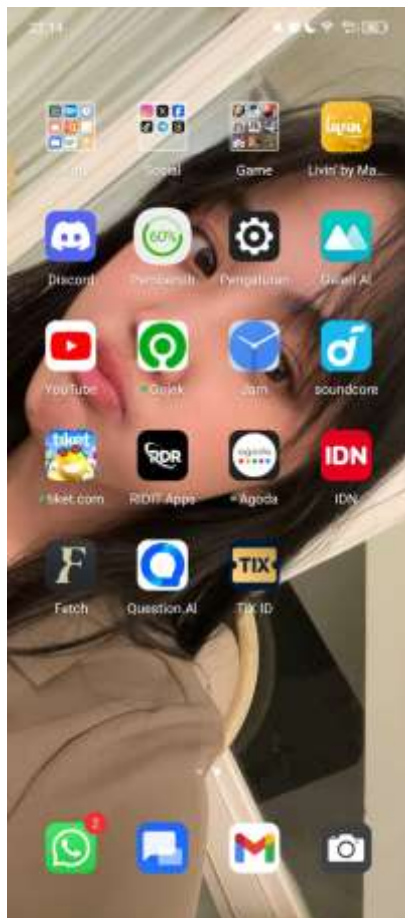
- Merinci implementasi aktivitas dalam aplikasi Android yang bertujuan untuk menampilkan sinopsis dengan trailer menggunakan `WebView`. Pertama, pada metode `onCreate`, layout aktivitas diatur menggunakan metode `setContentView` dengan merujuk pada layout `sinopsis_(film)`.

- Selanjutnya, sebuah `WebView` dengan id "`webView1`" diinisialisasi dari layout XML dan kemudian dimanipulasi menggunakan beberapa konfigurasi. Video trailer dari YouTube disematkan ke dalam `WebView` menggunakan tag `iframe` HTML dan dimuat menggunakan metode `loadData`. Pengaturan JavaScript diaktifkan dengan `webView1.getSettings().setJavaScriptEnabled(true)`, dan `setWebChromeClient` digunakan untuk mendukung fungsionalitas Chrome pada WebView.

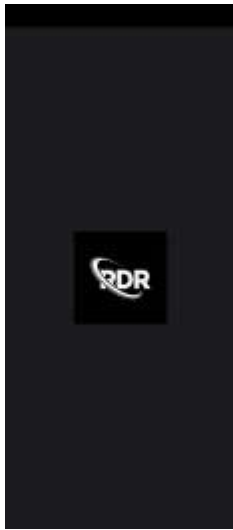
- Dengan pengaturan ini, aktivitas ini secara efektif menampilkan sinopsis film dalam format video melalui `WebView`, memberikan pengguna pengalaman visual yang langsung terintegrasi ke dalam aplikasi.

hasil semua program :

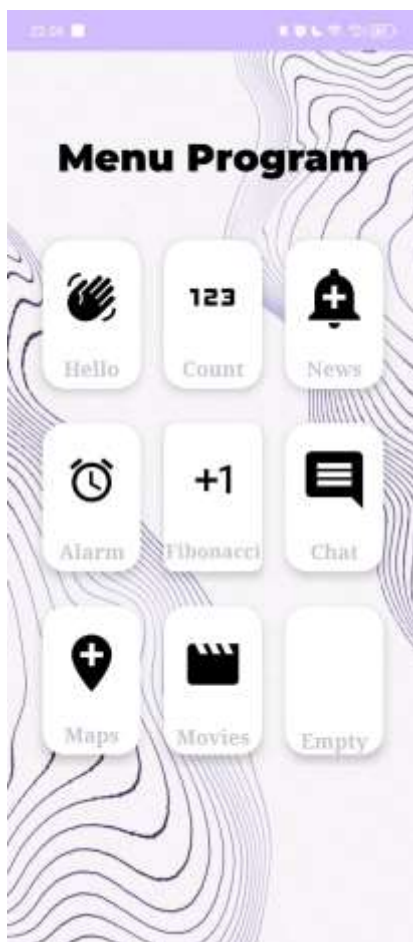
Nama Aplikasi : RIDIT Apps



Splash :



Menu Program :



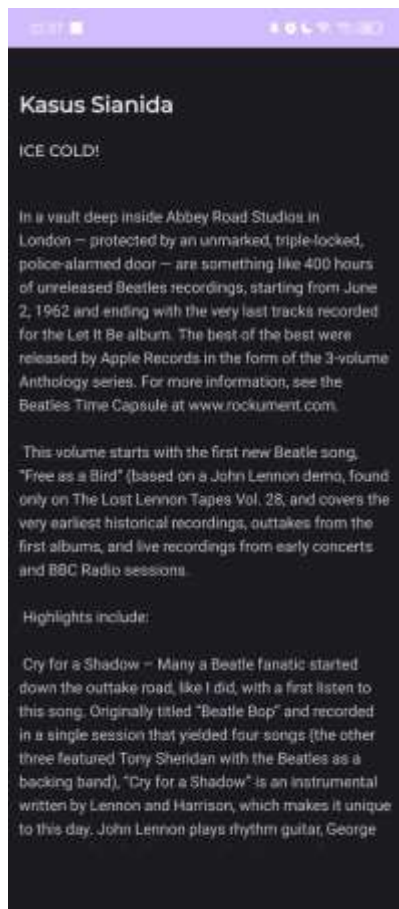
Hello:



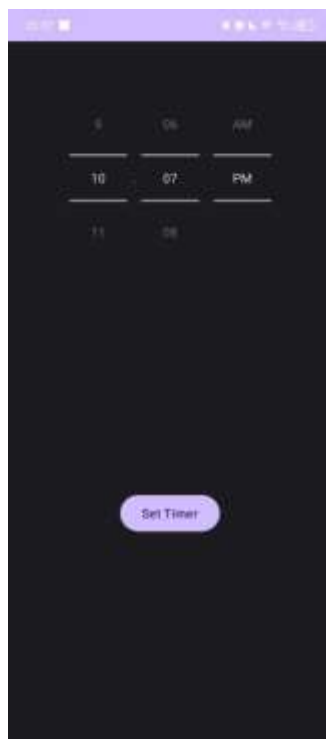
Count:



News:



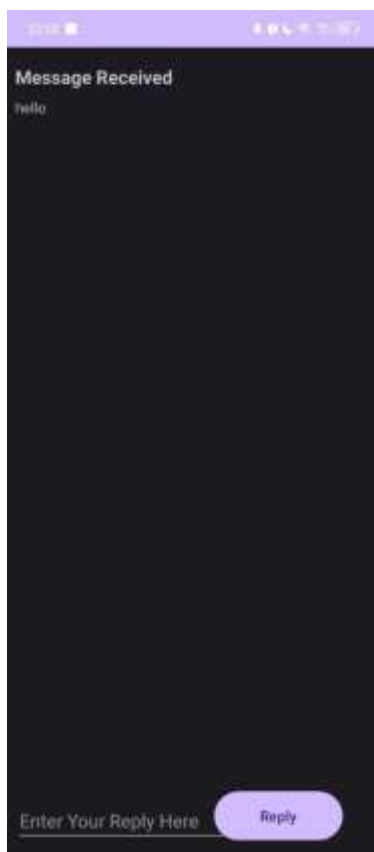
Alarm:



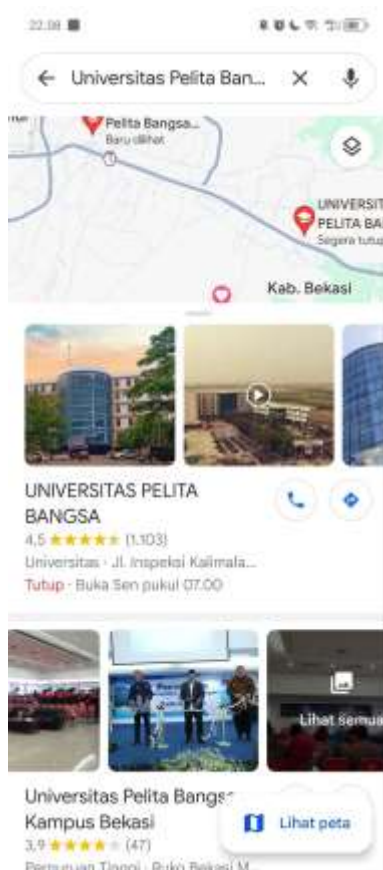
Fibonacci:



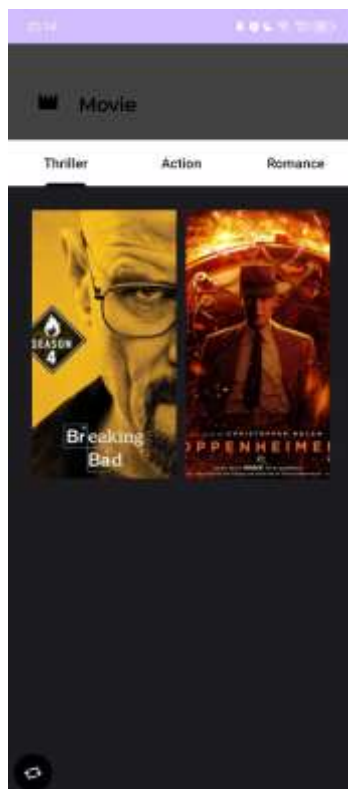
Chat:



Maps:



Sinopsi dan Trailer Film:



TERIMA KASIH