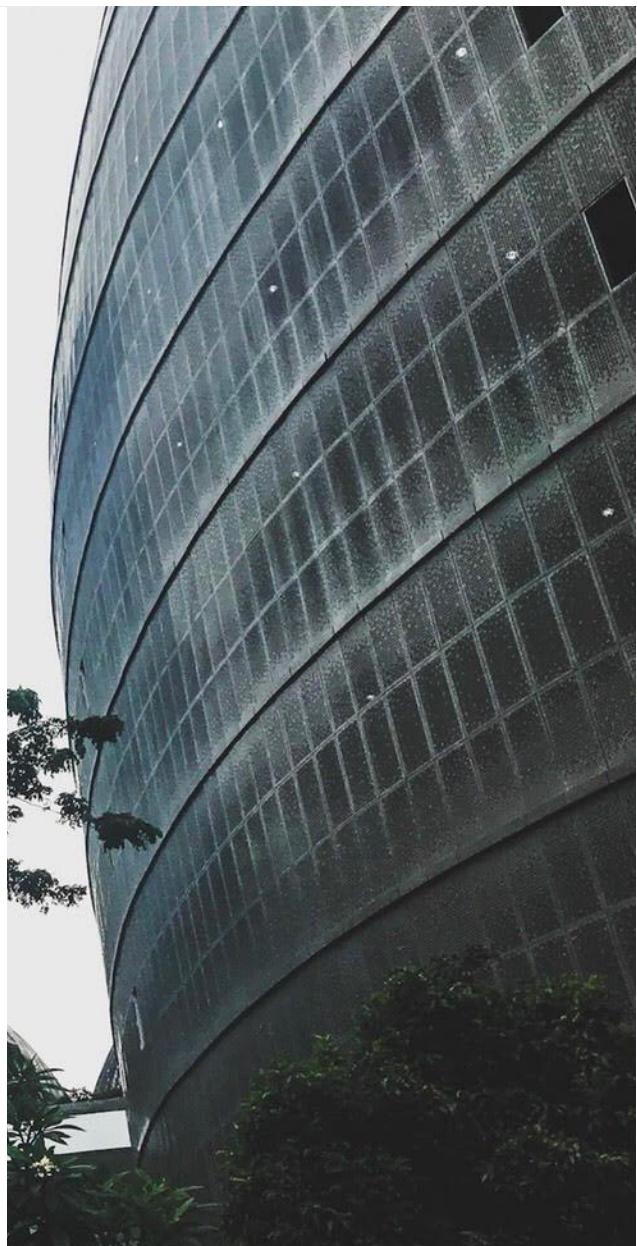


MODUL PRAKTIKUM

**IF633 – PEMROGRAMAN MOBILE
PROGRAM SARJANA S1 INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA**



**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA**

Gedung B Lantai 5, Kampus UMN
Jl. Scientia Boulevard, Gading Serpong, Tangerang, Banten-15811 Indonesia
Telp: +62-21.5422.0808 (ext. 1803), email: ict.lab@umn.ac.id, web: umn.ac.id

DAFTAR ISI

CAPAIAN PEMBELAJARAN PRAKTIKUM.....	3
MODUL 1 (PENGENALAN SISTEM OPERASI ANDROID)	4
MODUL 2 (ANDROID ACTIVITY DAN PENGATURAN LAYOUT, dan DEBUGGING)	11
MODUL 3 (RECYCLER VIEW, FLOATING ACTION BUTTON, DAN TOAST)	28
MODUL 4 (INTENT, ACTIVITY, FRAGMENT, DAN PASSING DATA ANTAR INTENT)	38
MODUL 5 (CUSTOM VIEW, CANVAS, PAINT DAN DRAWING)	56
MODUL 6 (ANIMASI DAN MULTIMEDIA)	75
MODUL 7 (MULTIMEDIA DAN KAMERA)	87
MODUL 8 (ANDROID STORAGE, SAVED INSTANCE AND SHARED PREFERENCES)	91
MODUL 9 (SQLITE DATABASE DAN ROOM)	108
MODUL 10 (BACKGROUND PROCESSES).....	125
MODUL 11 (WEB SERVICE).....	141
MODUL 12 (SENSOR)	164
MODUL 13 (GOOGLE MAP API)	171

CAPAIAN PEMBELAJARAN PRAKTIKUM

1. Mahasiswa mampu menjelaskan alur pengembangan aplikasi Android (C2).
2. Mahasiswa mampu menggunakan berbagai Teknik pengaturan layout untuk menghasilkan tampilan aplikasi yang rapih (C₃, C₄).
3. Mahasiswa mampu menggunakan berbagai komponen user-interface dan user-interaction untuk menghasilkan tampilan aplikasi dan membuat interaksi antartampilan (C₃).
4. Mahasiswa mampu menggunakan Activity, Intent, dan Fragment untuk menghasilkan tampilan yang dinamis dan fleksibel (C₃).
5. Mahasiswa mampu mengatur grafik yang ditampilkan pada layar perangkat (seperti kanvas) secara interaktif (C₄).
6. Mahasiswa mampu memanfaatkan dan menerapkan fitur multimedia dan animasi pada aplikasi yang dibuat (C₃).
7. Mahasiswa mampu menghasilkan aplikasi yang bermanfaat sesuai studi kasis atau kebutuhan masyarakat (C₄, C₅, C₆).
8. Mahasiswa mampu melakukan pengelolaan data pada media penyimpanan internal smartphone (C₄).
9. Mahasiswa mampu menghasilkan aplikasi sederhana yang memanfaatkan basis data internal menggunakan SQLite (C₄, C₅, C₆).
10. Mahasiswa mampu menerapkan penggunaan Service dan BackgroundProcess (C₃).
11. Mahasiswa mampu melakukan pengelolaan data pada basis data online yang diakses melalui web service (C₄).
12. Mahasiswa mampu memanfaatkan dan menerapkan sensor pada perangkat seperti sensor pergerakan dan sensor lokasi pada Aplikasi Android (C₄).
13. Mahasiswa mampu memanfaatkan dan menerapkan location-based-service pada aplikasi yang dibuat (C₃).
14. Mahasiswa mampu mempublikasikan aplikasi yang dihasilkan pada Google Play Store (C₄, C₅, C₆).

MODUL 1

(PENGENALAN SISTEM OPERASI ANDROID)

DESKRIPSI TEMA

- Pengenalan, Arsitektur, Versi, Perangkat, dan Distribusi Platform, Struktur File Android
- Instalasi
- Perkenalan Gradle

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menjelaskan cara kerja sistem operasi Android dan alur pengembangan aplikasi Android.

PENUNJANG PRAKTIKUM

1. Software Android Studio.

LANGKAH-LANGKAH PRAKTIKUM

Langkah Instalasi Android Studio (Instalasi Mudah)

Warning:

Develop Android pada Windows membutuhkan spesifikasi yang cukup mumpuni apabila menggunakan emulator Android!

Lakukan langkah ini apabila Anda menggunakan processor Intel dan menggunakan 2 partition di Windows (Mac dan 1 Partition lakukan tahap Langkah Instalasi Android Studio (Instalasi Otomatis)

1. Unduh installer dan file yang ada pada tautan berikut:

https://drive.google.com/open?id=1NgRkuyCQeHMhhgK7iYuQMqRi1bITo_Vj

- android-config_20180719.zip (**174 MB**)
 - android-gradle_20180719.zip (**381 MB**)
 - android-studio_20180719.zip (**5.49 GB**)
 - 7za.exe
 - 7shortcut.exe
 - batch_extract_mobile_20180719.bat
2. Taruhlah semua (6 file) tersebut ke dalam D:\INSTALLER-FTI\ (Apabila folder belum ada, buatlah folder tersebut)

3. Setelah semua file ditaruh, jalankan **batch_extract_mobile_20180719.bat**
PS: Pastikan Anda memiliki +/- 20GB sebelum menjalankan batch ini!
4. Tunggu hingga batch selesai dijalankan (+/- 10-45menit, tergantung dari komputer/laptop yang Anda miliki)
5. Apabila batch sudah selesai dijalankan tanpa error, batch akan membuat folder **D:\FTI\IF634_IS534_MOBILE**
PS: Mohon untuk tidak me-rename folder tersebut dikarenakan setting Android Studio banyak yang **PATH based**.
6. Buka Folder **D:\FTI\IF634_IS534_MOBILE\android-sdk\extras\intel\Hardware_Accelerated_Execution_Manager** dan jalankan **intelhaxm-android.exe** (selanjutnya akan disebut **HAXM**)
HAXM ini berfungsi untuk mempercepat jalannya *emulator* Android pada saat dijalankan oleh Android Studio. Apabila memungkinkan pada saat instalasi HAXM setting RAM menjadi 2 GB, apabila tidak memungkinkan 1 GB pun sudah cukup.
7. Setelah selesai memasang HAXM, jalankan Android Studio dengan membuka **D:\FTI\IF634_IS534_MOBILE\AndroidStudio** (Shortcut)
8. Selamat Anda sudah selesai melakukan instalasi Android Studio, lanjut ke Tahap Tutorial Android Studio.

Langkah Instalasi Android Studio (Instalasi Otomatis)

Warning:

Develop Android pada Windows membutuhkan spesifikasi yang cukup mumpuni apabila menggunakan emulator Android!

Pada saat instalasi dengan cara Otomatis ini Anda akan mengunduh BANYAK file dari Google, pastikan Anda memiliki kuota internet yang mencukupi. (> 5 GB)

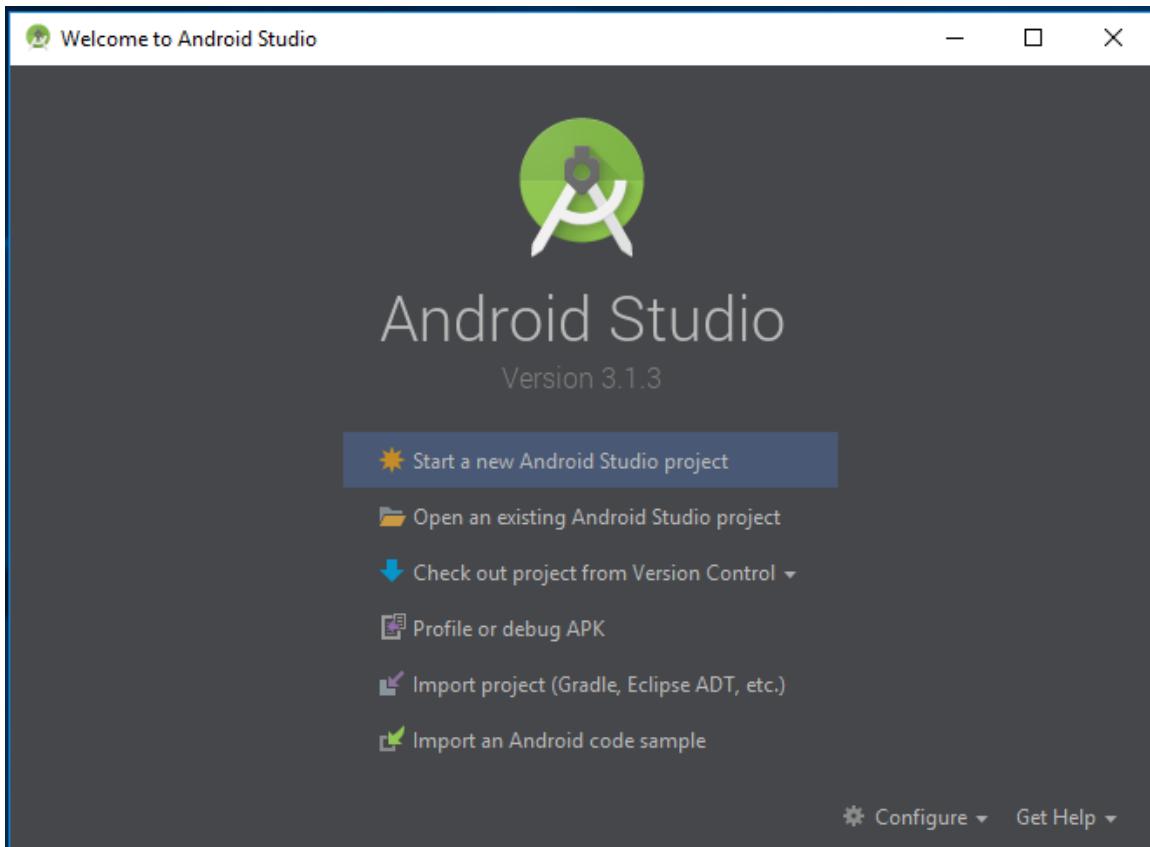
1. Unduh Installer Android Studio dari [sini](#).
2. Install Android Studio

Warning:

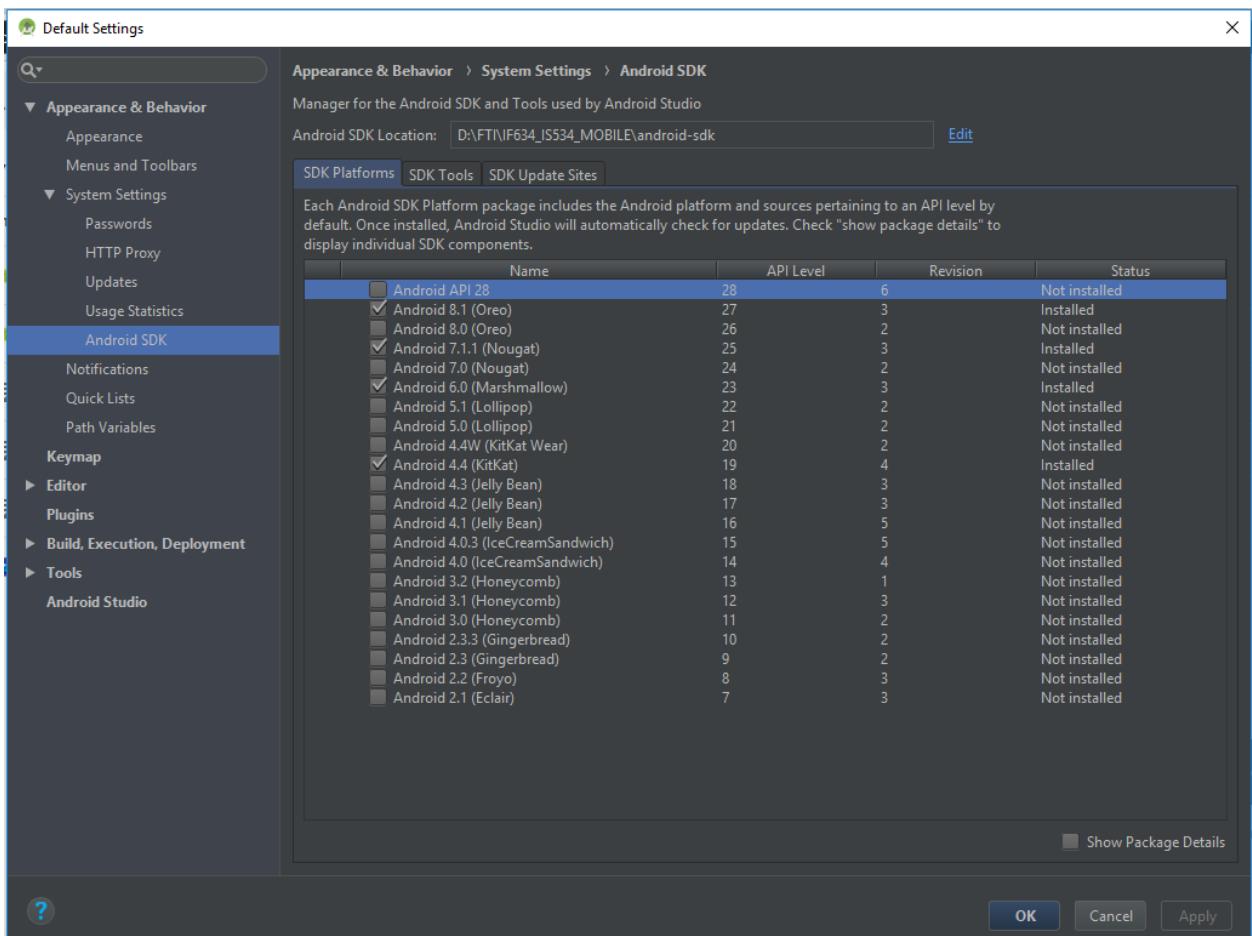
Wajib terkoneksi internet !

3. Ketika Instalasi Android Studio, apabila Anda menggunakan processor Intel, jangan lupa untuk memasang **Hardware Accelerated Execution Manager (HAXM)** ketika ditawarkan pada saat instalasi. Gunakan RAM 2 GB apabila memungkinkan, 1 GB untuk minimal RAM usage.

4. Bukalah Android Studio setelah selesai instalasi dan Anda akan diminta untuk melakukan konfigurasi awal.
5. Setelah selesai melakukan konfigurasi awal, Anda akan disuguhkan tampilan seperti berikut:



6. Tekan tombol Configure yang terletak di kanan bawah kemudian tekan SDK Manager (Tab SDK Platform) dan Anda akan disuguhkan tampilan sebagai berikut:



7. Pada kanan bawah, tekan checkbox Show Package Details.
8. Untuk menyamakan *environment* dengan Android Studio yang dipakai di kampus, lakukanlah instalasi untuk SDK sebagai berikut (Tab SDK Platform):
 - Android 8.1 (Oreo)
 - Android SDK Platform 27
 - Sources for Android 27
 - Android 7.1.1 (Nougat)
 - Android SDK Platform 25
 - Sources for Android 25
 - [!] Google APIs Intel x86 Atom System Image
 - [!] Google Play Intel x86 Atom System Image
 - Android 6.0 (Marshmallow)
 - Google APIs
 - Android SDK Platform 23

- Sources for Android 23
- Android 4.4 (KitKat)
 - Google APIs
 - Android SDK Platform 19
 - Sources for Android 19

[!] = Processor Intel Only

Untuk Instalasi pada processor non-Intel, Anda memiliki 2 Opsi:

- Jalur Emulator
 - Tambahkan pada Android 7.1.1 → Google APIs ARM 64 v8a System Image DAN Google APIs ARM EABI v7a System Image
- Jalur HP Sendiri
 - Apabila *Smartphone* Anda sendiri menggunakan Android 6.0 ke atas, Anda sudah bisa mengikuti perkuliahan dengan modal *Smartphone* yang dicolok ke PC dan dihubungkan dengan adb.
 - Tutorial ini tidak membahas cara mengkoneksikan *Smartphone* ke Android Studio, sehingga Anda perlu membaca lebih banyak mengenai hal ini.

9. Pada Tab **SDK Tools**, Centang **Show Package Details** lakukan instalasi pada tools berikut:

- Android SDK Build-Tools
 - 19.1.0
 - 20.0.0
 - 21.1.2
 - 22.0.1
 - 23.0.1
 - 23.0.2
 - 23.0.3
 - 24.0.0
 - 24.0.1
 - 24.0.2
 - 24.0.3
 - 25.0.0
 - 25.0.1
 - 25.0.2
 - 25.0.3
 - 26.0.0
 - 26.0.1
 - 26.0.2
 - 26.0.3
 - 27.0.0
 - 27.0.1
 - 27.0.2
 - 27.0.3
 - 28.0.0
- Android Emulator
- Android SDK Platform-Tools
- Android SDK Tools
- Documentation for Android SDK

- Google Play APK Extension Library
 - Google Play Licensing Library
 - Google Play Services
 - Google USB Driver
 - Google Web Driver
 - Instant Apps Development SDK
 - [!] Intel x86 Emulator Accelerator (HAXM Installer)
 - Support Repository
 - ConstraintLayout for Android
 - 1.0.0
 - 1.0.1
 - 1.0.2
 - Solver for ConstraintLayout
 - 1.0.0
 - 1.0.1
 - 1.0.2
 - Android Support Repository
 - Google Repository
10. Setelah selesai melakukan instalasi, selamat, Anda sudah memasang Android Studio yang sama dengan lab UMN. Lanjutkan ke Tahap Tutorial Android Studio.

Tutorial: Struktur File Android

Pada langkah ini Anda akan mempelajari struktur file pada program Android.

1. Bukalah Android Studio dan pilih **Start a new Android Studio Project**
2. Pada saat pembuatan projek Android, Anda akan menginput beberapa hal:

Application Name merupakan nama dari aplikasi yang dibuat, boleh menggunakan spasi. Default projek di pembelajaran UMN adalah

Week [xx]_[NIM]. Contoh: **Week 01_12001**.

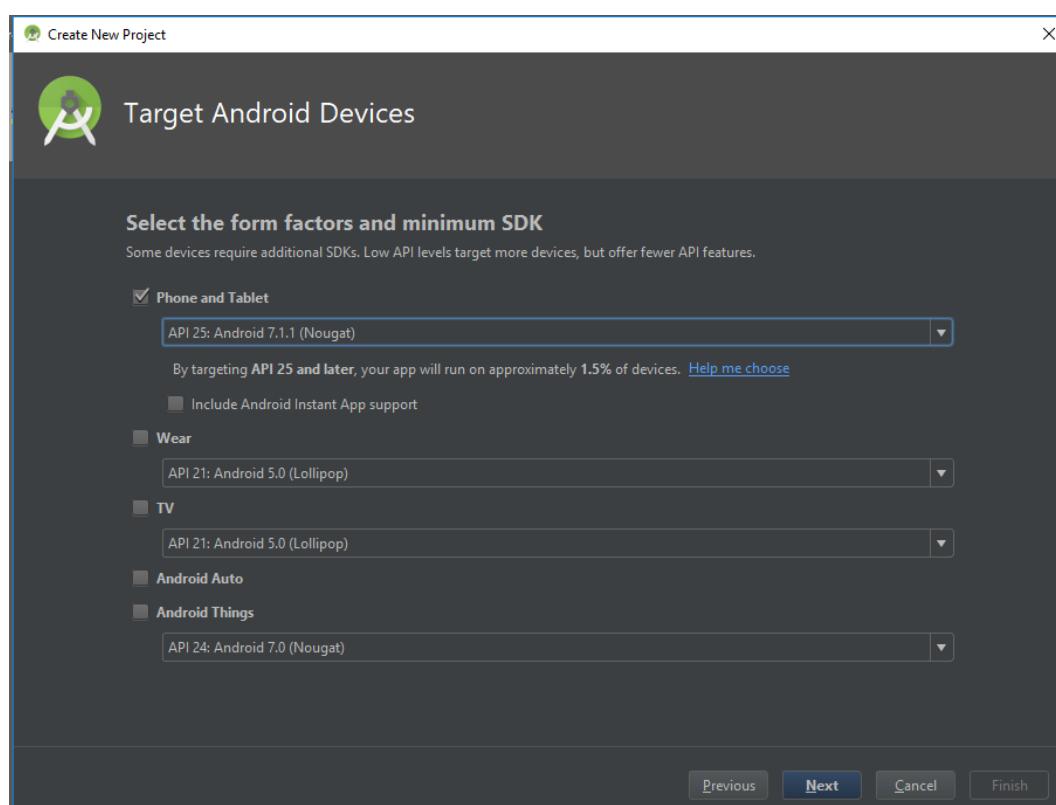
Company Domain merupakan nama domain dari perusahaan yang digunakan. Untuk projek di UMN gunakan Domain **umn.ac.id**.

Project Location merupakan directory di mana projek tersebut akan dibuat. Default pada projek di pembelajaran UMN adalah: **D:\FTI\IF634_IS534_MOBILE\workspaces**

Apabila selesai menginput data yang dibutuhkan, tekanlah tombol **Next**.

3. Selanjutnya Anda akan dihadapkan pada Target Android Devices.

Pada pembelajaran di UMN, gunakan Phone and Tablet **API 25: Android 7.1.1 (Nougat)** tanpa menggunakan Android Instant App Support, kemudian tekan tombol **Next**



4. Pada **Add an Activity to Mobile**, di pertemuan ini, gunakanlah **Empty Activity**, kemudian tekan **Next**
5. Kemudian pada **Configure Activity**, Anda akan memberikan nama untuk 2 file.

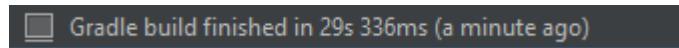
Activity Name merupakan nama “halaman” yang akan dibuat sebagai “halaman” utama pada projek Android yang akan dibuat.

Layout Name merupakan nama “template halaman” yang akan digunakan oleh **Activity Name** yang dibuat.

PS: Pada projek Android umumnya, Activity dan “halaman” lainnya berupa **.java** dan Layout atau “template halaman” merupakan file **.xml**.

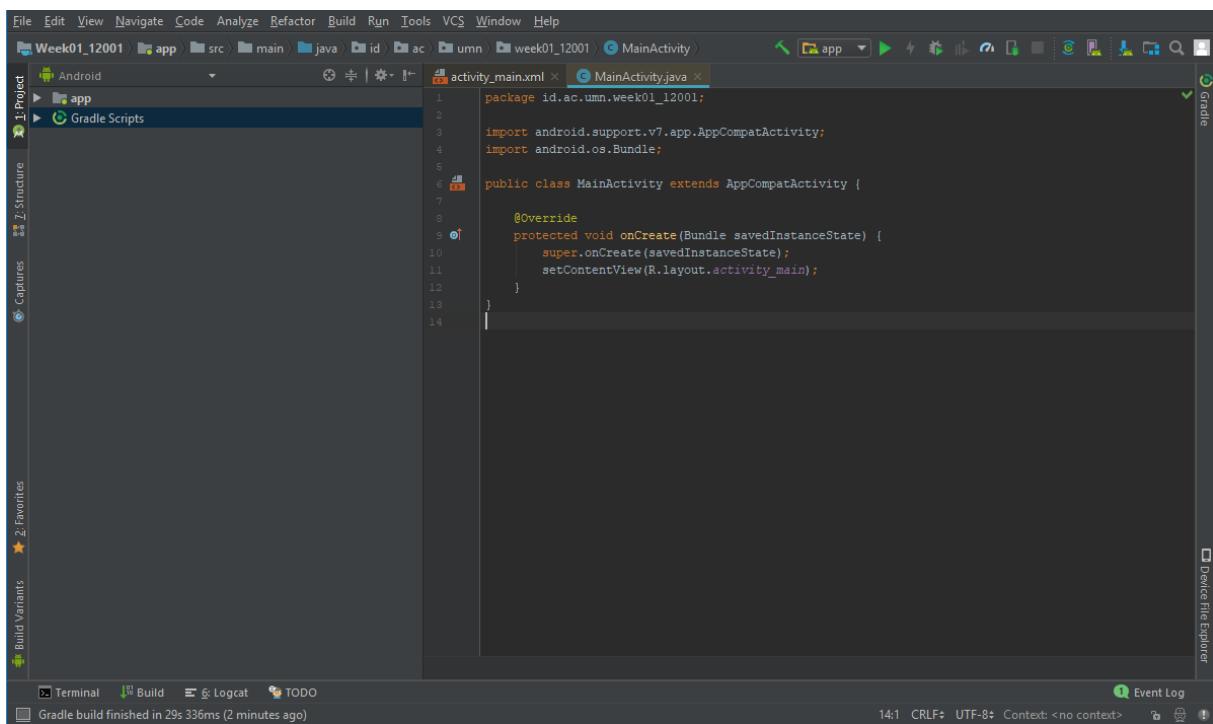
Untuk Activity Name dan Layout Name Anda dibebaskan menggunakan nama apapun, selama Anda mengingatnya (karena akan digunakan pada tahapan selanjutnya)

Setelah selesai Configure Activity, tekan tombol Finish kemudian tunggu Gradle hingga selesai membuat Filenya.



Tulisan di atas dapat dilihat setelah IDE menunggu Gradle selesai membuat projek. Pengertian Gradle akan dibahas pada tahapan selanjutnya.

6. Setelah **Gradle build finished**, Anda akan disuguhkan tampilan sebagai berikut.



```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Week01_12001 app src main java id ac umn week01_12001 MainActivity
Android app Gradle Scripts
activity_main.xml MainActivity.java
1 package id.ac.umn.week01_12001;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13
14

```

Event Log Device File Explorer

Terminal Build Logcat TODO

Gradle build finished in 29s 336ms (2 minutes ago)

14:1 CRLF: UTF-8 Context: < no context >

7. Pada bagian kiri, tekanlah tombol Dropdown pada Folder **app** untuk melihat Struktur Folder dan file yang ada pada Android.

Penjelasan:

app\manifest\AndroidManifest.xml

merupakan konfigurasi yang berisi nama aplikasi, nama activity, hingga action yang ada pada activity android tersebut dalam bentuk xml. AndroidManifest.xml ini juga sebagai penentu permission yang boleh digunakan oleh Android.

Misalnya:

Di bawah tag <manifest> tambahkan tag

<uses-permission android:name="android.permission.INTERNET" />

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="id.ac.umn.week01_12001">

    <uses-permission android:name="android.permission.INTERNET" />

```

(Package merupakan domain name yang dibalik + nama projek, samakan dengan nama projek yang Anda buat.)

Tag <uses-permission> ini menjelaskan bahwa aplikasi yang Anda buat ini nantinya akan meminta permission dari Android OS untuk melakukan koneksi internet.

app\java\[packagename]

Folder ini merupakan tempat di mana Activity dan semua file java dari logic program Anda diletakkan.

app\java\[packagename]\ (androidTest)

Folder ini merupakan tempat di mana Android UI Test akan diletakkan.

app\java\[packagename]\ (test)

Folder ini merupakan tempat di mana Android/Java Logic Test akan diletakkan.

app\res\drawable

Folder ini merupakan tempat di mana gambar vector yang digunakan pada aplikasi yang ada miliki diletakkan.

app\res\layout

Folder ini merupakan tempat dari layout yang dibutuhkan oleh aplikasi Android yang Anda buat, berupa file XML.

app\res\mipmap

Folder ini merupakan tempat dimana asset (khususnya icon dalam bentuk png dan berbagai ukuran, tergantung besar kecilnya layar).

app\res\values

Folder ini merupakan tempat di mana value statis yang digunakan oleh projek Android diletakkan. Misalnya: String tulisan yang ada pada button / "Halaman" Android, warna warna yang digunakan oleh aplikasi Android, hingga style / templating warna dan tulisan yang digunakan oleh aplikasi Android Anda.

Tutorial: Perkenalan Gradle

Gradle dalam dunia pemrograman sebenarnya merupakan tools yang dibuat untuk memudahkan otomatisasi pembuatan Aplikasi.

Dalam dunia Android, seringkali programmer menggunakan Gradle sebagai “konfigurasi” sederhana untuk menambahkan package (library) external di luar yang telah disediakan oleh Android itu sendiri.

1. Setelah melihat folder **app**, selanjutkan kita akan melihat **Gradle Scripts**.

File yang akan dilihat adalah:

build.gradle (Project: [namaproject])

Gradle file ini merupakan file dimana kita akan mengedit repository center (tempat dimana kita mengunduh package / library yang akan diunduh, atau mengganti versi dari Gradle yang digunakan).

(Pada Pemrograman Mobile kita sebisa mungkin tidak mengedit file ini)

build.gradle (Module: app)

Gradle file ini merupakan file dimana kita akan menambah package eksternal yang akan kita gunakan nanti.

2. Untuk mencoba apa yang dimaksud, bukalah

app\java\[namapackage]\[namaactivity] yang telah dibuat.

3. Ketiklah kode ini pada bagian **import**.

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.RecyclerView;
```

4. Hover mouse pada tulisan yang berwarna merah (**RecyclerView**) tersebut, kemudian bacalah error yang muncul.

5. Selanjutnya bukalah Gradle Scripts\build.gradle (Module: app) dan tambahkan kode seperti berikut pada **dependencies**.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    implementation 'com.android.support:recyclerview-v7:27.1.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

6. Pada kanan atas akan muncul link untuk “**Sync Now**”, tekanlah link tersebut.

PS[1]:

Pada proses ini Gradle akan mengunduh package tersebut ke dalam projek Android Anda.

PS[2]:

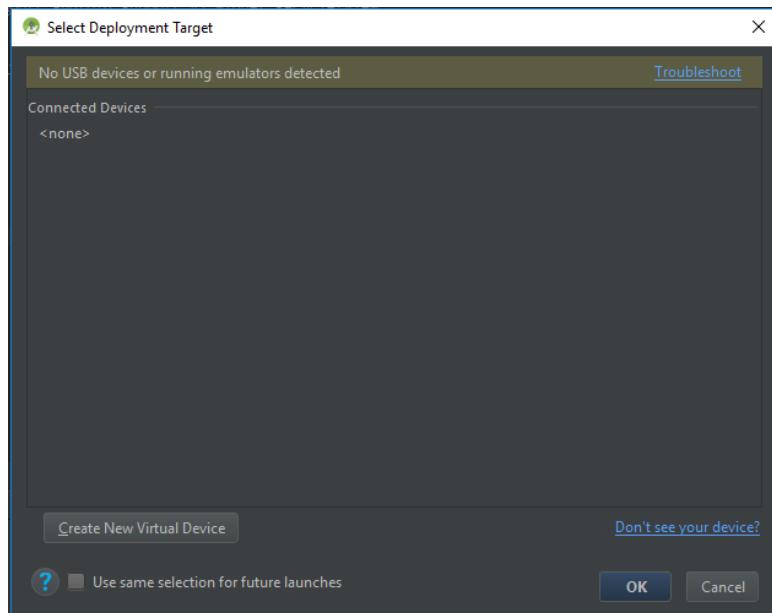
Apabila Anda memasang Android Studio dengan tahap Instalasi Mudah, package sudah disediakan sehingga Anda tidak perlu mengunduhnya lagi.

7. Bukalah kembali **app\java\[namapackage]\[namaactivity]**.
8. Lihatlah apakah masih error?

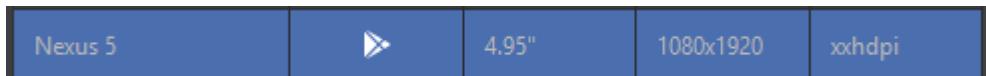
Tutorial: Create & Run Hello World

Pada tahap ini Anda akan menjalankan Aplikasi yang sudah dibuat sebelumnya dengan menggunakan Android Emulator.

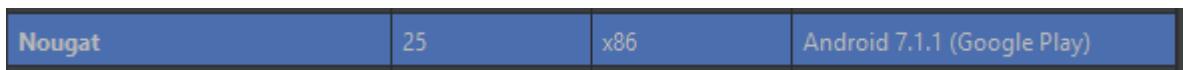
1. Bukalah tab **Run** pada Android Studio kemudian pilih **Run 'app'** (Shortcut: Shift + F10).
2. Pada tab Select Deployment Target, apabila tampilan masih kosong dan masih muncul seperti gambar berikut, tekanlah tombol **Create New Virtual Device**, apabila sudah ada, jalankan apa yang sudah ada (Android 7.1.1)



3. Pada **Select Hardware**, Pilih **Nexus 5**, kemudian **Next**



4. Pada System Image, Pilih **Nougat (API Level 25, ABI x86, Target Android 7.1.1 Google Play)**, kemudian **Next**



PS:

Apabila Anda menggunakan processor non-intel, Gunakan **Nougat (API Level 25, ABI ARM, Target Android 7.1.1)**.

5. Pada Verify Configuration, tekan **Finish** dan Akan dikembalikan pada **Select Deployment Target** kembali.

6. Pilih **Nexus 5 API 25** dan tekan tombol **OK**, kemudian seduh kopi / teh / air hangat / air dingin sembari menunggu emulator dijalankan dan program dijalankan ke dalam Android Emulator (+/- 5 menit on HDD).
7. Apabila sudah berhasil dijalankan hasil akhir dari Aplikasi Android yang sudah dibuat adalah sebagai berikut.

(Tulisan Week 01_12001 akan berbeda tergantung dari Application Name yang diketik).



8. Selamat Anda sudah berhasil menjalankan Hello World pada Android.

Tambahan:

Cobalah untuk mengganti tulisan Hello World! menjadi "[NAMA] – [NIM]" pada Aplikasi Android yang sudah Anda buat.

REFERENSI

MODUL 2

(ANDROID ACTIVITY DAN PENGATURAN LAYOUT, dan DEBUGGING)

DESKRIPSI TEMA

- Activity
- Relative, Linear, Table, Grid dan Constraint Layout
- UI Component
- Intent

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menggunakan berbagai teknik pengaturan layout untuk menghasilkan tampilan aplikasi yang rapih serta mampu melakukan proses debugging aplikasi yang telah dibuat.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Android Res Layout pada Android

Pada langkah ini Anda akan mempelajari struktur file pada program Android.

Android menyediakan pembuatan layout dengan menggunakan **xml** pada folder **app\res\layout**. Pada Langkah ini akan akan mempelajari lebih dalam mengenai struktur **xml** pada **app\res\layout**.

1. Bukalah Android Studio dan pilih **Start a new Android Studio Project**
2. Buatlah project dengan format:

Name : **Week [xx]_[NIM]**. Contoh: **Week 01_12001**.

Company Domain : **umn.ac.id**.

Project Location : **D:\FTI\IF634_IS534_MOBILE\workspaces**

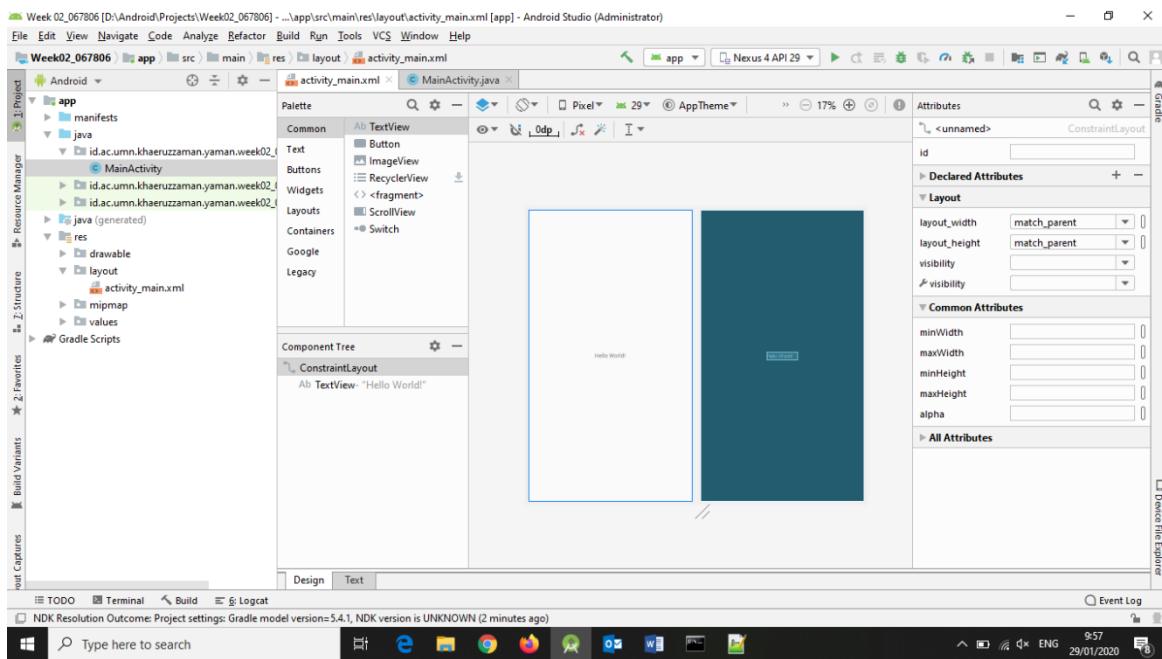
Target Android : **(Phone and Tablet) API 25: Android 7.1.1 (Nougat)**

Activity : **Empty Activity**

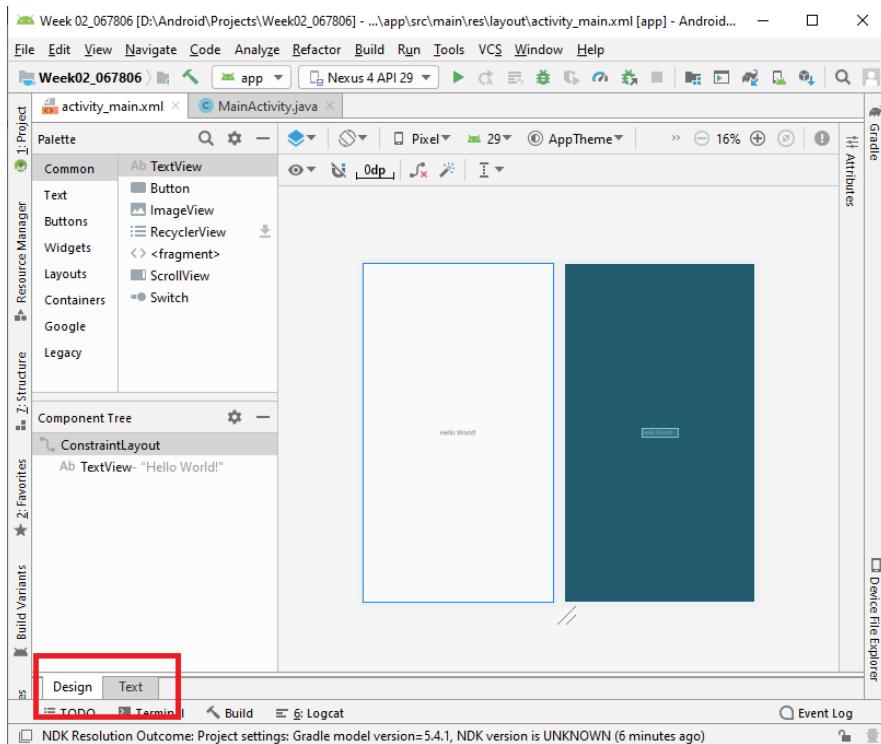
Activity Name : **FirstActivity**

Layout Name : **activity_first**

3. Setelah Gradle build finished, bukalah **activity_first.xml** yang terletak pada **app\res\layout\activity_first.xml** dan Anda akan diberikan tampilan seperti berikut:



4. Pada bagian bawah bagian tampilan xml, Anda akan diberikan 2 opsi pilihan, **Design mode** dan **Text mode**, Pada praktikum di perkuliahan, Anda akan lebih menggunakan **Text mode**, oleh karena itu, tekanlah tombol **Text**.



5. Ketika tombol **Text** ditekan, Anda akan diberikan tampilan xml seperti berikut

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Keterangan:

```
<?xml version="1.0" encoding="utf-8"?>
```

Merupakan tag "pembuka" yang menyatakan bahwa tipe file ini adalah XML.

```
<android.support.constraint.ConstraintLayout ...>
```

Merupakan tag yang menyatakan bahwa pada layout ini akan menggunakan **ConstraintLayout**.

ConstraintLayout merupakan layout *default* yang digunakan pada Android sekarang ini dikarenakan fleksibilitas dalam meletakkan UI Component lainnya di dalam Layout.

Sebelum adanya **ConstraintLayout**, layout yang memiliki fleksibilitas yang sama mengenai UI Component adalah **RelativeLayout**, namun dikarenakan Google sudah menyatakan bahwa **ConstraintLayout** lebih fleksibel dan tidak makan resource, maka pada perkuliahan yang akan lebih digunakan adalah **ConstraintLayout**, bukan **RelativeLayout**.

Namun selain **ConstraintLayout** juga ada beberapa Layout konvensional lainnya seperti:

LinearLayout merupakan layout *default* konvensional pada Android. Layout ini akan meletakkan posisi dari UI Component tergantung dari atribut **android:orientation** yang digunakan (horizontal / vertical).

Berikut adalah contoh ketika menggunakan LinearLayout sebagai Layout default (Perhatikan bahwa di dalam tag TextView sudah tidak lagi menggunakan atribut `app:layout_constraintxxx_toXxxOf="parent"` yang hanya dipergunakan ketika menggunakan ConstraintLayout)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".FirstActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:text="Hello World!" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:text="Hello World!" />

</LinearLayout>
```

Untuk mengenal lebih lanjut tentang **LinearLayout**, Anda bisa membuka tautan <https://developer.android.com/guide/topics/ui/layout/linear>

TableLayout merupakan layout yang meletakkan posisi dari UI Component tergantung dari tag **TableRow** yang didefinisikan dalam **TableLayout** itu sendiri. Penambahan dari UI Component dalam **TableRow** akan secara otomatis menambah Column dari **TableLayout** secara keseluruhan.

Berikut adalah contoh ketika menggunakan **TableLayout** sebagai layout default.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FirstActivity">

    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="2dp"
            android:text="Testing"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="2dp"
            android:text="Testing"/>
    </TableRow>

    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="2dp"
            android:text="Testing"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="2dp"
            android:text="Testing"/>
    </TableRow>
</TableLayout>
```

WARNING:

TableLayout merupakan salah satu Layout pada Android yang cukup boros dalam penggunaan resources. Apabila menggunakan **TableLayout** untuk data yang cukup besar (> 30 row) mohon pergunakan layout lainnya.

Untuk mengenal lebih lanjut tentang **TableLayout**, Anda bisa membuka tautan <https://developer.android.com/reference/android/widget/TableLayout>

GridLayout merupakan layout yang meletakkan posisi dari UI Component tergantung dari jumlah *row* dan *column* yang didefinisikan dengan menggunakan atribut **android:rowCount** dan **android:columnCount**.

Penggunaan **GridLayout** apabila digabungkan dengan *Adapter-based Component*, misal **CardView** (tidak dipelajari pada perkuliahan minggu ini) merupakan hal yang standar dalam Pemrograman Android.

Berikut adalah contoh ketika menggunakan **GridLayout** sederhana sebagai layout default.

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:rowCount="2"
    android:columnCount="2"
    tools:context=".FirstActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:text="Testing"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:text="Testing"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:text="Testing"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:text="Testing"/>
</GridLayout>
```

Untuk mengenal lebih lanjut tentang **GridLayout**, Anda bisa membuka tautan <https://developer.android.com/reference/android/widget/GridLayout>

Tutorial: UI Component

UI Component merupakan element penting dalam Android yang dapat digunakan untuk interaksi user terhadap aplikasi yang dibuat.

Common UI Component yang ada di dalam Android adalah **Button**, **TextView**, **EditText** **ImageView**, **RecyclerView** (*adapter-based*), dll. Pada Tutorial ini Anda akan membuat UI Component.

1. Gunakan **activity_first.xml** pada tutorial
2. Gunakan **LinearLayout** sebagai default layout dengan **orientation** Vertical
3. Tambahkan kode berikut ke dalam **LinearLayout**



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textView"
        android:layout_width="100dp"
        android:layout_height="33dp"
        android:text="Angka 1:" />
    <EditText
        android:id="@+id/angka1"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:hint="@string/angka1"
        android:inputType="numberDecimal" />
</LinearLayout>
```

LinearLayout ini ada di dalam **LinearLayout** default yang sudah dibuat, sehingga menjadi **Nested LinearLayout**.

Keterangan:

android:layout_width & android:layout_height

mendeskripsikan lebar dari komponen / layout yang dibuat, bisa dalam bentuk "match_parent" di mana lebar/tinggi UI Component akan disamakan dengan UI Component sebelumnya, "wrap_content" lebar/tinggi UI Component akan disamakan dengan isi (konten text / gambar), atau dengan ukuran manual, seperti **100sp**, **4dp**, **12px**, **10pt**.

Untuk pengertian lanjutan mengenai ukuran, bisa dibaca pada tautan <https://developer.android.com/guide/topics/resources/more-resources#Dimension>

android:id

mendeskripsikan id dari UI Component yang akan digunakan agar dapat dimodifikasi pada saat Programming. Selalu menambahkan "@+id/" sebagai penanda bahwa id yang dibuat adalah id kustom yang akan digunakan pada aplikasi yang dibuat.

android:hint

mendeskripsikan tulisan pada EditText pada saat input belum dimasukkan.

android:inputType

mendeskripsikan tipe input yang boleh digunakan pada EditText.

4. Tambahkan juga kode berikut ke dalam **activity_first.xml** setelah `</LinearLayout>` pada langkah sebelumnya

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="100dp"
        android:layout_height="33dp"
        android:text="Angka 2:" />
    <EditText
        android:id="@+id/angka2"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:hint="@string/angka2"
        android:inputType="numberDecimal" />
</LinearLayout>

```

5. Tambahkan juga kode berikut ke dalam **activity_first.xml** setelah `</LinearLayout>` pada langkah sebelumnya, untuk menambahkan empat buah button dengan menggunakan TableLayout:

```

<TableLayout
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
        <Button
            android:id="@+id/btnTambah"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Tambah" />
        <Button
            android:id="@+id/btnKurang"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Kurangi" />
    </TableRow>

```

```

<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <Button
            android:id="@+id	btnKali"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Kali" />
        <Button
            android:id="@+id	btnBagi"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Bagi" />
    </TableRow>
</TableLayout>

```

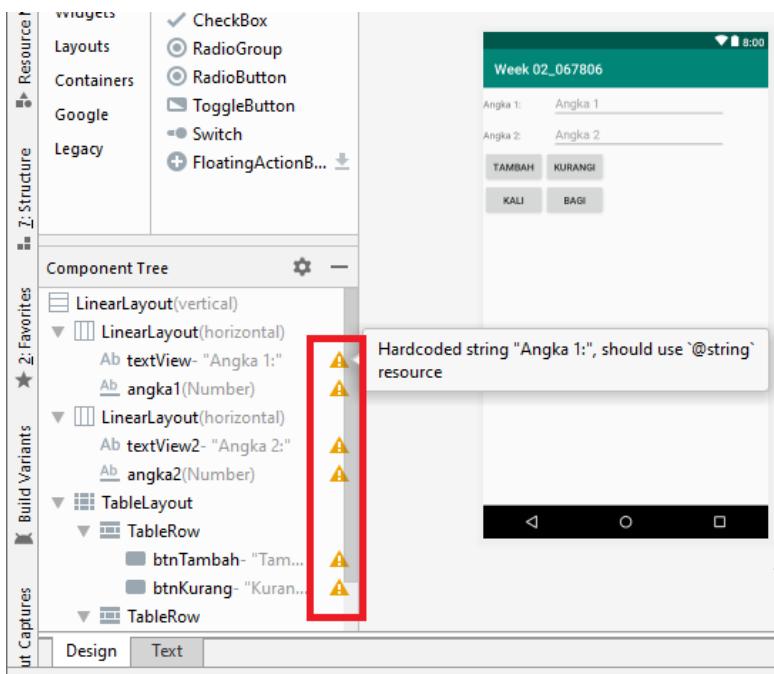
6. Tambahkan juga kode berikut ke dalam `activity_main.xml` setelah `</TableLayout>` pada langkah sebelumnya, untuk menambahkan `TextView` sebagai hasil dari operasi di atas:

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textView3"
        android:layout_width="100dp"
        android:layout_height="33dp"
        android:text="Hasil :" />
    | 
    <TextView
        android:id="@+id/hasil"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:text="" />
</LinearLayout>

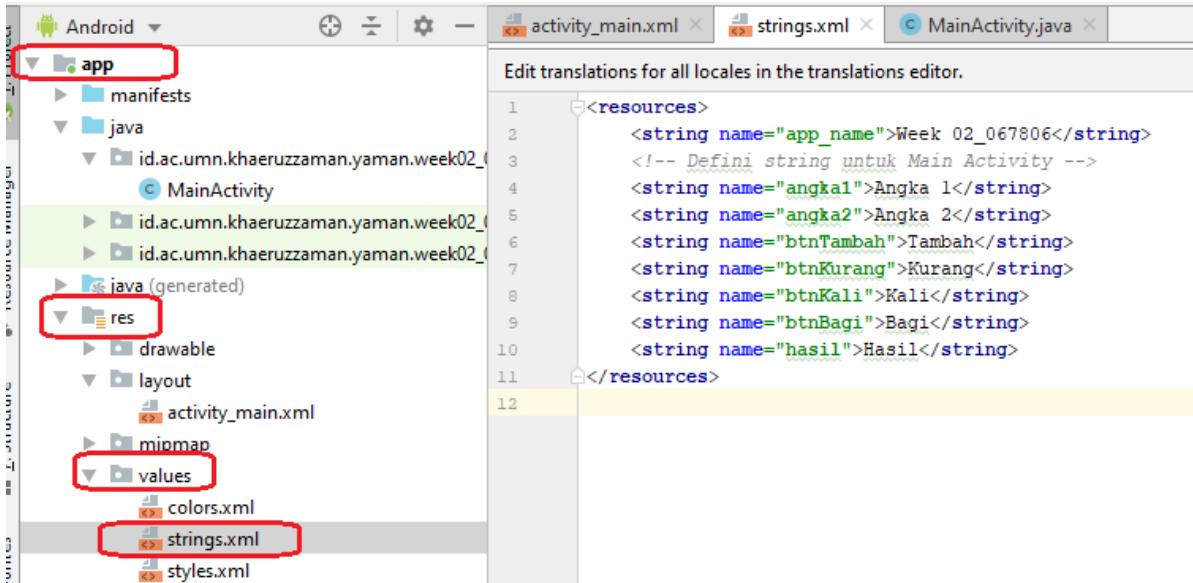
```

7. Perhatikan pada atribut `android:text` dan `android:hint` masih di-*highlight* berwarna kuning. Hal ini dikarenakan kita masih menulis **String** yang ada secara langsung pada atribut tersebut.



Best practice untuk menggunakan **String** statis (yang tidak akan dimodifikasi tulisannya) seperti ini adalah dengan menggunakan resource string yang terletak pada **app\res\values\strings.xml**.

Bukalah **app\res\values\strings.xml** dan masukkan kode berikut ke dalamnya.



8. Setelah menambahkan kode di atas, gantilah seluruh **android:hint** dan **android:text** yang ada dengan menggunakan String yang sudah didefinisikan dengan menggunakan "@string/[namastring]", misal **android:hint="@string/first_activity_nama"**.

INFO:

Memang akan terkesan lebih panjang dan susah pada saat pemberian nama dengan menggunakan **strings.xml** seperti ini. Hanya saja akan sangat membantu ketika Anda akan menggunakan konsep i18n (internationalization).

9. Bukalah **app\java\[namapackage]\MainActivity** yang sudah dibuat secara otomatis oleh Android Studio pada saat pembuatan Activity dan masukkan kode berikut

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

10. Tambahkanlah beberapa baris kode untuk dapat menghubungkan definisi-definisi komponen UI yang ada pada layout **activity_main.xml** dengan kode Java pada **MainActivity.java** seperti terlihat pada gambar berikut:

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView; } Penambahan Class-Class yang harus diimport

public class MainActivity extends AppCompatActivity {
    EditText angka1, angka2; } Deklarasi object (variable) yang merepresentasikan komponen UI
    TextView hasil;
    Button btnTambah, btnKurang, btnKali, btnBagi; }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); }
        setContentView(R.layout.activity_main);
        angka1 = (EditText) this.findViewById(R.id.angka1);
        angka2 = (EditText) this.findViewById(R.id.angka2);
        hasil = (TextView) this.findViewById(R.id.hasil);
        btnTambah = (Button) this.findViewById(R.id.btnTambah);
        btnKurang = (Button) this.findViewById(R.id.btnKurang);
        btnKali = (Button) this.findViewById(R.id.btnKali);
        btnBagi = (Button) this.findViewById(R.id.btnBagi); }

}
  
```

} Menghubungkan object (variable) pada code Java dengan object komponen UI pada layout .xml file

Keterangan:

Angka1 = (EditText) findViewById(R.id.angka1);

Line ini digunakan untuk mencari EditText yang sudah didefinisikan pada **activity_main.xml**. Akan mencari id dengan nama `@+id/angka1` pada seluruh xml layout yang ada.

11. Untuk menambahkan interaksi terhadap button-button yang ada, maka kita harus tambahkan **OnClickListener** terhadap masing-masing button di atas. Tambahkan kode dibawah ini setelah baris `btnBagi = ...:`

```

    btnTambah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { hitung( operator: '+' ); } } );
    btnKurang.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { hitung( operator: '-' ); } } );
    btnKali.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { hitung( operator: '*' ); } } );
    btnBagi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { hitung( operator: '/' ); } } );

btnTambah.setOnClickListener(...);

```

Line ini digunakan untuk menambahkan event **OnClick** pada Button `btnTambah`.

12. Selanjutnya untuk melengkapi interaksi yang dilakukan pada saat button-button tersebut di-click, di mana pada method **OnClick(View v)** memanggil method umum menghitung angka1 dengan angka2 dengan nama **hitung**. Tambahkan implementasi kode method **hitung** dibawah ini sebelum "}" – kurung kurawal tutup – sebagai penutup block Class **MainActivity**.

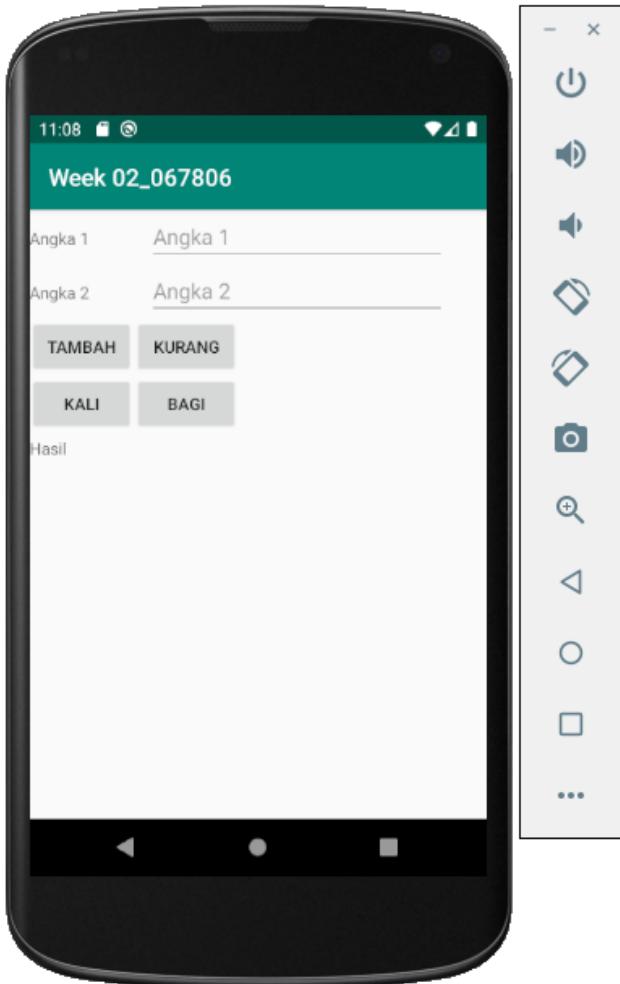
```

protected void hitung(char operator){
    double operand1 = Double.parseDouble(angka1.getText().toString());
    double operand2 = Double.parseDouble(angka2.getText().toString());
    double result = 0.0;
    switch(operator){
        case('+') : result = operand1 + operand2; break;
        case('-') : result = operand1 - operand2; break;
        case('*') : result = operand1 * operand2; break;
        case('/') : result = operand1 / operand2;
    }
    hasil.setText(String.valueOf(result));
}

```

13. Terakhir, pastikan error dan warning pada program kita baik yang java dan yang xml sudah tidak ada.

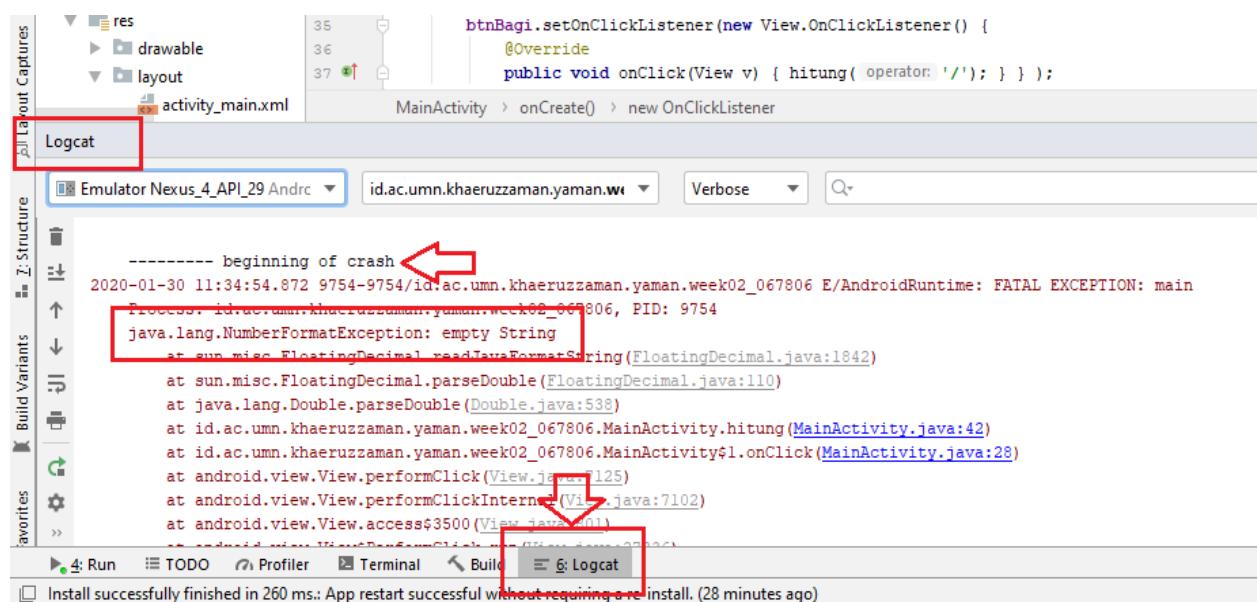
Jalankan project yang telah kita buat dengan cara menu “Run” -> “Run ‘app’”.



Tutorial: Debugging

Debugging adalah suatu proses untuk mencari dan memperbaiki kesalahan, error, bugs dan / atau behavior (tingkah laku) yang tidak diharapkan pada program kita. Semua program umumnya mempunyai bugs, mulai dari behavior yang tidak benar dalam app kita, sampai ke behavior yang menggunakan memory yang sangat banyak atau memakai resources jaringan yang tinggi bahkan sampai app kita betul-betul freez (tidak melakukan apa-apa).

Seperti app kita yang telah kita buat di atas belum terbebas dari bugs. Coba lakukan test terhadap app kita, misalnya setelah app jalan langsung klik salah satu button, app kita akan “crash” atau dengan kata lain, salah satu dari isian (angka 1 atau angka 2) kita kosongkan lalu kita click salah satu button maka akan terjadi “run time error” dengan menghasilkan **NumberFormatException**. Penjelasan error atau exception dapat dilihat pada “logcat” window seperti gambar berikut:



Untuk mengetahui secara detail behavior dari app kita pada saat dijalankan, Android Studio menyediakan tool yang disebut Debugger yang berfungsi sebagai alat untuk debugging. Berikut adalah langkah-langkah untuk melakukan debugging.

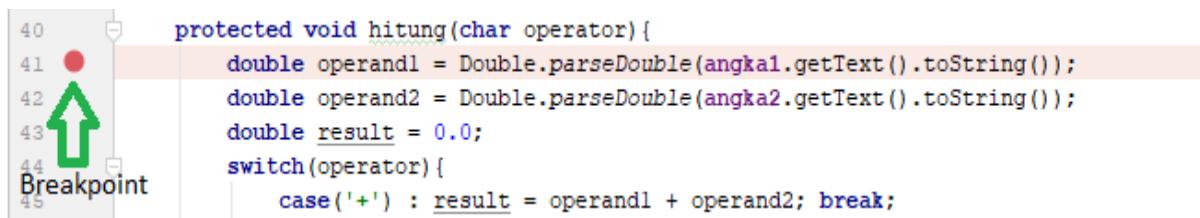
1. Menjalankan App pada mode “debug”. Untuk melakukan debugging kita harus menjalankan app kita pada mode debug yaitu dengan cara pilih menu “Run” → “Debug ‘app’”. Jika app kita sedang berjalan baik pada emulator maupun pada perangkat smartphone, kita akan ditanya “apakah kita ingin restart app kita pada mode debug?”. Klik **Restart app**.

Pada proses debugging, Android studio juga melakukan build dan run app kita di emulator atau perangkat yang terhubung. Pada saat debugger di initialisasi, pada perangkat kita akan muncul

tulisan "Waiting for Debugger" sebelum app dijalankan.

Untuk melihat panel debug klik tab debug di area bawah Android Studio (atau pilih menu "View ➔ Tool Windows ➔ Debug").

2. Tentukan **breakpoint**. Breakpoint adalah tempat dimana kita menginginkan eksekusi terhadap app kita berhenti sejenak (jeda atau pause) untuk tujuan memeriksa dan mengevaluasi variable serta isinya, menjalankan kode baris per baris untuk mengetahui penyebab terjadi error pada saat program dijalankan (runtime error). Untuk menempatkan sebuah breakpoint di kode Java kita. Klik sebelah kanan nomor baris, maka akan ada lingkaran merah pertanda baris tersebut menjadi breakpoint. Untuk menghapus breakpoint, klik ulang lingkaran merah tersebut maka breakpoint akan terhapus.

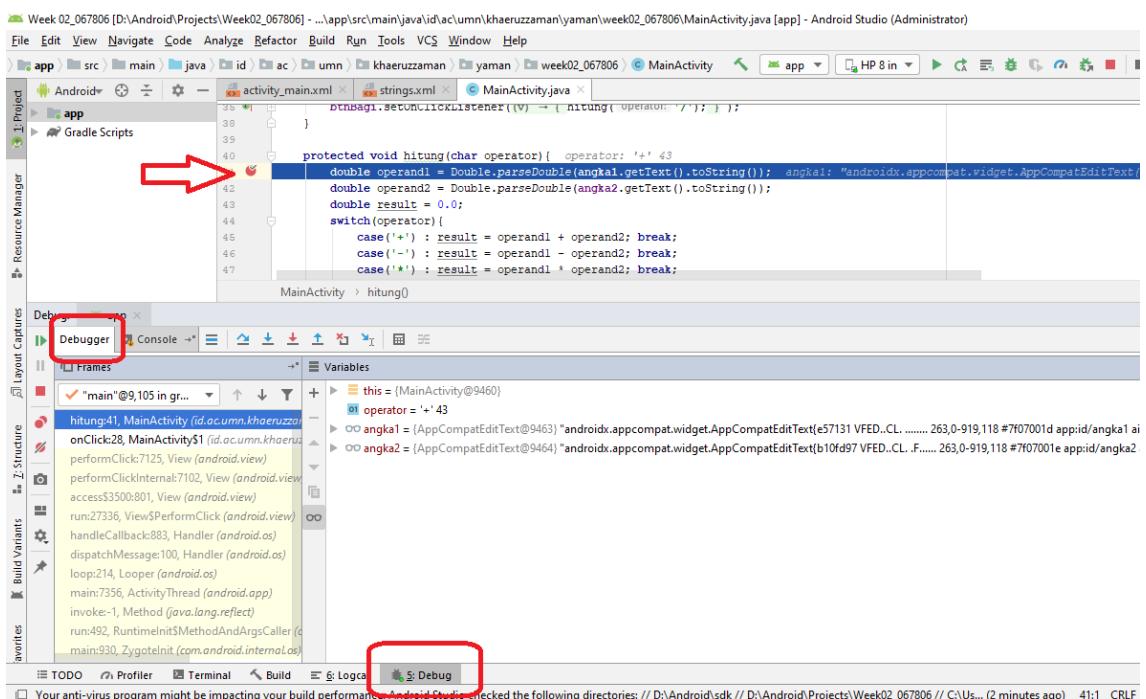


```

protected void hitung(char operator){
    double operand1 = Double.parseDouble(angka1.getText().toString());
    double operand2 = Double.parseDouble(angka2.getText().toString());
    double result = 0.0;
    switch(operator) {
        case('+') : result = operand1 + operand2; break;

```

3. Menjalankan app langkah demi langkah. Pada saat app dijalankan pada mode debug dan mencapai breakpoint, maka eksekusi app akan dijeda (diberhentikan sementara). Contoh jika kita jalankan app yang diatas dengan mode debug dengan meletakkan sebuah breakpoint pada baris nomor 41, maka pada saat kita meng-klik salah satu button, maka eksekusi akan berhenti sementara pada saat eksekusi mencapai baris 41 tersebut seperti pada gambar berikut:



Untuk meneruskan eksekusi ada beberapa pilihan:

- a. Step Over (**Run ➔ Step Over**). Debugger menjalankan baris dimana ia berhenti lalu menuju baris berikutnya dan berhenti sementara juga.
- b. Step Into (**Run ➔ Step Into**). Jika pada baris tersebut (dimana ia berhenti) memanggil sebuah method, maka debugger akan masuk kedalam method yang dipanggil.
- c. Step Out (**Run ➔ Step Out**). Debugger akan menyelesaikan sampai akhir dari blok atau akhir dari method dan kembali ke baris semula dimana kita lakukan Step Into.
- d. Resume (**Run ➔ Resume Program**).

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project anda dengan nama **Week02_[NIM]** sudah dapat dijalankan dan berhasil melakukan perhitungan yang benar ketika button Tambah, Kurang, Kali, dan Bagi. Copy paste file berikut ke isian tugas pada eLearning:
 - a. MainActivity.java
 - b. activity_main.xml
 - c. string.xml
 - d. manifest.xml
2. Lakukan pengujian terhadap app anda dengan skenario di bawah ini:
 - a. Kosongkan Angka 1 dan Angka 2, lalu klik salah satu button
 - b. Kosongkan Angka 2, lalu klik button Bagi.
 - c. Isi 0 (nol) pada Agka 2, lalu klik button Bagi.

Tuliskan penjelasan apa yang terjadi, jika terjadi error jelaskan apa error nya dan kemungkinan penyebabnya, serta langkah-langkah apa yang dapat memperbaiki error tersebut untuk masing-masing skenario di atas. Submit Tulisan anda melalui isian pada eLearning.

3. Lakukan debugging pada program anda dengan meletakkan dua breakpoint bebas letaknya. Gunakan langkah-langkah step-over, step-into, step-out, dan resume. Tuliskan kesimpulan hasil debugging anda dan submit ke isian assignment di eLearning.

REFERENSI

MODUL 3

(RECYCLER VIEW, FLOATING ACTION BUTTON, DAN TOAST)

DESKRIPSI TEMA

- RecyclerView
- Floating Action Button (FAB)
- Toast

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menggunakan berbagai komponen user-interface dan user-interaction untuk menghasilkan tampilan aplikasi dan membuat interaksi antartampilan. Mahasiswa mampu menggunakan RecyclerView untuk menampilkan list (atau daftar), menggunakan Floating Action Button (FAB) untuk menambah item ke dalam list, serta Toast untuk menampilkan item pada list yang di-klik oleh user.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Interaksi antar RecyclerView, Floating Action Button, dan Toast

Pada langkah ini Anda akan mempelajari penggunaan RecyclerView untuk menampilkan list (atau daftar), penggunaan Floating Action Button (FAB) untuk menambah item ke dalam list, serta Toast untuk menampilkan item pada list yang diklik oleh user.

RecyclerView adalah sebuah sub-Class dari **ViewGroup** yang lebih efisien dalam penggunaan resources untuk menampilkan *scrollable* list. RecyclerView hanya membuat item-item dari sebuah list dalam jumlah terbatas dan akan menggunakan ulang (reuse) untuk menampilkan isi yang terlihat saja (visible content).

Bukalah Android Studio dan pilih **Start a new Android Studio Project**

1. Buatlah project dengan format:

Name	: Week [xx]_[NIM]. Contoh: Week 03_12001.
Company Domain	: umn.ac.id.
Project Location	: D:\FTI\IF634_IS534_MOBILE\workspaces\
Target Android	: (Phone and Tablet) API 25: Android 7.1.1 (Nougat)
Activity	: Empty Activity

Activity Name : **MainActivity**
 Layout Name : **activity_main**

2. Penggunaan **Basic Activity** template akan menyediakan sebuah a floating action button (FAB) dan app bar pada Activity layout (**activity_main.xml**), dan sebuah layout untuk Activity content (**content_main.xml**). Setelah Gradle build finished, bukalah folder **app\res\layout** pastikan terdapat dua layout file: **activity_main.xml** dan **content_main.xml**.
3. Pembuatan Data set untuk ditampilkan pada RecyclerView berupa daftar kata ["Kata 1", "Kata 2", "Kata 3", , "Kata 20"]. Buka file **MainActivity.java** dan tambahkan sebuah private member variable dengan tipe data **LinkedList<String>** dengan nama **mDaftarKata**. Dengan cara tambahkan "import statement" sebelum **public Class MainActivity** Kemudian tambahkan deklarasi variable **mDaftarKata** setelah **public Class MainActivity**, seperti terlihat pada potongan kode di bawah ini:

```
import java.util.LinkedList;
public class MainActivity extends AppCompatActivity {

    private final LinkedList<String> mDaftarKata = new
    LinkedList<>();

    @Override
```

Setelah itu tambahkan kode di bawah ini untuk mengisi variable **mDaftarKata** tersebut yang diletakkan pada method **onCreate(...)** sebelum "}" tutup kurung kurawal (sebelum akhir dari method).

```
for (int i = 1; i < 21; i++) {
    mDaftarKata.add("Kata " + i);
}
} // penutup method onCreate()
```

4. Mengganti ikon FAB, default ikon dari FAB adalah ikon kirim e-mail. Untuk keperluan tutorial ini, ikon yang digunakan adalah ikon "+". Buka file **activity_main.xml**, ganti

```
app:srcCompat="@android:drawable/ic_dialog_email"/>
dengan
app:srcCompat="@android:drawable/ic_menu_add"/>
```

- Menambahkan **RecyclerView** pada layout. Buka file **content_main.xml** dengan mode **Text**. Ganti seluruh **TextView** element dengan kode berikut:

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

- Membuat layout untuk menampilkan masing-masing item di dalam list. Class **Adapter** membutuhkan sebuah layout untuk menampilkan sebuah item pada list. Semua item dalam list akan menggunakan layout ini. Untuk tutorial ini, kita gunakan sebuah layout sederhana menggunakan sebuah **LinearLayout Vertikal** dengan sebuah **TextView**. Klik kanan pada folder **app > res > layout**, dan pilih **New > Layout resource file**. Setelah jendela **New Resource File** muncul, isi nama file dengan **daftarkata_item** dan Root Element dengan **LinearLayout**. Lalu klik **OK** untuk konfirmasi dan menutup jendela tersebut.
- Buka file layout **daftarkata_item.xml** menggunakan mode **Text**, dan ganti seluruh kodennya dengan kode berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="6dp">

    <TextView
        android:id="@+id/kata"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="24sp"/>
</LinearLayout>
```

8. Membuat Custom Class **Adapter** yang akan menghubungkan data dengan view di dalam list. Klik kanan pada folder **app > java > [domain].week03_[NIM]** kemudian pilih **New > Java Class**. Beri nama Class dengan **DaftarKataAdapter.java**. Extend sebuah adapter generic untuk RecyclerView sehingga dapat menggunakan View holder yang diperuntukan buat app yang dibuat di dalam class adapter **DaftarKataAdapter**. Sehingga kelas tersebut memiliki signature sebagai berikut:

```
public class DaftarKataAdapter extends  
RecyclerView.Adapter<DaftarKataAdapter.KataViewHolder> {  
  
}
```

Inner class **KataViewHolder** akan menampilkan error, karena class tersebut belum dideklarasikan dan didefinisikan. Untuk memperbaiki error tersebut, arahkan kursor pada kata **KataViewHolder** klik kana pada "lampa error warna merah" atau tekan "Alt+Enter" pilih **Implement methods**. Maka method-method yang wajib diimplementasikan seperti: **onCreateViewHolder**, **getItemCount**, dan **onBindViewHolder** akan disisipkan kedalam kelas tersebut dengan placeholder yang masih kosong.

Sehingga class DaftarKataAdapter memiliki penggalan kode seperti berikut:

```
import android.view.ViewGroup;  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
public class DaftarKataAdapter extends  
RecyclerView.Adapter<DaftarKataAdapter.KataViewHolder> {  
    @NonNull  
    @Override  
    public DaftarKataAdapter.KataViewHolder onCreateViewHolder(  
        @NonNull ViewGroup parent, int viewType) {  
        return null;  
    }  
    @Override  
    public void onBindViewHolder(  
        @NonNull DaftarKataAdapter.KataViewHolder holder,  
        int position) {  
    }  
    @Override  
    public int getItemCount() {  
        return 0;  
    }  
}
```

9. Selanjutnya buat View Holder Adapter dengan nama yang telah kita sebutkan sebelumnya: **KataViewHolder** di dalam kelas **DaftarKelasAdapter**. Buatlah class inner KataViewHolder di dalam class. DaftarKataAdapter yang meng-extends class **RecyclerView.ViewHolder**, serta tambahkan variable dengan tipe data **TextView** dan **Adapter**. Untuk melengkapi class tersebut kita harus sediakan sebuah konstruktor untuk class tersebut agar bisa di-initialisasi. Class inner KataViewHolder akan memiliki setidaknya kode sebagai berikut:

```
class KataViewHolder extends RecyclerView.ViewHolder{
    public final TextView kataItemView;
    final DaftarKataAdapter mAdapter;

    public KataViewHolder(@NonNull View itemView,
                         DaftarKataAdapter adapter) {
        super(itemView);
        kataItemView = itemView.findViewById(R.id.kata);
        this.mAdapter = adapter;
    }

}
```

10. Simpan data daftar kata-kata ke dalam Adapter. Untuk menampung data di dalam adapter dibutuhkan sebuah linkedList String dengan nama **mDaftarKata**. Sementara untuk membuat sebuah view untuk sebuah item list dibutuhkan layout inflater (tipe data **LayoutInflater**) yang membaca sebuah XML descriptor dan mengkonversi menjadi sebuah view item. Kedua variable di atas harus disiapkan konstruktor pada class **DaftarKataAdapter**. Berikut petikan kode deklarasi dan konstruktor tersebut:

```
public class DaftarKataAdapter extends
    RecyclerView.Adapter<DaftarKataAdapter.KataViewHolder> {

    private final LinkedList<String> mDaftarKata;
    private LayoutInflater mInflater;

    DaftarKataAdapter(Context context, LinkedList<String>
        daftarKata) {
        mInflater = LayoutInflater.from(context);
        mDaftarKata = daftarKata;
    }

    @NonNull
    //
```

11. Berikut adalah meng-update dan mengimplementasikan method-method yang sebelumnya masih memiliki placeholder yang kosong. Untuk `getItemCount()` kita tambahkan return `mDaftarKata.size()` untuk memberikan jumlah daftar kata pada list secara dinamis, untuk method `onCreateViewHolder(...)` kita tambahkan proses konversi dan memberikan hasil berupa View holder, serta untuk method `onBindViewHolder(...)` mengisikan konten ke dalam view holder. Sehingga kode lengkap ketiga method tersebut dapat dilihat di bawah ini:

```

@NonNull
@Override
public DaftarKataAdapter.KataViewHolder onCreateViewHolder(
    @NonNull ViewGroup parent, int viewType) {
    View mItemView = mInflater.inflate(R.layout.daftarkata_item,
        parent, false);
    return new KataViewHolder(mItemView, this);
}

@Override
public void onBindViewHolder(@NonNull
    DaftarKataAdapter.KataViewHolder holder, int position) {
    String mCurrent = mDaftarKata.get(position);
    holder.kataItemView.setText(mCurrent);
}

@Override
public int getItemCount() {
    return mDaftarKata.size();
}

```

12. Sampai langkah ini, kita sudah memiliki view untuk daftar kata, adapter untuk menyambungkan view dengan data yang telah dilengkapi dengan view holdernya. Langkah selanjutnya adalah membuat RecyclerView pada Activity kita (**MainActivity.java**). Buka file **MainActivity.java**, tambahkan beberapa kode untuk langkah-langkahnya. Tambahkan object **RecyckerView** dan Adapter (**DaftarKataAdapter**).

```

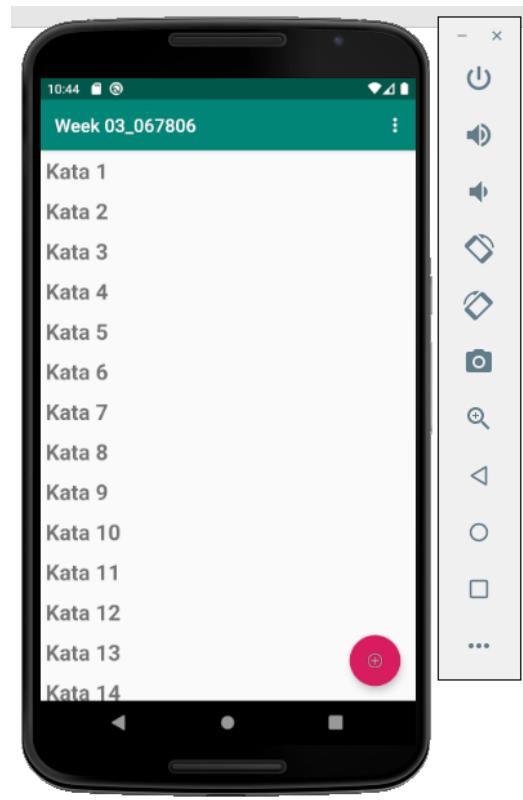
private RecyclerView mRecyclerView;
private DaftarKataAdapter mAdapter;

```

Kemudian untuk method `onCreate(...)` tambahkan penyambungan object RecyclerView dengan object yang ada pada layout file, inisialisasi `DaftarKataAdapter`, jadikan adapter tersebut sebagai default adapter untuk object `RecyclerView` dan set Layout manager `RecyclerView` object dengan Linear Layout.

```
mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
    mAdapter = new DaftarKataAdapter(this, mDaftarKata);
    mRecyclerView.setAdapter(mAdapter);
    mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
} // penutup method onCreate()
```

Pada tahap ini, jika tidak ada error yang terjadi, app anda sudah dapat dijalankan dengan tampilan seperti pada gambar di bawah ini:



13. App yang dihasilkan masih belum terdapat interaksi yang signifikan selain hanya daftar kata yang dapat di-scroll ke atas dan ke bawah. Tugas selanjutnya adalah bagaimana app ini dapat ditambahkan interaksi yang sesuai yang kita harapkan. Pertama, jika kita klik pada elemen (atau salah satu item) pada daftar di recyclerview. Kita tambahkan interaksi sederhana dengan meng-ubah "**Kata X**" dengan "**Kata X pernah diklik**" dan menampilkan **Android Toast** (An Android Toast adalah sebuah pesan yang akan muncul pada layer serupa dengan ToolTips atau notifikasi pop-up. Sebuah

Toast ditampilkan di atas konten utama dari sebuah Activity dan akan tampil hanya pada waktu yang pendek jika kita meng-klik atau memilih item / kata tersebut. Buka file **DaftarKataAdapter.java**, tambahkan “**implements View.OnClickListener**” pada signature class inner **KataViewHolder**, seperti terlihat di bawah ini:

```
class KataViewHolder extends RecyclerView.ViewHolder
    implements View.OnClickListener{
```

Karena interface **View.OnClickListener** wajibkan kita untuk mengimplementasi method **onClick()**. Implementasi method **onClick()** kita buat sederhana dan pada konstruktor perlu ditambahkan method **setOnItemClickListener()** untuk **itemView** sehingga **RecyclerView** object yang ada dapat mendengar dan merespon interaksi yang diterima dari user. Berikut tambahan kode pada konstruktor dan **onClick(...)** method.

```
public KataViewHolder(@NonNull View itemView, DaftarKataAdapter
adapter) {
    super(itemView);
    kataItemView = itemView.findViewById(R.id.kata);
    this.mAdapter = adapter;
    itemView.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    int mPosition = getLayoutPosition();
    String element = mDaftarKata.get(mPosition);
    mDaftarKata.set(mPosition, element + " pernah diklik ");
    mAdapter.notifyDataSetChanged();
    Toast.makeText(v.getContext(),
        element +" dipilih",
        Toast.LENGTH_LONG).show();
}
```

14. Selanjutnya adalah penambahan interaksi penambahan kata baru pada daftar kata di recyclerView object, jika user klik pada Floating Action Button (FAB). Default action yang disediakan ada saat kita menggunakan template “**BasicActivity**” dan klik pada FAB adalah menampilkan Snackbar. Untuk latihan ini, kita akan mengganti reaksi yang akan dilakukan oleh app jika user klik FAB, yaitu menambah sebuah anggota baru dengan kata: “**Kata X telah ditambahkan**” dan disisipkan pada akhir daftar. Untuk melakukan hal ini, buka file **MainActivity.java**, lalu ganti implementasi method **onClick(View view)** yang ada pada implementasi **fab.setOnClickListener(new ...)** yang semula memanggil **Snackbar.make(view)** menjadi sebagai berikut:

```
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int jumlahKata = mDaftarKata.size();
        mDaftarKata.addLast("Kata " + (jumlahKata + 1) +
            " telah ditambahkan");
        Objects.requireNonNull
        (mRecyclerView.getAdapter()).notifyItemInserted(jumlahKata);
        mRecyclerView.smoothScrollToPosition(jumlahKata);
    }
});
```

15. Demikian Tutorial pengembangan sebuah Android app dengan memanfaatkan RecyclerView, Floating Action Button (FAB) dan Android Toast.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project anda dengan nama **Week03_[NIM]** sudah dapat dijalankan dan berhasil menampilkan daftar kata pada layer menggunakan RecyclerView, item yang di-klik pada daftar kata ter-update isinya, serta daftar kata pada recyclerView bertambah jika user klik Floating Action Button (FAB). Copy paste file berikut ke isian tugas pada eLearning:
 - a. **MainActivity.java**
 - b. **DaftarKataAdapter.java**
 - c. **activity_main.xml**
 - d. **content_main.xml**
 - e. **daftarkata_item.xml**
 - f. **string.xml**
 - g. **manifest.xml**

2. Lakukan pengujian terhadap app anda dengan scenario beberapa penambahan dan pemilihan beberapa item pada daftar kata di recyclerView. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 4

(INTENT, ACTIVITY, FRAGMENT, DAN PASSING DATA ANTAR INTENT)

DESKRIPSI TEMA

- Intent dan Passing data antar Intent
- Activity dan Lifecyclenya
- Fragment dan penggunaannya

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menggunakan Activity dan Fragment serta melakukan pengiriman dan penerimaan data antar Activity dan/atau Fragment menggunakan class Intent.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Intent dan Passing Data antar Intent

Secara sederhana, Intent merupakan cara pada Android untuk berpindah halaman (Activity/Fragment). Pada tutorial ini Anda akan mempelajari cara berpindah halaman dengan menggunakan Intent (baik explicit maupun Implicit) dan mem-passing data dari satu Activity ke Activity lainnya.

Bukalah Android Studio dan pilih **Start a new Android Studio Project**

1. Buatlah project dengan format:

Name : Week [xxA]_[NIM]. Contoh: Week 04a_12001.

Company Domain : umn.ac.id.

Project Location : D:\FTI\IF634_IS534_MOBILE\workspaces\

Target Android : (Phone and Tablet) API 25: Android 7.1.1 (Nougat)

Activity : Empty Activity

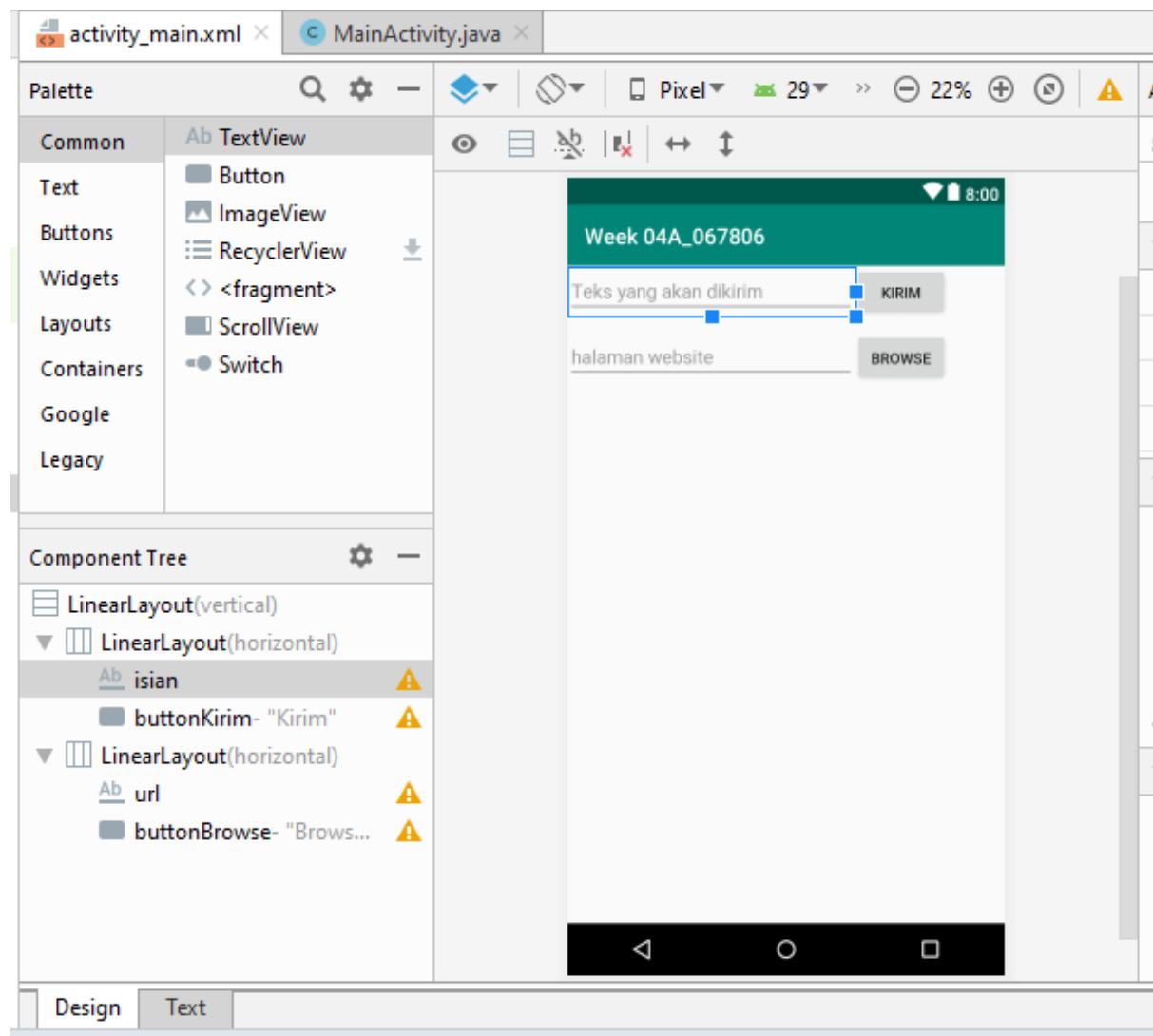
Activity Name : MainActivity

Layout Name : activity_main

2. Bukalah dan edit file layout **app > rest > layout activity_main.xml**. Gunakan **LinearLayout** (Vertical) sebagai root layout manager, serta

(Horizontal) sebagai sub layout yang menampilkan **EditText** dan **Button**. Baris pertama akan

digunakan untuk memanfaatkan passing data menggunakan **Intent Explicit** yang akan membuka Activity berikutnya dan baris kedua akan digunakan untuk memanfaatkan passing data menggunakan **Intent Implicit** yang akan memanggil browser untuk menampilkan halaman yang diinputkan user via **EditText**. Gunakan “Desain” mode sehingga tampilan seperti pada gambar berikut:



Atau gunakan “Text” mode sehingga file **activity_main.xml** seperti terlihat pada kode sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```

```
<EditText
    android:id="@+id/isian"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:hint="Teks yang akan dikirim" />
<Button
    android:id="@+id/buttonKirim"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Kirim" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="horizontal">
<EditText
    android:id="@+id/url"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:hint="halaman website" />
<Button
    android:id="@+id/buttonBrowse"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Browse" />
</LinearLayout>
</LinearLayout>
```

3. Buka dan edit file **MainActivity.java** untuk dapat berinteraksi dengan object-object yang ada pada layout di atas. Deklarasikan dua **EditText** object dengan nama **etIsian** dan **etUrl** dan dua **Button** object dengan nama **btnKirim** dan **btnBrowse**. Tambahkan juga pada **onCreate(...)** method untuk tersambung dengan object pada layout object untuk masing-masing View object di atas. Kode pada **MainActivity.java** akan terlihat seperti berikut:

```
public class MainActivity extends AppCompatActivity {
    private EditText etIsian, etUrl;
    private Button btnKirim, btnBrowse;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etIsian = findViewById(R.id.isian);
        etUrl = findViewById(R.id.url);
        btnBrowse = findViewById(R.id.buttonBrowse);
        btnKirim = findViewById(R.id.buttonKirim);
    }
}
```

4. Selanjutnya kita akan eksperimen dengan passing data menggunakan **Implicit Intent**, di mana ketika user klik tombol **Browse**, maka app anda akan mencari dan membuka web browser yang telah didefinisikan oleh Android Runtime. Jika isian Halaman Website kosong, app akan tetap membuka web browser dengan default website UMN (<http://www.umn.ac.id>). Tambahkan kode berikut pada method **onCreate(...)** melanjutkan langkah di atas:

```
btnBrowse.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String urlText = etUrl.getText().toString();
        if(urlText.isEmpty()){
            urlText = "http://www.umn.ac.id/";
        }
        Intent browseIntent = new Intent(Intent.ACTION_VIEW);
        browseIntent.setData(Uri.parse(urlText));
        if(browseIntent.resolveActivity(getApplicationContext()) != null){
            startActivity(browseIntent);
        }
    }
}..
```

5. Sementara untuk eksperimen dengan **Explicit Intent**, kita harus siapkan **Activity** yang akan menerima request dari **MainActivity** ketika user meng-klik tombol “**Kirim**”. Buat Activity baru dengan

nama **ActivityDua**, dengan cara klik kanan pada folder **app > java > [domain].week04a_[NIM]** lalu pilih **New > Activity > Empty Activity**. Setelah dialog isian muncul, isi nama dengan **ActivityDua** dan nama layout dengan **activity_dua** lalu klik “**Finish**”.

6. Buka dan edit file **Android Manifest.xml** pada folder **app > manifests**. Temukan element **<activity>** yang baru saja dibuat oleh Android Studio untuk Activity kedua.

```
<activity android:name=".ActivityDua"></activity>
```

Ganti skrip tersebut dengan:

```
<activity android:name=".ActivityDua"
    android:label="Activity Kedua"
    android:parentActivityName=".MainActivity">
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=
            "id.ac.umn.khaeruzzaman.yaman.week04a_067806.MainActivity" />
```

Dengan menggunakan attribute **parentActivityName**, kita mengindikasikan bahwa **MainActivity** adalah parent dari **ActivityDua**. Hubungan ini dapat digunakan untuk Up Navigation pada app kita. App bar pada **ActivityDua** akan memiliki panah kiri (←) sehingga user dapat melakukan navigasi “upward” ke **MainActivity**.

7. Buka dan edit layout file **activity_dua.xml** pada folder **app > res > layout**. Untuk membuat sebuah **TextView** yang akan menampilkan pesan atau teks yang diterima dari **MainActivity**, sebuah **EditText** untuk mengisikan pesan balasan yang dikirim balik ke **MainActivity**, dan sebuah button untuk mengirim balasan tersebut. Sekilas tampilan Activity ke dua ini mirip dengan **ActivityMain** dengan hanya memiliki satu **Button** saja. Berikut skrip lengkap dari **activity_dua.xml**:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ActivityDua">
    <TextView
        android:id="@+id/pesanDiterima"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/pesanBalik"
            android:layout_width="270dp"
            android:layout_height="wrap_content"
            android:hint="Jawaban" />
        <Button
            android:id="@+id/kirimBalik"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="kirimBalik"
            android:text="Balas" />
    </LinearLayout>
</LinearLayout>

```

Catatan: layout file ini memiliki sebuah error pada object Button **kirimBalik** untuk attribute **android:onClick="kirimBalik"** karena pada activity **ActivityDua.java** belum ada implementasi method dengan nama dan signature **public void kirimBalik(View view)** yang akan menangani klik button pada button **kirimBalik**.

8. Buka file **ActivityDua.java** untuk membuat sebuah TextView object yang menerima pesan dari MainActivity, EditText object yang dapat diisi jawaban yang akan dikirim balik ke MainActivity, sebuah Button yang akan menangani interaksi klik dari user, serta beberapa method yang harus diimplementasikan agar proses passing data dari dan ke MainActivity berjalan dengan baik. Berikut kode yang harus ditambahkan pada **ActivityDua.java**:

```

public class ActivityDua extends AppCompatActivity {
    private TextView tvPesanDiterima;
    private EditText etJawaban;
    private Button btnBalasKirim;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dua);
        tvPesanDiterima = findViewById(R.id.pesanDiterima);
        etJawaban = findViewById(R.id.pesanBalik);
        btnBalasKirim = findViewById(R.id.kirimBalik);
        Intent mainIntent = getIntent();
        String pesanDiterima =
            mainIntent.getStringExtra("PesandariMain");
        tvPesanDiterima.setText(pesanDiterima);
    }
    public void kirimBalik(View view) {
        String jawaban = etJawaban.getText().toString();
        Intent balasIntent = new Intent();
        balasIntent.putExtra("Jawaban", jawaban);
        setResult(RESULT_OK, balasIntent);
        finish();
    }
}
  
```

Object `mainIntent` adalah object yang merepresentasikan Intent yang memanggil `ActivityDua` yaitu `MainActivity`. Method `getStringExtra("PesandariMain")` mengambil data yang dibawa / dikirim oleh `mainIntent` tersebut dengan nama "`PesandariMain`" yang selanjutnya ditampilkan pada object `TextView tvPesanDiterima`. Method `kirimBalik(...)` wajib ada pada `ActivityDua` karena object `Button kirimBalik` yang ada pada layout file `activity_dua.xml` memiliki atribut `android:onClick="kirimBalik"`. Pada Method ini, untuk kembalik ke `MainActivity` dengan mengirimkan hasil (atau pesan) tidak menggunakan method `startActivity(...)` tetapi menggunakan method `setResult(...)` karena `ActivityDua` ini aktif karena dipanggil oleh `MainActivity` dengan menggunakan `startActivityForResult()` yang akan menunggu feedback atau balasan dari activity yang dipanggilnya, dalam eksperimen ini method `setResult(...)` sebagai bentuk feedback dari activity yang dipanggil. Method `setResult(...)` ini akan berkomunikasi dengan callback method `onActivityResult(...)` yang diimplementasi pada activity pemanggil yaitu `MainActivity`.

9. Buka dan tambahkan TextView pada **activity_main.xml** untuk dapat menampung pesan jawaban yang diberikan oleh **ActivityDua**. Letakkan penambahan TextView dibawah ini setelah inner LinearLayout yang kedua:

```
<TextView
    android:id="@+id/jawaban"
    android:layout_width="match_parent"
    android:layout_height="60dp" />
```

10. Buka dan lengkapi **MainActivity.java** untuk mendeklarasikan variable TextView **tvJawaban** yang diletakkan didalam class MainActivity sebelum **onCreate(...)** method sebagai berikut:

```
tvJawaban = findViewById(R.id.jawaban);
```

Selanjutnya lakukan koneksi variable **tvJawaban** dengan object TextView **@+id/jawaban** yang ada pada layout file **activity_main.xml** di dalam method **onCreate(...)**, seperti di bawah ini:

```
private TextView tvJawaban;
```

Masih di dalam method **onCreate(...)** implementasikan penanganan klik pada button **btnKirim** menggunakan method **setOnItemClickListener(...)** yang akan mengirim sebuah pesan dengan nama **"PesanDariMain"** yang dibawa oleh Intent yang merepresentasikan activity yang dipanggil yaitu **ActivityDua.class** dengan menggunakan method **startActivityForResult(...)**. Berikut kode implementasi dari penanganan button tersebut:

```
btnKirim.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intentDua = new Intent(MainActivity.this,
                                         ActivityDua.class);
        String isian = etIsian.getText().toString();
        intentDua.putExtra("PesanDariMain",isian);
        startActivityForResult(intentDua, 1);
    }
});
```

11. Untuk dapat menerima kembali jawaban dari child activity dan/atau dari activity yang dipanggil menggunakan **startActivityForResult(...)**, kita harus mengimplementasikan call-back method **onActivityResult(...)** pada **MainActivity**. Sebagai overriding method, kode awal yang dilakukan alah memanggil method yang pada parent class dengan menggunakan keyword **super**, yaitu **super.onActivityResult(...)**. Kemudian yakinkan bahwa nilai **requestCode** yang dikirim oleh method

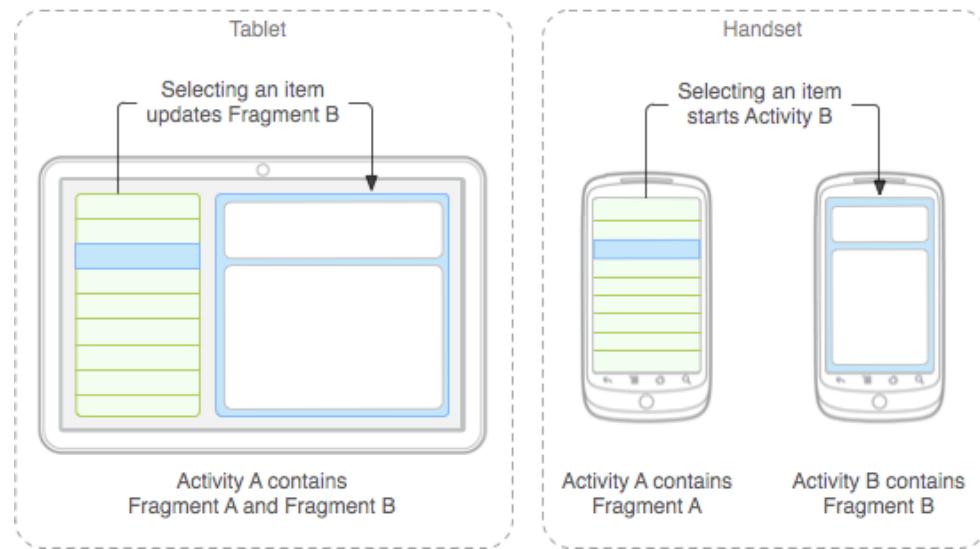
`startActivityForResult(...)` dengan yang diterima harus sama. Untuk kasus eksperimen ini nilainya adalah **1 (satu)**. Begitu juga dengan `resultCode`, harus sama antara yang dikirim oleh method `setResult(...)` yang ada pada activity yang dipanggil (**ActivityDua**) dengan yang diterima oleh method ini. Untuk eksperimen ini nilainya adalah **RESULT_OK**. Berikut kode pada method tersebut:

```
@Override
public void onActivityResult(int requestCode,
                             int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1) {
        if (resultCode == RESULT_OK) {
            String jawaban = data.getStringExtra("Jawaban");
            tvJawaban.setText(jawaban);
        }
    }
}
```

12. Demikian Tutorial pengembangan sebuah Android app dengan memanfaatkan Activity, Intent (implicit dan explicit), serta passing data antar Activity melalui Intent.

Tutorial: Fragment (via Layout)

Fragment mewakili perilaku atau bagian dari antarmuka pengguna dalam Activity. Anda bisa mengombinasikan beberapa fragment dalam satu Activity untuk membangun UI multipanel dan menggunakan kembali sebuah fragment dalam beberapa Activity. Anda bisa menganggap fragment sebagai bagian modular dari Activities, yang memiliki daur hidup sendiri, menerima kejadian masukkan sendiri, dan yang bisa Anda tambahkan atau hapus saat aktivitas berjalan (semacam "sub aktivitas" yang bisa digunakan kembali dalam aktivitas berbeda). Salah satu kegunaan yang praktis dari penggunaan fragment adalah ketika harus membuat 2 buah antarmuka yang mirip antara Tablet dan Smartphone yang dapat dilihat pada gambar berikut:



Fragment memiliki Lifecycle yang berbeda dari Activity, sehingga Anda harus mempelajarinya lebih lanjut. Lifecycle pada Fragment dapat dilihat pada tautan:

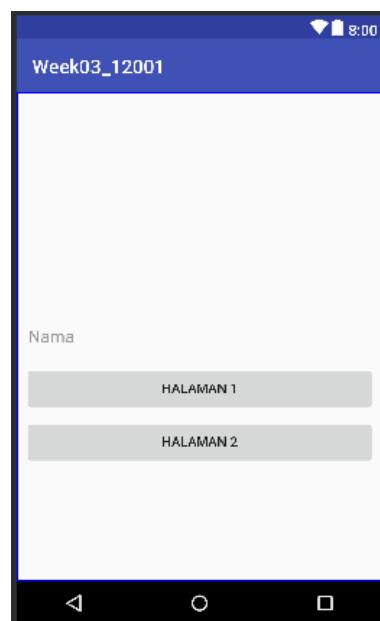
<https://developer.android.com/guide/components/fragments?hl=id>.

Pada tutorial ini Anda akan belajar untuk membuat Fragment sederhana dengan menggunakan metode layout.

1. Bukalah app\res\layout\activity_first.xml
2. Tambahkan 2 buah button tetap di bawah EditText pada app\java\[namapackage]\MainActivity, dengan id “main_button_change_1” dan “main_button_change_2”, width “match_parent”, text “Halaman 1” dan “Halaman 2”

Layout tetap menggunakan ConstraintLayout, Anda boleh mempergunakan Tab **Design** / Drag-Drop-UI, tapi tetap disarankan tetap menggunakan Tab **Text**.

Buatlah tampilan semirip mungkin dengan gambar di bawah ini.



3. Buatlah dua buah Activity baru (EmptyActivity) dengan nama **SecondActivity** dan **ThirdActivity**.
4. Tambahkanlah pada **app\java\[namapackage]\MainActivity** kode untuk berpindah Activity dari **MainActivity** ke **SecondActivity** dan dari **MainActivity** ke **ThirdActivity** pada saat Button `main_button_change_1` dan `main_button_change_2` pada **app\res\layout\activity_main.xml** ditekan.
5. Tambahkan dua buah layout, dengan Root Element-nya berupa LinearLayout dengan orientation vertical, masing-masing berikan nama `fragment_first.xml` dan `fragment_second.xml`.
6. Pada `fragment_first.xml`, tambahkan sebuah EditText dengan id "fragment_first_edittext_tulisan", width "match_parent", hint "Tulisan" dan sebuah Button dengan id "fragment_first_button_berubah" dan text "Ganti Tulisan".

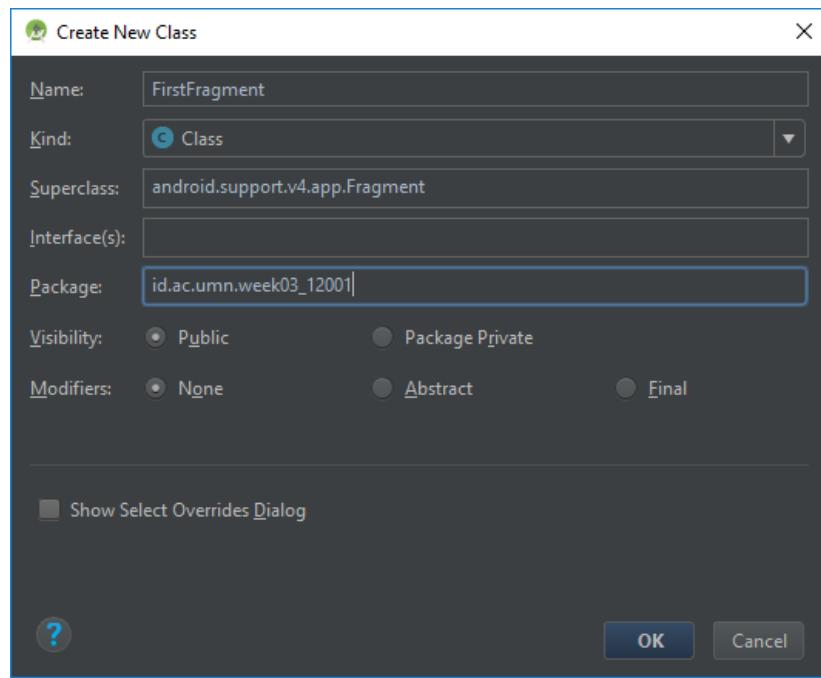
Hasil Akhir pada **fragment_first.xml** dapat dilihat pada gambar di bawah.



7. Pada **fragment_second.xml**, tambahkan sebuah TextView dengan id "fragment_second_textview_tulisan", width "wrap_content", dan text "initial TextView ". Modifikasilah sehingga hasil akhir pada **fragment_second.xml** dapat dilihat pada gambar di bawah.



8. Buatlah sebuah class Java (Klik kanan pada package → New → Java Class dengan nama "FirstFragment" dan merupakan **superclass** dari "android.support.v4.app.Fragment".



(Samakan package dengan package yang Anda buat)

Apabila nanti pada computer di UMN terdapat error pada gradle yang menyatakan bahwa **implementation 'com.android.support:support-v4:27.1.1'** tidak dapat ditemukan, gantilah dengan **implementation 'com.android.support:support-fragment:27.1.1'** kemudian Sync gradle ulang.

9. Bukalah **FirstFragment** yang sudah dibuat, dan tambahkan / modifikasi kode seperti di bawah ini.

```
public class FirstFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                            @Nullable ViewGroup container,
                            @Nullable Bundle savedInstanceState) {

        View view = inflater.inflate(R.layout.fragment_first, container, false);
        return view;
    }
}
```

Fungsi dari inflater ini adalah sebagai **penempel** layout (xml) terhadap **ViewGroup** yang disediakan pada Fragment / Activity.

10. Buatlah satu class lagi dengan nama **SecondFragment** dengan superclass yang sama seperti **FirstFragment** dan tambahkan / modifikasi kode seperti di bawah ini.

```
public class SecondFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                            @Nullable ViewGroup container,
                            @Nullable Bundle savedInstanceState) {

        View view = inflater.inflate(R.layout.fragment_second, container, false);
        return view;
    }
}
```

11. Bukalah `activity_second.xml` dan tambahkanlah 2 buah fragment dengan id "second_activity_fragment_1" dan "second_activity_fragment_2", width "match_parent" dan height "wrap_content" dan hubungkan fragment 1 dengan **FirstFragment** dan fragment 2 dengan **SecondFragment**.

Kode lengkapnya dapat dilihat di bawah sini namun ...

Cobalah untuk mengerjakannya tanpa melihat kode di bawah ini terlebih dahulu.

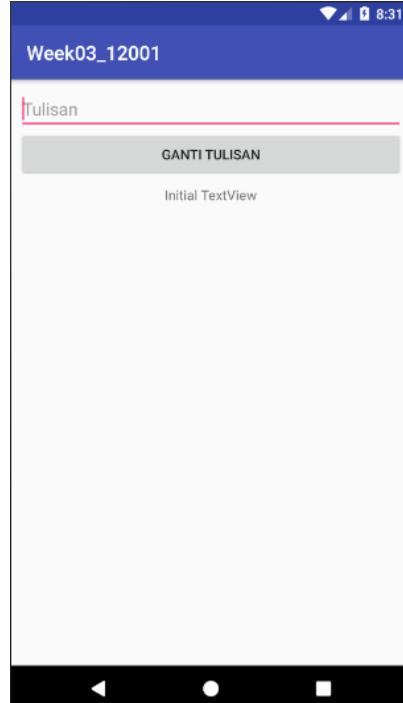
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity">

    <fragment
        android:id="@+id/second_activity_fragment_1"
        android:name="id.ac.umn.week03_12001.FirstFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <fragment
        android:id="@+id/second_activity_fragment_2"
        android:name="id.ac.umn.week03_12001.SecondFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/second_activity_fragment_1" />
</android.support.constraint.ConstraintLayout>
```

(Samakan name dengan package yang Anda buat)

12. Jalankan aplikasi yang sudah Anda buat, dan tekanlah tombol dengan tulisan **Halaman 2** pada **MainActivity**, seharusnya aplikasi akan memunculkan tampilan sebagai berikut



13. Selamat, Anda sudah berhasil membuat Fragment dengan menggunakan Layout secara manual.

Tutorial: Fragment (via Programmatic)

Pada tutorial sebelumnya, Anda sudah berhasil membuat Fragment dengan menggunakan layout, dengan cara membuat 2 Class Fragment dan 2 Layout Fragment dan hanya menempelkan Fragment tersebut secara langsung pada <fragment> yang terdapat di Activity. Pada tutorial ini Anda akan me-“nempelkan” Fragment terhadap <fragment> pada Activity secara *programmatic*.

Kita akan menggunakan beberapa Class baru yang bertugas untuk mengelola seluruh Fragment yang ada pada aplikasi Android yaitu **FragmentManager** dan **FragmentTransaction**, kedua Class ini sangat berguna apabila Anda akan membuat Fragment yang bersifat dinamis, misalnya dengan menekan satu tombol Anda akan mengubah tampilan dari separuh halaman Activity.

1. Buatlah sebuah Activity baru (Empty Activity) dengan nama **ThirdActivity** dan layout dengan nama **activity_third.xml**.
2. Buatlah kode agar pada **MainActivity**, tombol `main_button_change_2` dapat berpindah ke **ThirdActivity** pada saat ditekan tombolnya.
3. Buka file **activity_third.xml** dan tuliskanlah kode di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ThirdActivity">

    <FrameLayout
        android:id="@+id/third_activity_fragment_1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <FrameLayout
        android:id="@+id/third_activity_fragment_2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintTop_toBottomOf="@+id/third_activity_fragment_1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

Perhatikan bahwa sudah tidak menggunakan tag `<fragment>` namun sudah diganti menjadi `<FrameLayout>`

Hal ini dikarenakan untuk membuat Fragment bersifat programmatic dibutuhkan `<FrameLayout>` bukan `<fragment>`.

4. Bukanlah **ThirdActivity** dan tambahkan kode yang ada di bawah ini.

```
public class ThirdActivity extends AppCompatActivity {

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

        FragmentManager fragmentManager = getSupportFragmentManager();
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

        Fragment firstFragment = new FirstFragment();
        fragmentTransaction.replace(R.id.third_activity_fragment_1, firstFragment);

        Fragment secondFragment = new SecondFragment();
        fragmentTransaction.replace(R.id.third_activity_fragment_2, secondFragment);
        fragmentTransaction.commit();
    }
}
```

Perhatikan pada saat import Fragment, FragmentManager, dan FragmentTransaction. Pada Praktikum ini kita menggunakan android.support.v4.app.xxx untuk ketiganya.

Selamat, Anda sudah berhasil membuat Fragment secara programmatic.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project anda dengan nama **Weeko4a_[NIM]** dan Eksperimen dengan Fragment baik yang menggunakan Layout (manual) maupun yang programmatic sudah dapat dijalankan dan berhasil menampilkan sesuai dengan yang diharapkan. Zip (kompres) file-file berikut dan submit ke isian tugas pada eLearning:
 - a. Weeko4a_[NIM]: (Activity, Intent dan Passing Data antar Intent)
 - 1) MainActivity.java
 - 2) ActivityDua.java
 - 3) activity_main.xml
 - 4) activity_dua.xml
 - 5) AndroidManifest.xml
 - b. Fragment menggunakan Layout
 - 1) MainActivity.java
 - 2) SecondActivity.java
 - 3) ThirdActivity.java

- 4) FirstFragment.java
 - 5) SecondFragment.java
 - 6) Activity_main.xml
 - 7) fragment_first.xml
 - 8) fragment_second.xml
 - 9) activity_second.xml
 - 10) activity_third.xml
- c. Fragment via Programmatic
- 1) MainActivity.java
 - 2) SecondActivity.java
 - 3) ThirdActivity.java
 - 4) FirstFragment.java
 - 5) SecondFragment.java
 - 6) Activity_main.xml
 - 7) fragment_first.xml
 - 8) fragment_second.xml
 - 9) activity_second.xml
 - 10) activity_third.xml
2. Lakukan pengujian terhadap app anda dengan skenario berbagai kemungkinan yang mengirim data dari satu activity ke activity lainnya. Beri catatan dan kesimpulan dari hasil pengujian app anda submit ke eLearning portal.

REFERENSI

MODUL 5

(CUSTOM VIEW, CANVAS, PAINT DAN DRAWING)

DESKRIPSI TEMA

- Custom View
- Canvas
- Paint
- Drawing

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat Custom View serta memanfaatkan Canvas object dan Paint object untuk menggambar berbagai bentuk seperti kotak, lingkaran, garis serta Jalur (atau Path).

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Membuat app Menggambar sederhana

App yang akan dibuat pada modul praktikum ini adalah sebuah app menggambar pada layar android dengan menggunakan beberapa class yang terkait seperti subclass dari View (biasa disebut Custom View yang menggabungkan Class Class yang dibutuhkan), Canvas (sebagai area untuk menggambar), Paint (sebagai tool dan configurasi untuk menggambar). App yang akan dibuat ini juga membutuhkan logika pemrograman serta pengetahuan dan keterampilan konsep pemrograman Java yang cukup lengkap.

Bukalah Android Studio dan pilih **Start a new Android Studio Project**

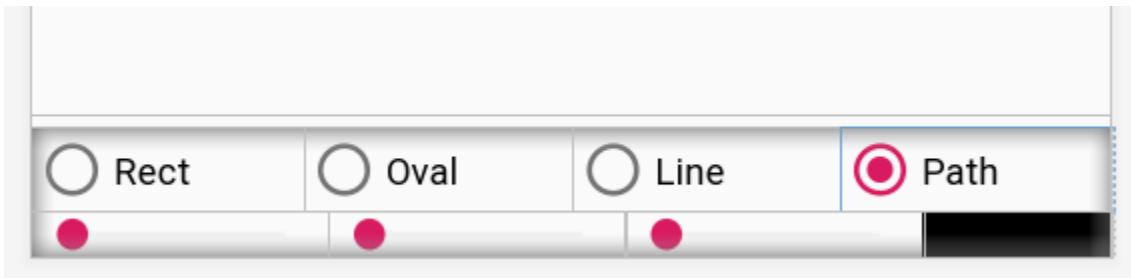
1. Buatlah project dengan format:

Name	:	Week [xx]_[NIM]. Contoh: Week 05A_12001.
Company Domain	:	umn.ac.id.
Project Location	:	D:\FTI\IF634_IS534_MOBILE\workspaces\
Target Android	:	(Phone and Tablet) API 25: Android 7.1.1 (Nougat)
Minimum Android	:	API 21: Android 5.0 (lollipop)
Activity	:	Empty Activity
Activity Name	:	MainActivity
Layout Name	:	activity_main

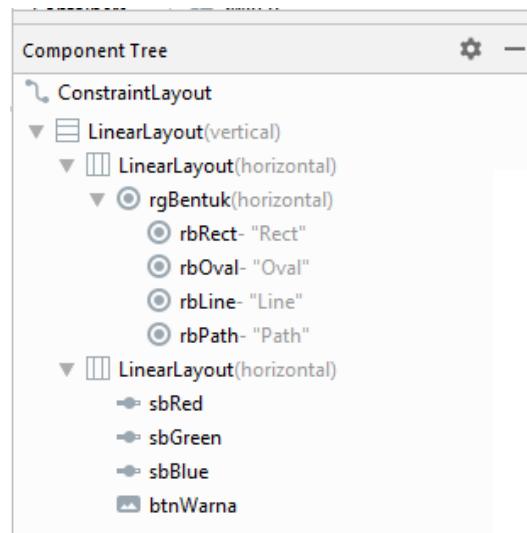
2. App yang akan dibuat tidak menggunakan AppBar. Untuk menyembunyikan AppBar, buka file `app > res > styles.xml`. Ganti Style tag dengan nama “**AppTheme**” dengan kode berikut:

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

3. Seperti pada modul sebelumnya, buka dan edit file layout `app > res > layout activity_main.xml`. Root Layout yang digunakan tetap ConstraintLayout. Hapus tag TextView yang menampilkan “hallo Word” ganti dengan pengaturan layout menggunakan nested ListedLayout. Untuk mendapatkan tampilan seperti gambar di bawah ini:



LinearLayout vertical berisi dua LinearLayout horizontal yang diletakkan tepat di bagian paling bawah layer dengan cara menetapkan constraint bottom ke “parent”. LinearLayout pertama berisi sebuah RadioGroup yang berisi empat buah RadioButton yang merepresentasikan bentuk: Rectangle (Kotak), Oval (Lingkaran atau elip), Garis Lurus dan Path. LinearLayout kedua berisi tiga buah SeekBar untuk masing-masing warna dasar: Red, Green, Blue yang memiliki nilai antara 0 sampai 255; dan sebuah dummy button untuk menampilkan hasil warna yang dihasilkan dari perpaduan tiga nilai Red, Green, Blue. Berikut Struktur dari Layout dan komponen:



Text mode dari layout ini dapat dilihat seperti dibawah ini:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <RadioGroup
                android:id="@+id/rgBentuk"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <RadioButton
                    android:id="@+id/rbRect"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:checked="false"
                    android:text="Rect" />
                <RadioButton
                    android:id="@+id/rbOval"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:text="Oval" />
                <RadioButton
                    android:id="@+id/rbLine"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:text="Line" />
                <RadioButton
                    android:id="@+id/rbPath"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
```

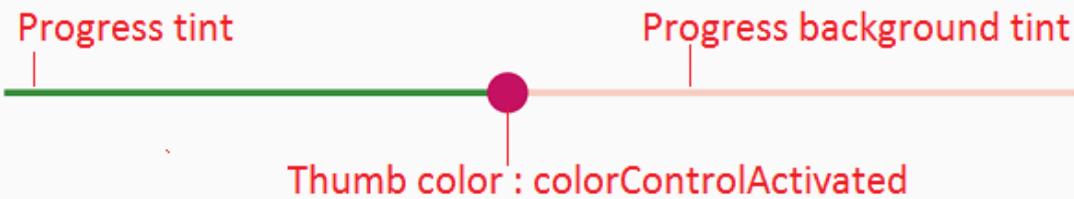
```

        android:layout_weight="1"
        android:checked="true"
        android:text="Path" />
    </RadioGroup>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <SeekBar
        android:id="@+id/sbRed"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:max="255" />
    <SeekBar
        android:id="@+id/sbGreen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:max="255" />
    <SeekBar
        android:id="@+id/sbBlue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:max="255" />
    <ImageButton
        android:id="@+id	btnWarna"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#oooooooo" />
</LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

4. SeekBar yang ada pada app ini digunakan untuk merepresentasikan warna dasar Red, Green, Blue. Untuk itu, sangat tepat kalau kita gunakan warna sesuai dengan fungsinya dengan cara mengganti attribute pada style masing-masing seekBar, Attribut yang harus diganti adalah: **android:progressTint**, dan **android:colorControlActivated** yang dilakukan pada file **app > res > styles.xml**. Berikut kode penambahan pada file style agar seekBar memiliki warna yang sesuai:

```
<style name="seekBarRed" parent="Widget.AppCompat.SeekBar">
    <item name="android:progressTint">#FFoooo</item>
    <item name="android:colorControlActivated">#FFoooo</item>
</style>
<style name="seekBarGreen" parent="Widget.AppCompat.SeekBar">
    <item name="android:progressTint">#ooFFoo</item>
    <item name="android:colorControlActivated">#ooFFoo</item>
</style>
<style name="seekBarBlue" parent="Widget.AppCompat.SeekBar">
    <item name="android:progressTint">#ooooFF</item>
```



- Setelah mendefinisikan style untuk masing-masing seekBar: **seekBarRed**, **seekBarGreen**, **seekBarBlue**. Berikutnya adalah menambahkan style ini pada masing-masing seekBar di dalam file layout app > res > activity_main.xml dengan menyisipkan attribute

```
        android:theme="@style/seekBar [Red|Green|Blue]"
```

Sehingga masing-masing seekBar menjadi seperti pada skrip di bawah ini:

```
<SeekBar
    android:id="@+id/sbRed"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:theme="@style/seekBarRed"
    android:max="255" />
<SeekBar
    android:id="@+id/sbGreen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:theme="@style/seekBarGreen"
    android:max="255" />
<SeekBar
    android:id="@+id/sbBlue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:theme="@style/seekBarBlue"
    android:max="255" />
```

6. Selanjutnya adalah membuka dan mengedit file **MainActivity.java** untuk membuat object-object yang telah disiapkan pada layout file seperti **RadioGroup**, **RadioButton**, **SeekBar**, dan **Button**. SeekBar harus semua dibuatkan *objectnya*, dummy ImageButton untuk preview warna, dan RadioGroup. Sementara untuk RadioButton cukup satu buah object yang generic, karena dalam sebuah RadioGroup hanya satu RadioButton yang bisa dipilih dengan memanfaatkan method **getCheckedRadioButtonId()** di dalam penanganan perubah pilihan oleh user dengan method **setOnCheckedChangeListener(...)**. Berikut kode setelah penambahan object-object yang dijelaskan di atas.

```
// ... import Class yang dibutuhkan

public class MainActivity extends AppCompatActivity {
    private SeekBar sbRed, sbGreen, sbBlue;
    private RadioGroup rgBentuk;
    private RadioButton rbPilih;
    private ImageButton btnWarna;

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

        sbRed = findViewById(R.id.sbRed);
        sbGreen = findViewById(R.id.sbGreen);
        sbBlue = findViewById(R.id.sbBlue);
        btnWarna = findViewById(R.id.btnWarna);
        rgBentuk = findViewById(R.id.rgBentuk);

    }
}
```

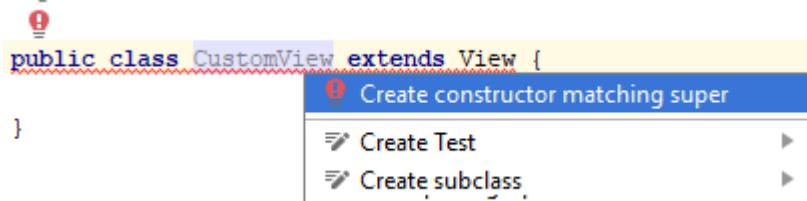
Langkah berikutnya, membuat Custom View (sebuah class yang meng-extends class). Class ini yang akan digunakan untuk menggambarkan object-object. Bentuk (sesuai dengan pilihan user menggunakan RadioGroup) menggunakan warna yang dimasukkan oleh user menggunakan tiga buah SeekBar, warna dasar. Untuk membuat Class ini, Klik kanan pada folder dimana terletak **MainActivity.java** yaitu **app > java > [domain].week05_[NIM]**, lalu pilih **New > Java Class**. Setelah jendela "Create New Class" muncul isi nama dengan **CustomView** dan parent class dengan **View**

(`android.view`) → `android.view.View`, maka kode Java yang dibuatkan oleh Android Studio sebagai berikut:

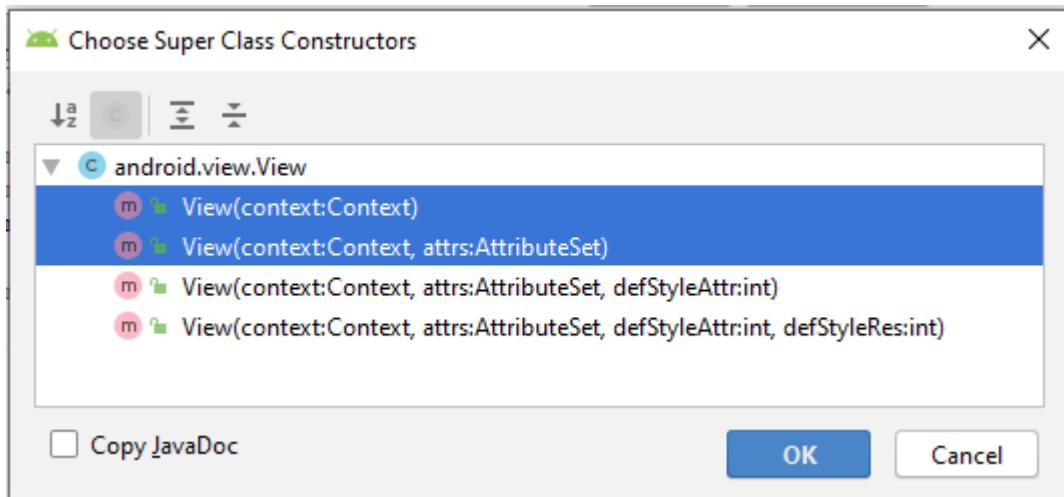
```
import android.view.View;

public class CustomView extends View {
```

Kode tersebut langsung memiliki Error karena subclass dari View membutuhkan setidaknya default Konstruktor. Untuk menambahkan konstruktor yang dibutuhkan, arahkan kursor (pointer mouse) nama class (CustomView) lalu gunakan ALT+ENTER dan pilih “**Create constructor matching super**”



yang akan memunculkan jendela “**Choose Super Class Constructors**” seperti gambar berikut:



Pilih dua konstruktor pertama dengan menggunakan **CTRL+CLICK** pada keduanya, sehingga kode Custom View menjadi seperti di bawah ini:

```
public class CustomView extends View {  
  
    public CustomView(Context context) {  
        super(context);  
    }  
  
    public CustomView(Context context, @Nullable AttributeSet attrs) {  
        super(context, attrs);  
    }  
}
```

Tambahkan CustomView ini kedalam layout file **activity_main.xml**. Custom view ini diletakkan tepat diatas Layout vertical pertama. Kita harus memberi id terhadap layout ini misal dengan nama id **layoutcontrol** karena akan digunakan sebagai Button constraint oleh **CustomView**. Untuk nama lengkap **CustomView** sangat tergantung dari nama package (atau domain) serta nama dari project anda. Ganti nama lengkap dengan **[domain].week05_[NIM].CustomView**. Berikut kode tambahan dan perubahan pad **activity_main.xml**.

```
<id.ac.umn.week05_067806.CustomButton  
    android:id="@+id/customView"  
    android:layout_width="match_parent"  
    android:layout_height="odp"  
    app:layout_constraintBottom_toTopOf="@id/layoutcontrol"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />  
<LinearLayout  
    android:id="@+id/layoutcontrol"
```

9. Selanjutnya adalah bagaimana kita membuat skenario atau behaviour dari app yang kita buat ini, serta mengimplementasikannya pada koding kita. Skenario atau behavior dari dari app ini adalah user menentukan **bentuk** serta **warna** stroke (atau online) yang ingin digambarkan pada kanvas (atau area drawing) yang ada pada CustomView. Letak dan ukuran dari bentuk yang akan digambarkan dilakukan dengan meng-klik dan grad (menggeser) pointer pada layyar. Pemilihan **bentuk** dan **warna** dilakukan pada **MainActivity.java** sementara proses menggambar dilakukan

pada **CustomView.java**. Untuk mengimplementasi ini perlu ada **static** variable “**bentuk**” dengan tipe data String dan “**warna**” dengan tipe data **int** sebagai hasil dari method **Color.rgb(int red, int green, int blue)** pada class **CustomView** serta mutator method **public void gantiBentuk(String bentukBaru)** dan **public void gantiWarna(int red, int green, int blue)**. Berikut penambahan kode pada **CustomView.java**.

```
public class CustomView extends View {
    private static String bentuk = "Path";
    private static int warna = Color.rgb(0,0,0);

    public void gantiBentuk(String bentukBaru){
        bentuk = bentukBaru;
    }
    public void gantiWarna(int red, int green, int blue){
        warna = Color.rgb(red, green, blue);
    }
}
```

10. Buka dan edit **MainActivity.java** untuk membuat object **CustomView** dan menghubungkan dengan object pada layout **activity_main.xml**. Sebelum **onCreate(...)** method tambahkan deklarasi:

```
private CustomView customView;
```

dan di dalam **onCreate(...)** tambahkan:

```
customView = findViewById(R.id.customView);
customView = new CustomView(this);
```

11. Agar **MainActivity.java** dapat mengirimkan nilai “**bentuk**” yang benar, maka penanganan interaksi user pada **RadioGroup** harus ditangani dengan mengimplementasi method **setOnCheckedChangeListener(...)** untuk object **rgBentuk** sebagai berikut:

```
rgBentuk.setOnCheckedChangeListener(
    new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId)
        {
            int intRb = rgBentuk.getCheckedRadioButtonId();
            rbPilih = findViewById(intRb);
            String bentuk = rbPilih.getText().toString();
            customView.gantiBentuk(bentuk);
        }
    });

```

Sementara untuk mengirimkan nilai “**Warna**” harus memiliki tiga instance variable int untuk menyimpan nilai untuk masing-masing warna: **red**, **green**, dan **blue**. Tambahkan tiga variable ini sebelum method **onCreate(...)** sebagai berikut:

```
private int red    = 0;  
private int green = 0;  
private int blue  = 0;
```

Serta implementasi penanganan masing-masing object seekBar menggunakan method **setOnSeekBarChangeListener(...)** sebagai berikut:

```
sbRed.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
    @Override  
    public void onProgressChanged(SeekBar seekBar, int progress,  
        boolean fromUser) {  
    }  
    @Override  
    public void onStartTrackingTouch(SeekBar seekBar) {  
    }  
    @Override  
    public void onStopTrackingTouch(SeekBar seekBar) {  
        red = sbRed.getProgress();  
        customView.gantiWarna(red, green, blue);  
        btnWarna.setBackgroundColor(Color.rgb(red, green, blue));  
    }  
}
```

Method yang diperlukan untuk diimplementasi agar mendapatkan nilai progress dari seekBar adalah **onStopTrackingTouch(...)** karena method ini akan dieksekusi saat user berhenti menggeser lingkaran pada seekBar. Di dalam method ini selain mengirim nilai warna” juga mengubah warna background dari dummy ImageButton **btnWarna**. Lakukan hal yang sama untuk seekBar lainnya yaitu **sbGreen** dan **sbBlue** seperti pada kode dibawah ini:

```

sbGreen.setOnSeekBarChangeListener(
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress,
            boolean fromUser) {
        }
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
        }
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
            green = sbGreen.getProgress();
            customView.gantiWarna(red, green, blue);
            btnWarna.setBackgroundColor(Color.rgb(red, green, blue));
        }
    });
}

sbBlue.setOnSeekBarChangeListener(
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress,
            boolean fromUser) {
        }
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
        }
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
            blue = sbBlue.getProgress();
            customView.gantiWarna(red, green, blue);
            btnWarna.setBackgroundColor(Color.rgb(red, green, blue));
        }
    });
}

```

12. Pada bagian **MainActivity.java** sudah lengkap, selanjutnya adalah mengedit **CustomView.java** untuk dapat menampilkan lukisan yang diinginkan oleh user. Dalam penanganan bentuk, **Path** memiliki behavior yang berbeda dibandingkan dengan bentuk yang lain. **Path** membutuhkan object dari class **Path (android.graphics.Path)** yang berisi titik-titik (x, y) yang dilalui pada saat user melakukan **drag** (menggeserkan pointer atau jarinya pada layer). Sementara untuk bentuk lainnya **Rect**, **Oval**, dan **Line** yang diperlukan titik awal pada saat user pertama kali tap (atau klik) jarinya pada layar, dan titik

akhir pada saat user melepas jarinya dari layar. Pada proses dragging diperlukan seperti: **pathKu** (tipe data Path), **pathX** dan **pathY** untuk menyimpan titik-titik yang dilalui pada saat dragging yang akan ditambahkan ke object **pathKu**. **awalY** dan **awalY** yang akan menyimpan titik awal saat user dragging yang digunakan untuk menggambar bentuk antara (menggunakan canvas yang berbeda) sampai user melepas jarinya dari layar. Variable **canvasKu** (tipe data **android.graphics.Canvas**) merupakan kanvas utama yang akan menampung bentuk-bentuk yang digambarkan oleh user. Variable **paintKu** (tipe data **android.graphics.Paint**) merupakan variable yang menyimpan style dari stroke yang digunakan, dan **warna** yang dipakai untuk melukis bentuk. Variable constant **TOLERANSI_SENTUH** yang menentukan jarak minimum antara titik sentuh jari pada layar. Berikut variable-variable yang ada pada class **CustomView**:

```
public class CustomView extends View {
    private static String bentuk = "Path";
    private static long warna = Color.rgb(0,0,0);
    private Paint paintKu;
    private Path pathKu;
    private static Canvas kanvasKu;
    private Bitmap bitmapKu;
    private float pathX, pathY,
               awalX, awalY, antarX, antarY;
    private static final float TOLERANSI_SENTUH = 4;
```

13. Melakukan inisialisasi variable-variable di atas pada konstruktur yang kedua sebagai berikut:

```
public CustomView(Context context, @Nullable AttributeSet attrs) {
    super(context, attrs);
    paintKu = new Paint();
    paintKu.setColor(warna);
    paintKu.setAntiAlias(true);
    paintKu.setDither(true);
    paintKu.setStyle(Paint.Style.STROKE);
    paintKu.setStrokeJoin(Paint.Join.ROUND);
    paintKu.setStrokeCap(Paint.Cap.ROUND);
    paintKu.setStrokeWidth(12);
    pathKu = new Path();
}
```

14. Proses menggambar (Drawing) pada sebuah custom View dibutuhkan 1. Meng-override method **onDraw()** untuk menggambar customView menggunakan sebuah **Canvas** object yang memanfaatkan style pada **Paint** object; 2. Menghitung ukuran dari pada View Ketika tampil pertama kali dan Ketika terjadi perubahan dengan meng-override method **onSizeChanged()**. 3. Gunakan method **invalidate()** untuk merespon klik user untuk menginvalidasi seluruh view, sehingga memaksa memanggil kembali method **onDraw()** untuk menggambar ulang view. Untuk dapat menerima dan merespons interaksi dari user, dibutuhkan untuk meng-override method **onTouchEvent(MotionEvent event)**. Action user yang akan digunakan pada praktikum kali ini adalah: **ACTION_DOWN** (saat user pertama kali sentuh layar), **ACTION_MOVE** (saat user drag/menggeser), dan **ACTION_UP** (saat user mengangkat jarinya). Berikut implementasi masing-masing method yang disebut di atas:

```
@Override
protected void onSizeChanged(int width, int height,
    int oldWidth, int oldHeight) {
    super.onSizeChanged(width, height, oldWidth, oldHeight);
    bitmapKu = Bitmap.createBitmap(width, height,
        Bitmap.Config.ARGB_8888);
    kanvasKu = new Canvas(bitmapKu);
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawBitmap(bitmapKu, 0, 0, null);
    // menggambar Bantuk antara(Rect, Oval dan Line)
    if (bentuk.equalsIgnoreCase("Rect")) {
        canvas.drawRect(awalX, awalY, antarX, antarY, paintKu);
    } else if (bentuk.equalsIgnoreCase("Oval")) {
        canvas.drawOval(awalX, awalY, antarX, antarY, paintKu);
    } else if (bentuk.equalsIgnoreCase("Line")) {
        canvas.drawLine(awalX, awalY, antarX, antarY, paintKu);
    }
}
```

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    float x = event.getX();
    float y = event.getY();
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            mulaiSentuh(x, y);
            break;
        case MotionEvent.ACTION_MOVE:
            geser(x, y);
            break;
        case MotionEvent.ACTION_UP:
            lepas();
            break;
        default:
    }
    return true;
}
```

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    float x = event.getX();
    float y = event.getY();
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            mulaiSentuh(x, y);
            break;
        case MotionEvent.ACTION_MOVE:
            geser(x, y);
            break;
        case MotionEvent.ACTION_UP:
            lepas();
            break;
        default:
    }
    return true;
}
```

```

private void mulaiSentuh(float x, float y) {
    if(bentuk.equalsIgnoreCase("Path")){
        pathKu.moveTo(x, y);
        pathX = x;
        pathY = y;
    } else {
        awalX = x;
        awalY = y;
    }
}

private void geser(float x, float y) {
    float dx = Math.abs(x - pathX);
    float dy = Math.abs(y - pathY);
    if(dx >= TOLERANSI_SENTUH || dy >= TOLERANSI_SENTUH) {
        paintKu.setColor(warna);
        if(bentuk.equalsIgnoreCase("Path")){
            pathKu.quadTo(pathX, pathY, (x + pathX) / 2,
                           (y + pathY) / 2);
            kanvasKu.drawPath(pathKu, paintKu);
            pathX = x;
            pathY = y;
        } else {
            antarX = x;
            antarY = y;
        }
    }
    invalidate();
}

private void lepas(){
    if(bentuk.equalsIgnoreCase("Path")){
        pathKu.reset();
    } else if(bentuk.equalsIgnoreCase("Rect")){
        kanvasKu.drawRect(awalX, awalY, antarX, antarY, paintKu);
    } else if(bentuk.equalsIgnoreCase("Oval")){
        kanvasKu.drawOval(awalX, awalY, antarX, antarY, paintKu);
    } else if(bentuk.equalsIgnoreCase("Line")){
        kanvasKu.drawLine(awalX, awalY, antarX, antarY, paintKu);
    }
}

```

15. Sampai tahap ini app sudah dapat membuat gambar dengan berbagai bentuk dan warna sesuai dengan yang diinginkan. Silahkan coba di run app anda, Jika masih ada error silahkan coba dilakukan debug dan coba diperbaiki program anda sampai selesai.
16. App ini belum ada fasilitas untuk menghapus isi kanvas untuk menggambar dari awal. Untuk itu kita lanjutkan untuk menambahkan “Custom image Button” menggunakan hasil drawing sebagai tombol yang jika user klik area tersebut maka isi dari kanvasKu akan dihapus dan user dapat memulai menggambar Kembali dengan kanvasKu kosong. Dibutuhkan sebuah instance variable Boolean dengan nama **clear** sebagai indicator status drawing sedang dalam proses pembersihan atau tidak. Tambahkan variable clear dengan initial value false sebelum konstruktor pertama seperti:

```
private boolean clear = false;
```

Gambar ikon  pada kanan atas layar secara manual drawing. Ikon tersebut harus selalu ada, maka penggambaran ikon dilakukan di dalam method **onDraw()** dengan menggunakan paint dan canvas object local (bukan paintKu dan kanvasKu). Proses penggambaran bentuk (Rect, Oval, dan Line) antara pada method **onDraw()** ini juga harus ditambahkan kondisi **if(!clear)** (jika dalam status bukan dalam proses pembersihan). Ubah **onDraw()** method menjadi seperti ini:

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawBitmap(bitmapKu, 0, 0, null);
    Paint lokalPaint = new Paint();
    lokalPaint.setStyle(Paint.Style.FILL);
    lokalPaint.setColor(Color.RED);
    canvas.drawCircle(50,50,50, lokalPaint);
    lokalPaint.setStyle(Paint.Style.STROKE);
    lokalPaint.setStrokeWidth(10);
    lokalPaint.setColor(Color.WHITE);
    canvas.drawLine(20,20,80,80,lokalPaint);
    canvas.drawLine(20,80,80,20,lokalPaint);

    if(!clear) {
        // menggambar Bentuk antara(Rect, Oval dan Line)
        if (bentuk.equalsIgnoreCase("Rect")) {
            canvas.drawRect(awalX, awalY, antarX, antarY, paintKu);
        } else if (bentuk.equalsIgnoreCase("Oval")) {
            canvas.drawOval(awalX, awalY, antarX, antarY, paintKu);
        } else if (bentuk.equalsIgnoreCase("Line")) {
            canvas.drawLine(awalX, awalY, antarX, antarY, paintKu);
        }
    }
```

```

    }
}

```

Ubah juga pada method **mulaiSentuh(...)** dengan menambahkan kondisi **if(x <= 100 && y <= 100)** → jika user meng-klik awal pada lokasi ikon clear tadi pojok kanan atas 100 x 100 pixel. Berikut kode method **mulaiSentuh(...)** agar dapat merespon proses penghapusan isi dari kanvasku menggunakan method **drawColor(Color.WHITE)**.

```

private void mulaiSentuh(float x, float y) {
    // Lingkaran Clear diklik
    if (x <= 100 && y <= 100) {
        kanvasKu.drawColor(Color.WHITE);
        clear = true;
        invalidate();
    } else {
        if (bentuk.equalsIgnoreCase("Path")) {
            pathKu.moveTo(x, y);
            pathX = x;
            pathY = y;
        } else {
            awalX = x;
            awalY = y;
        }
        clear = false;
    }
}

```

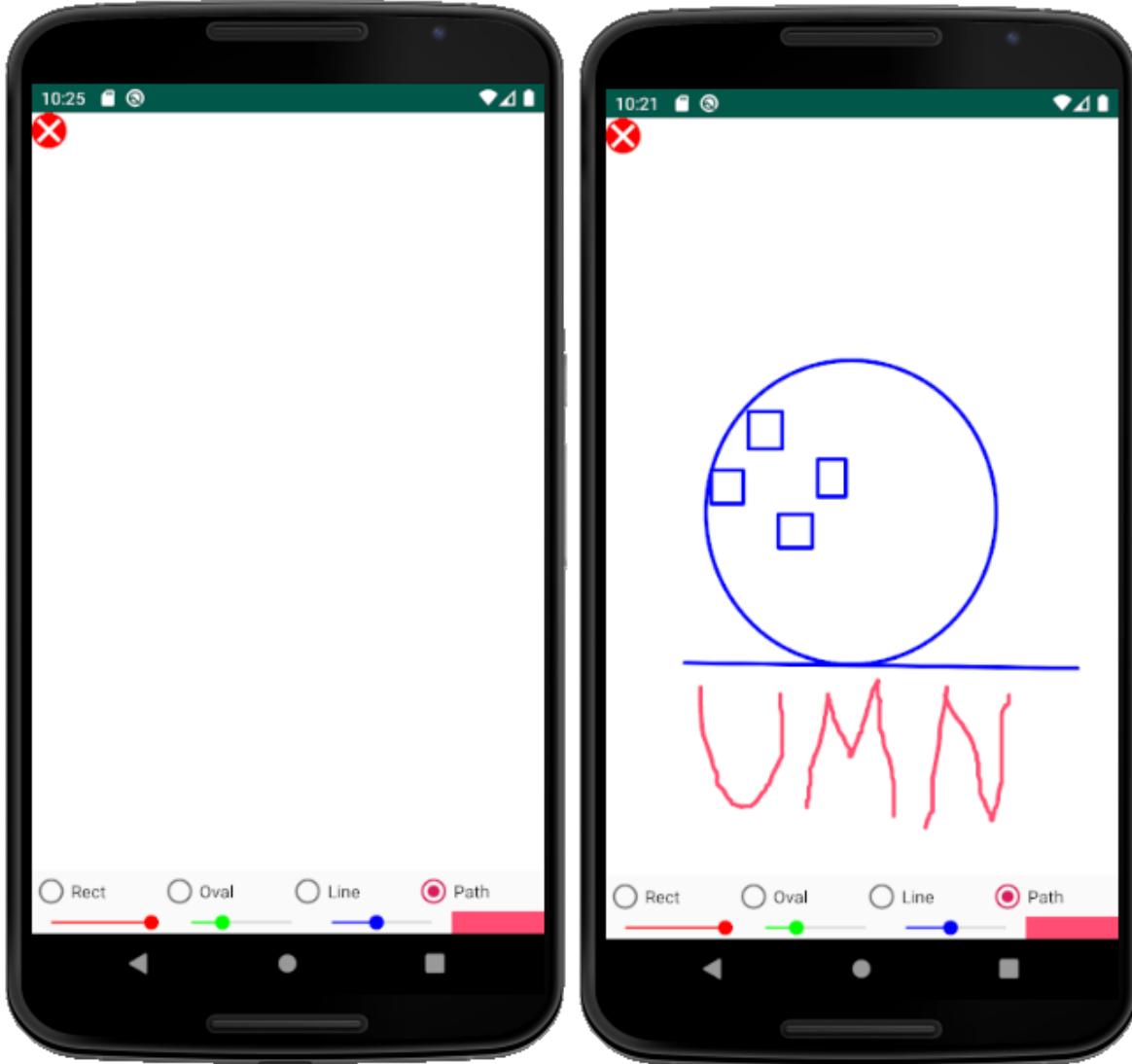
Method **lepas()** juga harus ditambahkan kondisi **if(clear)** seperti kode berikut:

```

private void lepas(){
    if(!clear) {
        if (bentuk.equalsIgnoreCase("Path")) {
            pathKu.reset();
        } else if (bentuk.equalsIgnoreCase("Rect")) {
            kanvasKu.drawRect(awalX, awalY, antarX, antarY, paintKu);
        } else if (bentuk.equalsIgnoreCase("Oval")) {
            kanvasKu.drawOval(awalX, awalY, antarX, antarY, paintKu);
        } else if (bentuk.equalsIgnoreCase("Line")) {
            kanvasKu.drawLine(awalX, awalY, antarX, antarY, paintKu);
        }
    }
}

```

17. Selamat, Anda sudah berhasil membuat app drawing sederhan. Tampilan pada emulator tampak seperti gambar di bawah ini.



Latihan, Tugas, dan Eksperimen

- Ikuti Langkah-langkah di atas sampai project anda dengan nama **Weeko5_[NIM]** dan Eksperimen dengan menggunakan custom View, Canvas, Paint, Bitmap, dan drawing dan clearing object pada kanvas (atau layar) berjalan dengan baik, dan berhasil menampilkan sesuai dengan yang diharapkan. Zip (kompres) file-file berikut dan submit ke isian tugas pada eLearning:
 - MainActivity.java

- 2) CustomView.java
 - 3) activity_main.xml
 - 4) styles.xml
 - 5) AndroidManifest.xml
2. Lakukan pengujian terhadap app anda dengan skenario kombinasi berbagai jenis bentuk dan warna serta klik pada ikon clear. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 6

(ANIMASI DAN MULTIMEDIA)

DESKRIPSI TEMA

- Animasi (Property, Drawable, dan Fisika)
- Multimedia
- Kamera

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang memiliki Animasi dengan mengendalikan property dari Objek, menggunakan beberapa gambar seperti melakukan flipping buku, menirukan gerakan fisika sederhana seperti spring dan fling (memutar objek). Disamping itu, mampu membuat Android app yang dapat menampilkan dan memutar video, serta mengintegrasikan penggunaan kamera untuk mengambil gambar dan video.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Membuat Animasi Property

App yang akan dibuat pada tutorial ini adalah sebuah app yang menampilkan animasi property dari sebuah lingkaran mengecil dan membesar serta mengulangi jumlah animasi tersebut. Berikut Langkah-langkah tutorial ini:

1. Buat Android Studio Project yang baru dengan nama **Weeko6A_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Buatlah sebuah Java Class pada folder **app > java > [domain].week6a_[NIM]** dengan nama **ViewAnimasi** yang meng-extends class **View**.

```
public class ViewAnimasi extends View {  
}
```

3. Seperti pada Lab Praktikum sebelumnya, sub class dari View wajib mengimplementasi Konstruktor, maka tambahkan konstruktor berikut pada class **ViewAnimasi** ini:

```
public class ViewAnimasi extends View {  
  
    public ViewAnimasi(Context context) {  
        super(context);  
    }  
    public ViewAnimasi(Context context, @Nullable AttributeSet attrs) {  
        super(context, attrs);  
    }  
}
```

4. Tambahkan beberapa constant, variable serta obyek-obyek pendukung melakukan Animasi property ini yang ditulis sebelum Konstruktor, yaitu:

```
private static final int DURASI = 4000;  
private static final long DELAY = 1000;  
private static final int PENGATURAN_WARNA = 5;  
private float mX;  
private float mY;  
private float mRadius;  
private final Paint mPaint = new Paint();  
private AnimatorSet mAnimatorSet = new AnimatorSet();
```

5. Method yang akan dilakukan pertama kali setelah menjalankan Konstruktor pada sebuah sub-class dari View adalah method **onSizeChanges(...)**. Buatlah object-object untuk membuat konfigurasi Animasi seperti membesar, mengecil, dan ulang pada method ini seperti pada contoh kode ini:

```

@Override
public void onSizeChanged(int w, int h, int oldw, int oldh) {
    ObjectAnimator membesar = ObjectAnimator.ofFloat(this,
        "radius", o, getWidth());
    membesar.setDuration(DURASI);
    membesar.setInterpolator(new LinearInterpolator());

    ObjectAnimator mengecil = ObjectAnimator.ofFloat(this,
        "radius", getWidth(), o);
    mengecil.setDuration(DURASI);
    mengecil.setInterpolator(new LinearOutSlowInInterpolator());
    mengecil.setStartDelay(DELAY);

    ObjectAnimator ulang = ObjectAnimator.ofFloat(this,
        "radius", o, getWidth());
    ulang.setStartDelay(DELAY);
    ulang.setDuration(DURASI);
    ulang.setRepeatCount(1);
    ulang.setRepeatMode(ValueAnimator.REVERSE);
    mAnimatorSet.play(membesar).before(mengecil);
    mAnimatorSet.play(ulang).after(mengecil);
}

```

6. Animasi Property yang digunakan pada tutorial ini adalah perubahan radius dari sebuah lingkaran dan memiliki nama property “radius” pada parameter **ObjectAnimator**, maka pada harus mengimplementasikan method **setRadius(...)** untuk mengganti property nilai global variable **mRadius** dan mengubah warna untuk menggambarkan lingkarannya dengan menggunakan method **setColor(...)** pada object global **mPaint**. Berikut contoh kode pada method **setRadius(...)**.

```

public void setRadius(float radius) {
    mRadius = radius;
    mPaint.setColor(Color.GREEN + (int) radius / PENGATURAN_WARNA);
    invalidate();
}

```

7. Pada sub-class View, proses penggambaran pada View dilakukan pada method **onDraw(...)** yang akan selalu dilakukan jika terjadi perubahan ukuran dan juga ada yang memanggil method **invalidate()**. **onDraw(...)** method pada tutorial ini sangat sederhana hanya menggambarkan lingkaran dengan titik pusat variable global **mX** dan **mY** dengan radius **mRadius**, serta style menggunakan object **mPaint**. Berikut kode method **onDraw(...)**:

```

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawCircle(mX, mY, mRadius, mPaint);
}
  
```

- Selanjutnya adalah menangani masukan dari user berupa klik pada layar. Jika user menyentuh pada layar, maka animasi akan dilakukan dengan titik pusat posisi user menyentuh layar. Interaksi dengan user ini dilakukan pada method **onTouchEvent(...)** dengan jenis **ACTION_DOWN**. Kode dibawah ini adalah implementasi dari method **onTouchEvent(...)**.

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    if (event.getActionMasked() == MotionEvent.ACTION_DOWN) {
        mX = event.getX();
        mY = event.getY();
        if(mAnimatorSet != null && mAnimatorSet.isRunning()) {
            mAnimatorSet.cancel();
        }
        mAnimatorSet.start();
    }
    return super.onTouchEvent(event);
}
  
```

- Langkah selanjutnya adalah memasukkan class **ViewAnimasi** ini ke dalam layout file **activity_main.xml** dengan mengganti obyek **textView** bawaan dengan object **ViewAnimasi** sehingga file **animasi_main.xml** seperti terlihat pada skrip XML berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <id.ac.umn.if633_1921.animasiproPERTY.ViewAnimasi
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

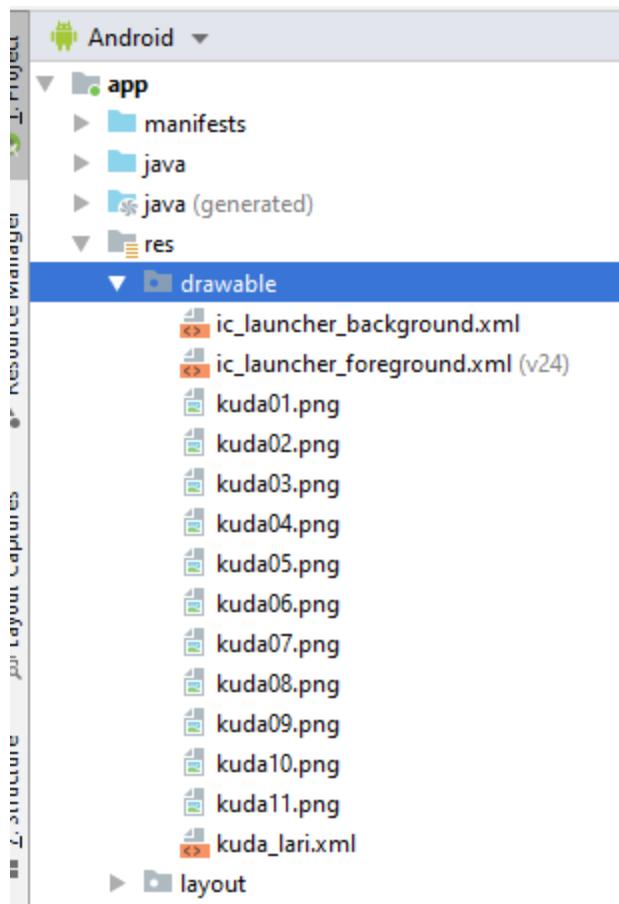
</androidx.constraintlayout.widget.ConstraintLayout>
  
```

10. Sementara untuk Java code pada MainActivity.java tidak ada perubahan, karena mengikuti default dengan menggunakan layout activity_main sebagai Content view nya.
11. Jalankan (run) project anda dan perhatikan apa yang terjadi jika anda klik atau sentuh layar pada app tersebut.

Tutorial: Membuat Animasi Drawable

Pada tutorial ini, kita akan membuat App yang menampilkan Animasi dari deretan gambar seperti pada flipping book dengan gambar yang sedikit perubahan sehingga tampak seperti bergerak. Deretan gambar kuda dapat di unduh dari tautan ini : <https://drive.google.com/open?id=1-NFtm8hFF3WBW5QNSPUedhSVbUJMozgi>. Ikuti Langkah-langkah berikut untuk membuat App sederhana tersebut:

1. Buat Android Studio Project yang baru dengan nama: **Weeko6B_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity** dan Layout Name: **activity_main**.
2. Ekstrak file **kuda_lari.zip** yang diunduh dari tautan di atas. Copy seluruh file png (kuda01.png sampai kuda11.png) ke dalam **folder app > res > drawable** pada project anda dengan cara **drag-and-drop**.
3. Masih pada folder yang sama (**app > res > drawable**). Klik kanan pilih **New > Drawable resource file**. Setelah jendela **New resource file** muncul isi nama dengan **kuda_lari**, lalu klik **OK**, maka pada **folder app > res > drawable** akan terdapat tambahan file **kuda01.png** sampai **kuda11.png** dan **kuda_lari.xml**, seperti terlihat pada gambar di bawah ini:



Catatan : nama file yang masukkan ke dalam folder app > res > drawable hanya boleh berisi deretan huruf kecil **a – z** (lower case), angka **(0 – 9)**, dan/atau **_** (underscore).

4. Buka dan edit file **kuda_lari.xml** untuk menambahkan animation-list yang berisi daftar gambar yang menyajikan animasi kuda berlari. Ganti tag **<selector ...> </selector>** dengan **<animasi-list> <item .../> </animation-list>** seperti terlihat pada skrip XML, berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<animation-list
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">
    <item android:drawable="@drawable/kudao1" android:duration="300" />
    <item android:drawable="@drawable/kudao2" android:duration="300" />
    <item android:drawable="@drawable/kudao3" android:duration="300" />
    <item android:drawable="@drawable/kudao4" android:duration="300" />
    <item android:drawable="@drawable/kudao5" android:duration="300" />
    <item android:drawable="@drawable/kudao6" android:duration="300" />
    <item android:drawable="@drawable/kudao7" android:duration="300" />
    <item android:drawable="@drawable/kudao8" android:duration="300" />
    <item android:drawable="@drawable/kudao9" android:duration="300" />
    <item android:drawable="@drawable/kudao10" android:duration="300" />
    <item android:drawable="@drawable/kudao11" android:duration="300" />
</animation-list>
  
```

Attribute **android:oneshot** pada tag “**animation-list**” dapat berisi “**true**” atau “**false**” untuk menentukan apakah animasi hanya dijalankan cukup satu kali atau berulang kali. “**false**” animasi akan dijalankan cukup satu kali atau berulang kali, “**false**” animasi akan dijalankan berulang-ulang (loop), “**true**” animasi akan dijalankan hanya satu kali. Attribute **android:duration** pada tag “**item**” menentukan durasi lama item tersebut ditampilkan dalam satuan millisecond.

5. Selanjutnya, sisipkan sebuah object **ImageView** pada layout untuk menampilkan animasi tersebut. Buka dan edit **activity_main.xml** dan ganti object **TextView** dengan object **ImageView** dengan nama **gambarKuda** sebagai mana terlihat pada skrip XML berikut:

```
<ImageView  
    android:id="@+id/gambarKuda"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
/>
```

6. Seperti tutorial-tutorial sebelumnya, kaitkan object **ImageView** yang ada pada layout **activity_main.xml** di atas dengan program java pada **MainActivity.java** dengan menambahkan variable **ImageView** , serta menambahkan object **AnimationDrawable** untuk dikaitkan dengan object animasi pada **kuda_lari.xml**. Animasi akan dijalankan jika user klik pada object **ImageView** tersebut dengan menangani event klik dengan menggunakan method **setOnClickListener(...)**. Berikut program java pada **MainActivity.java**.

```

public class MainActivity extends AppCompatActivity {
    AnimationDrawable animasiKuda;
    ImageView gambarKuda;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        gambarKuda = (ImageView) findViewById(R.id.gambarKuda);
        gambarKuda.setBackgroundResource(R.drawable.kuda_lari);
        animasiKuda = (AnimationDrawable) gambarKuda.getBackground();
        gambarKuda.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                animasiKuda.start();
            }
        });
    }
}

```

7. Jalankan (run) project anda dan perhatikan apa yang terjadi jika anda klik gambar kuda (object ImageView) app tersebut.

Tutorial: Membuat Animasi berbasis ilmu Fisika

Pada tutorial ini, kita akan membuat App yang menampilkan Animasi mengikuti animasi Gerakan fisika sehingga mendekati dengan realita. Ada dua jenis animasi fisika yang akan dibuat pada tutorial ini yaitu Spring dan Fling animasi. Animasi berbasis spring (per) akan meniru bagaimana sebuah benda tergantung pada per, kemudian kita tarik dan lepas benda tersebut, maka benda akan terus bergerak atas – bawah sampai mencapai titik equilibrium. Animasi berbasis fling menggunakan tenaga gesekan yang proporsional dengan kecepatannya. Ikuti Langkah tutorial berikut untuk dapat membuat animasi berbasis kaidah-kaidah ilmu fisika.

1. Buat Android Studio Project yang baru dengan nama: **Weeko6C_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Langkah awal yang harus dilakukan adalah mengaktifkan dukungan Dynamic Animation pada App anda dengan menambahkan skrip gradle berikut pada file **build.gradle(Module: app)** pada folder **Gradle Scripts** di dalam lingkup **dependencies{ ... }**:

```
implementation "com.android.support:support-dynamic-animation:29"
```

Catatan: angka 29 adalah versi Android API yang akan digunakan sesuaikan dengan nilai **compileSdkVersion** yang disebutkan sebelumnya.

3. Buatlah sebuah Java Class pada folder **app > java > [domain].week6c_[NIM]** dengan nama **TombolSpring** yang meng-extend AppCompatButton. Berikut definisi class ini:

```
public class TombolSpring extends AppCompatButton {
```

4. Seperti sub-class dari **View** sub-class dari **AppCompatButton** juga mengharuskan kita untuk menyediakan konstruktor. Tambahkan tiga konstruktor dengan cara menggunakan fasilitas Auto-fix dari Android Studio, atau menuliskan konstruktor berikut pada class ini:

```
public TombolSpring(Context context) {
    super(context);
}
public TombolSpring(Context context, AttributeSet attrs) {
    super(context, attrs);
}
public TombolSpring(Context context, AttributeSet attrs,
                     int defStyleAttr) {
    super(context, attrs, defStyleAttr);
}
```

5. Selanjutnya tambahkan pembuatan object **SpringAnimation** di dalam method **onTouchEvent(...)** dimana akan dijalankan pada saat user menyentuh object **TombolSpring** ini. Konstruktor **SpringAnimation** membutuhkan tiga parameter: **context** → umumnya object itu sendiri ("this"); jenis animasi dinamis → pada tutorial ini menggunakan perubahan property Y yaitu pergerakan vertical ("DynamicAnimation.Y"); dan posisi diam (rest position) dari object ini → kita gunakan nilai posisi dari object ini yaitu nilai koordinat Y ("this.getY()"). Berikut kode method **onTouchEvent(...)**:

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    switch (event.getActionMasked()) {
        case MotionEvent.ACTION_DOWN:
            final SpringAnimation anim = new
SpringAnimation(this,
                (FloatPropertyCompat) DynamicAnimation.Y,
                this.getY());
            anim.setStartVelocity(10000);
            anim.getSpring().setStiffness(STIFFNESS_LOW);
            anim.start();
            break;
        default:
    }
    return super.onTouchEvent(event);
}
```

6. Buatlah sebuah Java Class pada folder `app > java > [domain].week6c_[NIM]` dengan nama **TombolFling** yang meng-extend AppCompatButton. Berikut definisi class ini:

```
public class TombolFling extends AppCompatButton {
```

7. Tambahkan tiga konstruktor dengan menuliskan konstruktor-konstruktor berikut pada class ini:

```
public TombolFling(Context context) {
    super(context);
}
public TombolFling(Context context, AttributeSet attrs) {
    super(context, attrs);
}
public TombolFling(Context context, AttributeSet attrs,
                    int defStyleAttr) {
    super(context, attrs, defStyleAttr);
}
```

8. Seperti class TombolSpring, class ini juga akan dijalankan pada saat user menyentuh object ini, untuk hal ini kia harus mengimplementasikan method `onTouchEvent(...)`. Konstruktor **FlingAnimation** membutuhkan dua parameter: `context` → umumnya object itu sendiri ("this"); dan jenis animasi dinamis → pada tutorial ini menggunakan rotasi pada sumbu X ("DynamicAnimation.ROTATION.X"). Selanjutnya tentukan kecepatan awal dengan menggunakan method `setStartVelocity(...)`, dan nilai gesekan atau friksi dengan menggunakan method `setFriction(...)`. Method `onTouchEvent(...)` pada class **TombolFling** sebagai berikut:

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    switch (event.getActionMasked()) {
        case MotionEvent.ACTION_DOWN:
            FlingAnimation fling = new FlingAnimation(
                this, DynamicAnimation.ROTATION_X);
            fling.setStartVelocity(150)
                .setFriction(0.11f)
                .start();
            break;
        default:
    }
    return super.onTouchEvent(event);
}
```

9. Langkah berikutnya adalah memasukkan dua object class ini ke dalam file layout **activity_main.xml** dengan mengganti dari object **TextView** dengan skrip dibawah ini:

```
<id.ac.umn.week06c_067806.TombolSpring
    android:id="@+id/myspringbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Halo\nSpring"
    android:textSize="36sp"
    app:layout_constraintBottom_toTopOf="@+id/myflingbutton"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.434" />

<id.ac.umn.week06c_067806.TombolFling
    android:id="@+id/myflingbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Halo\nFling"
    android:textSize="36sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/myspringbutton"
    app:layout_constraintVertical_bias="0.434" />
```

10. Sementara untuk Java code pada **MainActivity.java** tidak ada perubahan, karena mengikuti default dengan menggunakan layout **activity_main** sebagai Content view nya.
11. Jalankan (run) project anda dan perhatikan apa yang terjadi jika anda klik atau sentuh layar pada tombol Spring dan/atau Tombol Fling.

Latihan, Tugas, dan Eksperimen:

1. Ikuti Langkah-langkah di atas sampai ke tiga project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan peribahan dan submit ke isian tugas pada eLearning:
2. Lakukan pengujian terhadap app anda dengan mengklik object-object yang disampaikan pada tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 7

(MULTIMEDIA DAN KAMERA)

DESKRIPSI TEMA

- Multimedia
- Kamera

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat menampilkan dan memutar video, serta mengintegrasikan penggunaan kamera untuk mengambil gambar dan video.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Menggunakan Kamera

App yang akan dibuat pada tutorial ini adalah sebuah app yang dapat menggunakan kamera untuk mengcapture foto maupun video. Berikut Langkah-langkah tutorial ini:

1. Buat Android Studio Project yang baru dengan nama: **Weeko7A_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Pemanfaatan kamera, dengan cara mengubah file **AndroidManifest.xml** pada folder **app > manifest** dengan menambahkan skrip XML berikut:

```
<uses-feature android:name="android.hardware.camera"
    android:required="true" />
```

3. Selanjutnya buka dan layout file **activity_main.xml** dengan mengganti obyek **textView** bawaan dengan menyiapkan **ImageView** dan **VideoView** untuk menampilkan hasil foto dan/atau video serta dua buah tombol untuk mengaktifkan kamera dengan memanggil app bawaan android yang menangani kamera sehingga file **activity_main.xml** seperti terlihat pada skrip XML berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="200dp" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ambil Foto" />

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="200dp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ambil Video" />
</LinearLayout>

```

4. Tambahkan penanganan kamera pada Java code pada **MainActivity.java** dengan menghubungkan ImageView, VideoView, dan dua tombol tersebut dengan membuat objek-objek dan memanggil findViewById seperti kode berikut:

```

public class MainActivity extends AppCompatActivity {
    private Button foto;
    private Button video;
    private ImageView kotakFoto;
    private VideoView kotakVideo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        foto = findViewById(R.id.button);
        video = findViewById(R.id.button2);
        kotakFoto = findViewById(R.id.imageView);
        kotakVideo = findViewById(R.id.videoView);
    }
}

```

5. Pemanggilan app bawaan Kamera dilakukan dengan penggunaan Intent Implicit **MediaStore.ACTION_IMAGE_CAPTURE** untuk ambil foto dan **MediaStore.ACTION_VIDEO_CAPTURE** untuk ambil video. Dalam hal ini MainActivity sebagai pemanggil Intent ini harus dapat menerima Kembali hasil foto dan video yang dihasilkan oleh Intent tersebut, maka digunakan method **startActivityForResult(...)**. Berikut adalah penambahan kode untuk MainActivity dalam menangani hal ini:

Variable constant

```
static final int REQUEST_IMAGE_CAPTURE = 1;
static final int REQUEST_VIDEO_CAPTURE = 2;
```

Menghubungkan VideoView dengan MediaController yang akan memutar atau memainkan hasil video, serta penanganan jika tombol Foto dan tombol Video diklik yang diletakkan pada method **onCreate(...)**:

```
MediaController controller = new MediaController(this);
controller.setMediaPlayer(kotakVideo);
kotakVideo.setMediaController(controller);
foto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent takePictureIntent = new
            Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(takePictureIntent,
                REQUEST_IMAGE_CAPTURE);
        }
    }
});

video.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent takeVideoIntent = new
            Intent(MediaStore.ACTION_VIDEO_CAPTURE);
        if (takeVideoIntent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(takeVideoIntent,
                REQUEST_VIDEO_CAPTURE);
        }
    }
});
```

6. Selanjutnya adalah mengimplementasi onActivityResult(...) method callback untuk menerima hasil dari Foto maupun Video, berikut kode dari method ini:

```
@Override
protected void onActivityResult(int requestCode,int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        kotakFoto.setImageBitmap(imageBitmap);
    } else if (requestCode == REQUEST_VIDEO_CAPTURE && resultCode == RESULT_OK) {
        Uri videoUri = data.getData();
        kotakVideo.setVideoURI(videoUri);
        kotakVideo.seekTo(100);
        kotakVideo.start();
    }
}
```

7. Jalankan (run) project anda dengan menggunakan Smartphone sebagai target devicenya agar kamera dapat digunakan langsung dan perhatikan apa yang terjadi jika anda klik atau sentuh layar pada tombol Foto dan tombol Video pada app tersebut.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai ke project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning;
2. Lakukan pengujian terhadap app anda dengan mengklik tombol Foto dan Video. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 8

(ANDROID STORAGE, SAVED INSTANCE AND SHARED PREFERENCES)

DESKRIPSI TEMA

- Android Storage
 - Temporary (cache)
 - Internal
 - External
- Saved Instance
- Shared Preferences

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat memanfaatkan Storage baik sementara / temporary / cache, internal dan external. Selain itu mahasiswa juga mampu membuat Android app yang dapat menyimpan data session sebagai saved instance dan shared preferences.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: TextEditor dan Storage

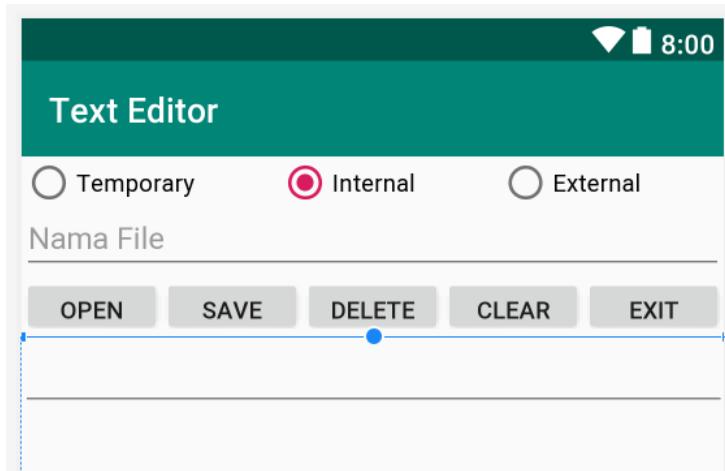
App yang akan dibuat tutorial ini adalah sebuah Text Editor sederhana yang dapat menyimpan plain text yang diketik pada Text area (multi-lines Edit Text widget). Tempat penyimpanan yang digunakan adalah temporary folder (cache), internal storage dan/atau External Storage (SDCard). Berikut langkah-langkah tutorial ini:

1. Buat Android Studio Project yang baru dengan nama: **Weeko8A_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Project yang dibuat membutuhkan pembacaan dan penulisan file ke external storage (SD Card), sehingga membutuhkan izin (permission) dari Android runtime yang dapat dilakukan dengan menambahkan XML skrip berikut pada file **AndroidManifest.xml** pada folder **app > manifest**:

```
<uses-permission android:name =  
    "android.permission.WRITE_EXTERNAL_STORAGE"/>
```

3. Selanjutnya membuat dan mengatur layout project ini yang dapat mengakomodasi beberapa widget yang dibutuhkan seperti:
 - a. Sebuah **RadioGroup** yang berisi tiga buah **RadioButton** untuk memilih lokasi / folder / direktori tempat menyimpan dan mengakses file. Ketiga pilihan tersebut meliputi: Temporary, Internal dan External;
 - b. Sebuah isian text untuk nama file menggunakan **EditText** widget
 - c. Lima buah tombol menggunakan widget **Button** untuk: **Open** → menampilkan daftar file yang ada pada storage sesuai pilihan radio button untuk dipilih dan akan menampilkan isi plain text file tersebut. **Save** → menyimpan isi teks yang ada pada text area (multiline **EditText**) ke file yang tertulis pada isian nama file pada folder sesuai pilihan **radioButton**; **Delete** → menghapus sebuah file yang memiliki nama pada isian nama file pada folder sesuai pilihan **radioButton**; **Clear** → mengosongkan isian pada nama file dan text area; **Exit** → keluar dari App dan menghapus semua files yang ada pada direktori sementara (cache).
 - d. Sebuah Text Area (multi-line **EditText** widget) yang berisi Teks yang akan disimpan ke file dan untuk menampilkan isi dari File (pilihan tombol Open).

Tampilan layout yang diharapkan pada tutorial ini adalah seperti pada gambar di bawah ini:



XML skrip untuk menghasilkan layout seperti di atas adalah sebagai berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="0dp"
        android:layout_height="30dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
        <RadioGroup
            android:id="@+id/rgJenis"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="fill_horizontal"
            android:orientation="horizontal">
            <RadioButton
                android:id="@+id/rbTemp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Temporary" />
            <RadioButton
                android:id="@+id/rbInternal"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:checked="true"
                android:text="Internal" />
            <RadioButton
                android:id="@+id/rbExternal"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="External" />
        </RadioGroup>
    </FrameLayout>
    <EditText
        android:id="@+id/etNamaFile"
        android:layout_width="0dp"
        android:layout_height="45dp"
        android:hint="Nama File"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/frameLayout" />

```

```

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="0dp"
    android:layout_height="40dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etNamaFile">
    <Button
        android:id="@+id	btnOpen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="openFile"
        android:text="Open" />
    <Button
        android:id="@+id	btnSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="saveFile"
        android:text="Save" />
    <Button
        android:id="@+id	btnDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="deleteFile"
        android:text="Delete" />
    <Button
        android:id="@+id	btnClear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="clearText"
        android:text="Clear" />
    <Button
        android:id="@+id	btnExit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="keluarApp"
        android:text="Exit" />
</LinearLayout>

<ScrollView
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <EditText
        android:id="@+id/etText"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="start|fill|top"
        android:inputType="textMultiLine" />
</LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Pada masing-masing object **Button** penanganan klik didefinisikan pada layout ini dengan memanfaatkan attribut **android:onClick="namaMethod"**. Pesan Error akan muncul pada kelima tombol di atas karena method yang disebutkan pada attribute **android:onClick** belum tersedia pada **MainActivity**. Pesan error ini akan hilang Ketika method-method sudah dideklarasikan dengan pola **public void namaMethod(View view) [...]**.

- Berikutnya adalah membuka dan menambahkan kode-koe Java pada **MainActivity** untuk mendeklarasikan dan membuat serta mengaitkan object-object yang telah disediakan pada layout sehingga dapat terhubung dan dapat menangani dan berinteraksi dengan user. Object-object yang perlu dibuatkan penanganannya adalah: **RadioGroup** → untuk mengetahui jenis File / Direktori yang digunakan (Temporary, Internal, atau External). **EditText** → untuk nama file dan isiin text. Sementara untuk **Button** penanganan Klik nya sudah ada method-method yang sebutkan pada attribut **android:onClick**, sementara untuk **radioButton** tidak diharuskan karena cukup menggunakan **radioGroup** yang mengetahui yang dipilih. Berikut kode Java untuk keperluan di atas:

```

public class MainActivity extends AppCompatActivity {
    private RadioGroup rgJenis;
    private EditText etFileName;
    private EditText etText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rgJenis = (RadioGroup) findViewById(R.id.rgJenis);
        etFileName = (EditText) findViewById(R.id.etNamaFile);
        etText = (EditText) findViewById(R.id.etText);
    }
}

```

Tambahkan method-method yang disebutkan pada attribute **android:onClick** pada **MainActivity.java** seperti kode beikut:

```

public void openFile(View view){ }
public void saveFile(View view){ }
public void deleteFile(View view) { }
public void keluarApp(View view){
    finishAffinity();
}
  
```

Untuk keluar dari App kita dapat menggunakan method **finishAffinity()**.

5. Penanganan pilihan jenis storage diperlukan empat variable dengan tipe data **File** yang menyimpan referensi object terhadap tempat penyimpanan (storage): **tempDir** -> cache [method **getChaceDir()**]; **lokalDir** -> internal storage [methid **getFilesDir()**]; **extDir** -> external Storage [method **getExternalDir()**]; dan **curDir** -> current atau pilihan storage sesuai dengan **radioButton** yang dipilih object radioGroup perlu ditambahkan penanganan listener **OnCheckedChangeListener** sehingga setiap perubahan pilihan akan secara otomatis referensi variable **curDir** menunjuk pada storage yang tepat. Berikut kode java yang perlu ditambahlan pada deklarasi variable:

```

private File tempDir;
private File lokalDir;
private File extDir;
private File curDir;
  
```

pada **onCreate()** method:

```

tempDir      = getCacheDir();
lokalDir     = getFilesDir();
if(Environment.MEDIA_MOUNTED.equals(
    Environment.getExternalStorageState())) {
    extDir      = getExternalFilesDir(null);
} else {
    findViewById(R.id.rbExternal).setEnabled(false);
    extDir = null;
}
curDir = lokalDir;
rgJenis.setOnCheckedChangeListener(new
    RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            String pilihan = ((RadioButton)
                findViewById(rgJenis.getCheckedRadioButtonId()))
                .getText().toString();
            if(pilihan.equalsIgnoreCase("Temporary")) curDir = tempDir;
            else if(pilihan.equalsIgnoreCase("Internal")) curDir = lokalDir;
            else curDir = extDir;
        }
    });
  
```

6. Selanjutnya melengkapi beberapa method yang belum lengkap, mulai dari yang paling sederhana yaitu **clearText** yang menghapus isi dari EditText nama file dan isi textArea sebagai berikut:

```
public void clearText(View view) {
    etText.setText("");
    etFileName.setText("");
}
```

7. Method **saveFile** akan menyimpan text yang ada pada text area ke dalam file sesuai dengan nama file dan storage yang dipilih. Untuk menuliskan text ke dalam file dapat menggunakan object **OutputStreamWriter** dengan konstruktor yang menerima input object dari class **FileOutputStream** method **write()** untuk menuliskan ke file output dan method **close()** untuk menutup dan meng-commit penyimpanan text tersebut. Konstruktor dan method di atas memiliki kemungkinan menghasilkan run-time Exception seperti **FileNotFoundException** dan **IOException**, sehingga diperlukan adanya **try-and-catch** block. Berikut adalah kode untuk method **saveFile**:

```
public void saveFile(View view){
    String nFile = etFileName.getText().toString();
    String isiText = etText.getText().toString();
    if(nFile.length() > 0 && isiText.length() > 0 && curDir != null){
        File file = new File(curDir,nFile);
        try{
            OutputStreamWriter writer = new OutputStreamWriter(
                new FileOutputStream(file));
            writer.write(isiText);
            writer.close();
            Toast.makeText(this,"Text sudah tersimpan",
            Toast.LENGTH_LONG).show();
        } catch (FileNotFoundException e){
            Toast.makeText(this,"File tidak Ditemukan",
            Toast.LENGTH_LONG).show();
        }catch (IOException e){
            Toast.makeText(this,"Ada kesalahan I/O",
            Toast.LENGTH_LONG).show();
        }
    }
}
```

8. Method **openFile** memiliki tugas yang cukup kompleks mulai dari menampilkan daftar file yang ada pada storage yang dipilih dengan menggunakan **PopupMenu**, setelah user memilih file pada daftar tersebut dilanjutkan dengan pembacaan isi dari file dan tampilkan ke dalam text area. Menu pada **PopupMenu** akan berisi daftar file pada storage yang dipilih sehingga sifatnya dinamis dan diisi

menggunakan program tetapi agar dapat menampilkan object PopMenu dibutuhkan menu file pada resources direktori meskipun kosong (tidak ada menu item). Untuk membuat file XML menu, pertama klik kanan pada folder **app > res** pilih **New > Android Resource Directory**, lalu pilih **Resource type : menu** dan **Directory name: menu** kemudian klik **OK** pada Dialog Window **New Resource Directory**. Setelah itu buat file XML menu dengan klik kanan pada folder **app > res > menu** dan pilih **New > Menu resource file**, isi **menu_kosong** pada isian **File Name** dan klik **OK** pada dialog window **New Resource File**. File **menu_kosong.xml** akan muncul pada folder **app > res > menu**. File ini tidak perlu untuk dilakukan perubahan karena isi menu akan dibuat dengan program secara dinamis.

9. Selanjutnya tambahkan variable dengan tipe data **Context**, tipe data **Button** (untuk tombol Open) dan variable statis dengan tipe data **PopupMenu**, seperti berikut:

```
private Context context;
private Button btnOpen;
private static PopupMenu pilihFile;
```

serta konfigurasi PopMenu di dalam method **onCreate()**, termasuk pengendalian jika user memilih salah satu item (file dari daftar yang ditampilkan) yang akan diisikan ke dalam isian nama file, sebagai berikut:

```
context = this;
btnOpen = (Button) findViewById(R.id.btnOpen);
pilihFile = new PopupMenu(context,btnOpen);
pilihFile.getMenuInflater().inflate(R.menu.menu_kosong,
pilihFile.getMenu());
pilihFile.setOnMenuItemClickListener(
new PopupMenu.OnMenuItemClickListener() {
@Override
public boolean onMenuItemClick(MenuItem item) {
etFileName.setText(item.getTitle().toString());
etFileName.setText("");
etText.setText("");
return true;
}
});
});
```

10. Kemudian lengkapi method **openFile**, dengan melakukan tindakan membuka file sesuai dengan nama file dan jenis storage serta membaca isi file tersebut dan menampilkan isinya ke text area. Untuk membaca file digunakan urutan pembuatan object **File → FileInputStream → InputStreamReader → BufferedReader**. Object dari class **BufferedReader** memiliki method **read()** [membaca per huruf] dan **readLine()** yang membaca baris-per-baris dan menambahkan baris

tersebut ke Text area. Beikut implementasi dari method **openFile** yang akan dijalankan pada saat user klik tombol **OPEN**:

```

public void openFile(View view){
    File[] files = null;
    if(curDir != null) files = curDir.listFiles();
    if(files != null) {
        int n = files.length;
        pilihFile.getMenu().clear();
        for(int i = 0; i < files.length; i++)
            pilihFile.getMenu().add(files[i].getName());
        pilihFile.show();
        bacaFile();
    } else {
        Toast.makeText(context,"Ada masalah akses folder " +
                    "atau folder masih kosong",Toast.LENGTH_LONG).show();
    }
}
private void bacaFile(){
    if(etFileName.getText().toString().length()>0){
        File file = new File(curDir,etFileName.getText().toString());
        String isiFile = "";
        try{
            InputStream inStream = new FileInputStream(file);
            if(inStream != null){
                InputStreamReader isReader = new
                    InputStreamReader(inStream);
                BufferedReader bReader = new BufferedReader(isReader);
                String terimaString = "";
                StringBuilder sb = new StringBuilder();
                while( (terimaString = bReader.readLine()) != null){
                    sb.append(terimaString).append("\n");
                }
                inStream.close();
                isiFile = sb.toString();
                etText.setText(isiFile);
            }
        }catch (FileNotFoundException e){
            Toast.makeText(context,"File tidak Ditemukan",
                    Toast.LENGTH_LONG).show();
        }catch (IOException e){
            Toast.makeText(context,"Error di I/O ", Toast.LENGTH_LONG)
                    .show();
        }
    }
}

```

11. Selanjutnya method **deleteFile** yang menghapus file yang tertera pada isian namaFile pada storage yang dipilih. Untuk temporary (cache) dan eksternal file menggunakan method **delete()** dan untuk internal storage menggunakan method **context.deleteFile()**. Berikut kode untuk implementasi method **deleteFile**:

```

public void deleteFile(View view) {
    if(etFileName.getText().toString().length() > 0) {
        boolean sukses = false;
        if( curDir != null && curDir == lokalDir){
            sukses = context.deleteFile(etFileName.getText()
                .toString());
        } else {
            sukses = new File(curDir, etFileName.getText().toString())
                .delete();
        }
        if (sukses) Toast.makeText(context, "File berhasil dihapus",
            Toast.LENGTH_LONG).show();
        else Toast.makeText(context, "File GAGAL dihapus",
            Toast.LENGTH_LONG).show();
        etFileName.setText("");
        etText.setText("");
    }
}

```

12. Langkah terakhir adalah melakukan pembersihan file yang tersimpan pada cache, pada saat user klik tombol “Exit” atau klik tombol bawaan “Back”, maka app akan keluar dari sistem Android run-time, untuk menghapus seluruh file yang ada pada cache dibutuhkan untuk meng-override method **onDestroy()** dengan menghapus semua file pada cache, berikut program pada method **onDestroy()**:

```

@Override
protected void onDestroy() {
    super.onDestroy();
    // hapus semua file di cache
    File[] tempFiles = tempDir.listFiles();
    for(File tempFile : tempFiles){
        if(tempFile.isFile()) tempFile.delete();
    }
}

```

13. Jalankan (run) project anda serta coba lakukan beberapa percobaan dengan memilih **radioButton** bergantian, penamaan file yang berbeda-beda dan klik tombol secara bergantian, perhatikan dan catat bagaimana behaviour dari app anda.

Catatan: untuk method `openFile` terjadi jeda antara pemilihan file dengan `popMenu` serta pengisian text area dikarenakan pemanggilan method `show()` dan `bacaFile()` hampir secara bersamaan, sementara method `bacaFile()` membaca isian nama file yang diisi hasil pilihan user pada `popMenu()`. Untuk itu setelah klik Open dan pilih file klik ulang Open tanpa pilih file di `popMenu`.

Tutorial: Saved Instance dan Shared Preference

App yang akan dibuat pada tutorial ini adalah sebuah app yang memanfaatkan penggunaan Saved Instance dan Shared Preferences yang dapat menyimpan data pada sesi (session) di App Android dalam bentuk pasangan Key-Value. Berikut Langkah-langkah turorial ini:

1. Buat Android Studio Project yang baru dengan nama: **Weeko8B_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity** dan Layout Name: **activity_main**.
2. App ini memiliki dua data yang akan disimpan pada session yaitu **counter** dan **warna**. Counter adalah nilai angka mulai dari 0 (nol) dan bertambah ketika user klik tombol “Tombol”, serta warna adalah warna background dari object **TextView** yang menampilkan counter. Perubahan warna menggunakan klik Button. Tampilan dari app ini menyerupai tampilan berikut:



3. Buka dan modifikasi file layout activity_main.xml untuk mendapatkan tampilan seperti di atas,

Layout utama menggunakan ConstraintLayout yang berisi:

- TextView yang menampilkan counter.
- LinearLayout vertical yang berisi:
 - LinearLayout horizontal dengan empat buah tombol warna.
 - LinearLayout horizontal dengan dua buah tombol “TAMBAH” DAN “RESET”.

Berikut skrip xml untuk file layout activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/tvCounter"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:background="@color/default_background"
        android:gravity="center"
        android:text="0"
        android:textColor="#FFFFFF"
        android:textSize="200sp"
        app:layout_constraintDimensionRatio="1:1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvCounter">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <Button
                android:id="@+id/button8"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:background="@color/hitam"
                android:onClick="gantiBackground"
                android:text="HITAM"
                android:textColor="#FFFFFF" />
        
```

```
<Button
    android:id="@+id/button9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:background="@color/merah"
    android:onClick="gantiBackground"
    android:text="MERAH"
    android:textColor="#FFFFFF" />
<Button
    android:id="@+id/button7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:background="@color/biru"
    android:onClick="gantiBackground"
    android:text="BIRU"
    android:textColor="#FFFFFF" />
<Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:background="@color/hijau"
    android:onClick="gantiBackground"
    android:text="HIJAU"
    android:textColor="#FFFFFF" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <Button
        android:id="@+id/button10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="tambahCounter"
        android:text="TAMBAH" />
    <Button
        android:id="@+id/button11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="reset"
        android:text="RESET" />
</LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Catatan: definisi warna @color hitam, @color=merah, @color=biru, @color=hijau, dan @color/default.background dilakukan dengan cara menambahkan skrip XML berikut pada file **color.xml** di folder **app>res>values**:

```
<color name="hitam">#000000</color>
<color name="merah">#CC0000</color>
<color name="biru">#0099CC</color>
<color name="hijau">#669900</color>
<color name="default_background">#E0E0E0</color>
```

Attribut **android:onClick** pada tombol warna hitam, merah, biru, dan hijau mereferensi pada method yang sama yaitu **gantiBackground** dimana akan membaca warna background dari tombol warna dan mengaplikasikan warna tersebut ke object **TextView** yang menampilkan counter.

4. Buka da edit file java **MainActivity.java** membuat object **TextView** yang ada pada layout dan implementasi method **gantiBackground()**, **tambahCounter()** dan **reset()** serta deklarasi variable **mCount** dan **mWarna**. Berikut deklarasi variable pada **MainActivity.java**:

```
private int mCount = 0;
private int mWarna;
private TextView tvCounter;
private Context context;
```

dan pembuatan object pada method **onCreate()**:

```
context = this;
tvCounter = (TextView) findViewById(R.id.tvCounter);
```

serta implementasi method di atas:

```
public void gantiBackground(View view) {
    int warna = ((ColorDrawable) view.getBackground()).getColor();
    mWarna = warna;
    tvCounter.setBackgroundColor(warna);
}
public void tambahCounter(View view) {
    mCount++;
    tvCounter.setText(String.valueOf(mCount));
}
public void reset(View view) {
    mCount = 0;
    tvCounter.setText(String.valueOf(mCount));
    mWarna = ContextCompat.getColor(context, R.color.default_background);
    tvCounter.setBackgroundColor(mWarna);
}
```

Pada tahapan ini App sudah dapat dijalankan. Jalankan (run) app dan tambahkan counter serta ganti background warna, setelah itu rotasi device anda dan perhatikan apa yang, pada saat App dirotasi, terjadi perubahan sesi pada app dan variable counter (mCount) dan warna (mWarna) Kembali kepada keadaan awal, karena tidak ada data yang disimpan pada sesi (session) App.

5. Langkah berikutnya kita akan menambahkan penyimpanan data pada sesi (session) App menggunakan Saved Instance state yaitu data counter dan warna background. Untuk keperluan ini dibutuhkan dua konstan **COUNTER_KEY** dan **WARNA_KEY** sebagai key dari data yang akan disimpan pada sesi, Berikut deklarasi konstan tersebut:

```
private final String COUNTER_KEY = "counter";
private final String WARNA_KEY = "warna";
```

Untuk mengisi nilai (value) untuk masing-masing key sehingga dapat disimpan pada sesi menggunakan Saved Instance adalah dengan meng-override method **onPause()** dan panggil method **putType()**, karena nilai yang disimpan adalah nilai tipe data int, maka gunakan method **.putInt(Key, Value)**. Berikut kode untuk override method **onSaveInstanceState**:

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt(COUNTER_KEY, mCount);
    outState.putInt(WARNA_KEY, mWarna);
}
```

Sementara untuk membaca isi dari data yang disimpan pada sesi menggunakan Saved Instance dilakukan di dalam method **onCreate()** dengan memanggil method **getType()**. Tipe data yang disimpan atau diisikan adalah tipe data int, sehingga untuk mengambil juga menggunakan method **getInt(...)**. Berikut penambahan kode Java pada **onCreate()**:

```
if (savedInstanceState != null) {
    mCount = savedInstanceState.getInt(COUNTER_KEY);
    if (mCount != 0) {
        tvCounter.setText(String.valueOf(mCount));
    }
    mWarna = savedInstanceState.getInt(WARNA_KEY);
    tvCounter.setBackgroundColor(mWarna);
}
```

6. Pada tahap ini App sudah menyimpan dua buah data counter dan warna pada sesi dengan menggunakan Saved Instance state. Jalankan (run) app, uji app dengan menambahkan counter dan

ganti warna lalu ganti rotasi (putar) device anda. Perhatikan dan catat apa yang terjadi, seharusnya pada saat device dirotasi warna dan nilai counter tidak kembali ke nilai sebelumnya, karena pada saat rotasi, Activity mengalami perubahan dan data counter dan warna disimpan di sesi. Ketika device dikembalikan posisi semula, activity kembali aktif atau resumed dan akan memanggil ulang method `onCreate()` dan membaca data counter dan warna pada sesi yang telah disimpan sebelumnya oleh method `onSaveInstanceState()`.

Lakukan exit dari app dengan klik tombol Back dan buka kembali app anda. Perhatikan dan catat apa yang terjadi? App tidak menampilkan counter dan warna terakhir sebelum App keluar, tetapi nilai count kembali nol dan warna kembali ke default color, karena data counter dan warna yang disimpan pada sesi menggunakan Saved Instance hanya aktif selama app atau sesi berjalan (sampai app keluar). Untuk mempertahankan nilai pada sesi dapat menggunakan Shared Preference yang dijelaskan Langkah selanjutnya.

7. Buka dan edit file MainActivity.java, tambahkan variable dengan tipe data SharedPreferences dan String nama file yang menyimpan data, sebagai berikut:

```
private SharedPreferences mPreferences;
private String sharedPrefFile;
```

Implementasi menyimpan data ke Shared Preference dilakukan dengan meng-override method `onPause()` menggunakan `SharedPreference.Editor` object dan method `putType(...)`. Seperti pada Saved Instance method yang digunakan `putInt(...)` karena tipe data untuk counter dan warna adalah integer dilanjutkan dengan memanggil method `apply()`. Berikut implementasi `onPause()`:

```
@Override
protected void onPause() {
    super.onPause();
    SharedPreferences.Editor preferencesEditor = mPreferences.edit();
    preferencesEditor.putInt(COUNTER_KEY, mCount);
    preferencesEditor.putInt(WARNA_KEY, mWarna);
    preferencesEditor.apply();
}
```

Dengan diimplementasikan `onPause()` maka method `onSaveInstanceState(...)` dapat dihapus karena sudah tidak dibutuhkan lagi. Sementara pada method `onCreate()` seluruh block `if(savedInstanceState != null)` dapat dihapus dan digantikan dengan setting file tempat menyimpan Shared Preference dan membaca data yang telah disimpan pada Shared Preference saat

menjalankan method **onPause()**. Berikut kode untuk setting nama file dan pembacaan data di dalam method **onCreate()**:

```
sharedPrefFile = context.getPackageName();  
mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);  
mCount = mPreferences.getInt(COUNTER_KEY, 0);  
tvCounter.setText(String.valueOf(mCount));  
mWarna = mPreferences.getInt(WARNA_KEY, mWarna);  
tvCounter.setBackgroundColor(mWarna);
```

8. Jalankan (run) app, lakukan eksperimen dengan menambah counter dan mengganti warna background, memutar device, dan keluar App dan jalankan kembali app tersebut. Perhatikan dan catat apa yang terjadi.

Latihan, Tugas, dan Eksperimen:

1. Ikuti Langkah-langkah di atas sampai project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas ke eLearning.
2. Lakukan pengujian terhadap app anda sesuai petunjuk tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 9

(SQLITE DATABASE DAN ROOM)

DESKRIPSI TEMA

- SQLite Database
 - Operasi SQLite Database (CRUD: Create, Read, Update, Delete)
- Room
 - Entity
 - DAO (Data Access Object)
 - LiveData
 - Room Database
 - Repository
 - ViewModel

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat memanfaatkan penyimpanan pada SQLite database lokal dengan mengintegrasikan menggunakan Room. App yang akan dibuat menyimpan data-data mahasiswa secara lokal di SQLite database yang kemudian dapat dilakukan operasi seperti membuat database, membaca, mengedit, dan menghapus baris dalam tabel di SQLite database.

PENUNJANG PRAKTIKUM

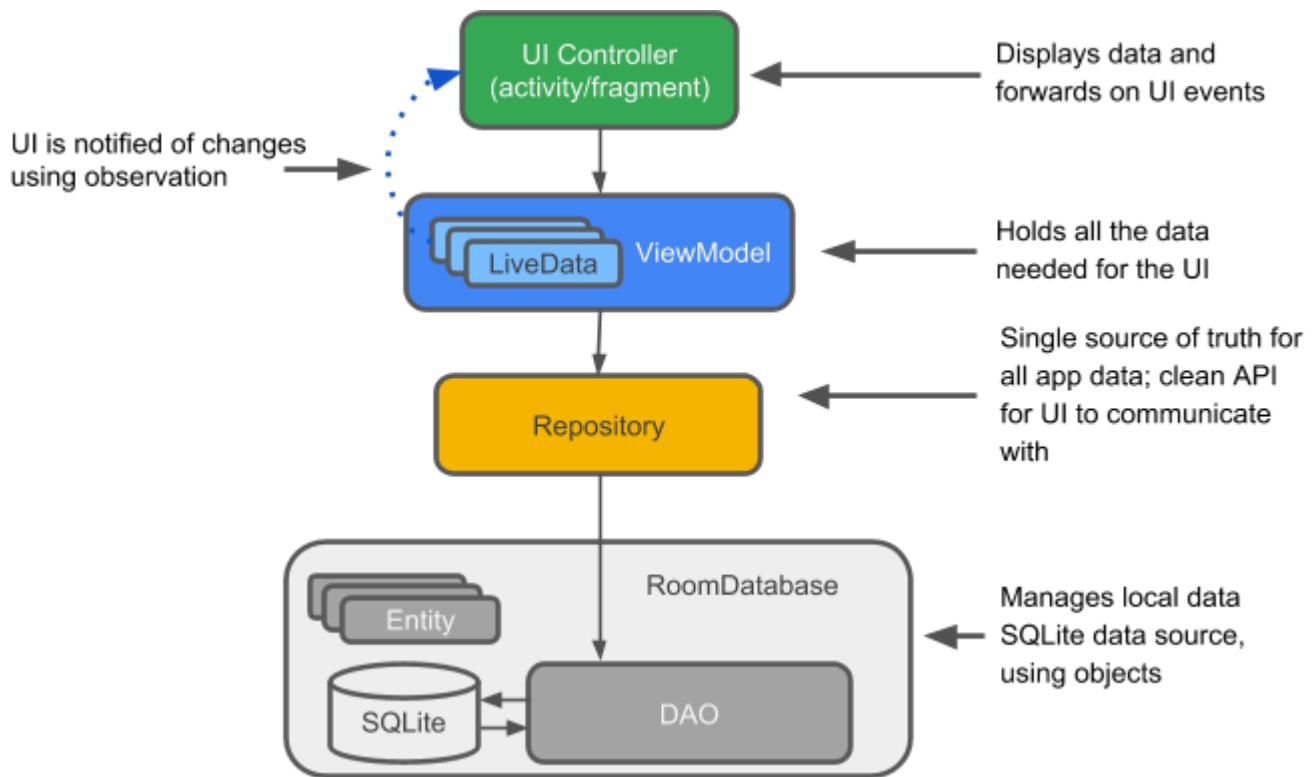
1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: SQLite Database with Room

App yang akan dibuat pada tutorial ini adalah sebuah App sederhana yang dapat menampilkan isi dari sebuah table pada sebuah SQLite database lokal yang ditampilkan pada RecyclerView. Selanjutnya pengguna dapat melihat, mengubah, menghapus data tersebut.

1. Buat Android Studio Project yang baru dengan nama: **Week09_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Basic Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Project yang dibuat menggunakan Architecture Components yang dapat saling bekerja sama sehingga menghasilkan App yang robust, testable, dan maintainable. Berikut diagram dari bentuk dasar arsitektur yang akan digunakan:



Untuk menambahkan komponen library Arsitektur di atas dibutuhkan untuk menambahkan secara manual ke dalam file Gradle. Tambahkan komponen **Library Room** dan **Lifecycle** ke dalam project dengan menambahkan kode berikut pada file **build.gradle (Module: app)** yang ada pada folder **Gradle Scripts** di dalam dependencies block:

```
// Room components
implementation "android.arch.persistence.room:runtime:$rootProject.roomVersion"
annotationProcessor
"android.arch.persistence.room:compiler:$rootProject.roomVersion"
androidTestImplementation
"android.arch.persistence.room:testing:$rootProject.roomVersion"
// Lifecycle components
implementation
"android.arch.lifecycle:extensions:$rootProject.archLifecycleVersion"
annotationProcessor
"android.arch.lifecycle:compiler:$rootProject.archLifecycleVersion"
```

Kemudian tambahkan kode berikut pada file **build.gradle(Project:Week09_[NIM])** pada folder **Gradle Scripts** di akhir file:

```
ext {
    roomVersion = '1.1.1'
    archLifecycleVersion = '1.1.1'
}
```

Kemudian Klik “Sync Now” yang ada pada kanan atas layar. Yakinkan project anda berhasil mem-build project anda dengan settingan gradle tersebut.

3. Membuat Entity Mahasiswa yang berupa sebuah Java class menjadi perantara antara SQLite database dengan object pada User Interface, dengan cara klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**, beri nama “**Mahasiswa**”. Agar class Mahasiswa ini dapat dengan mudah dijadikan object yang ditambahkan ke Intent, maka class Mahasiswa ini harus mengimplementasi interface **Serializable**. Buat konstruktor dan method-method getter dan setter untuk mempermudah penggunaan class ini. Serta masukkan anotasi-anotasi untuk keperluan integrasi dan link dengan database SQLite seperti **@Entity**, **@PrimaryKey**, **@NonNull**, dan **@ColumnInfo**. Berikut kode lengkap (selain import) class **Mahasiswa**:

```
@Entity(tableName = "tblMahasiswa")
public class Mahasiswa implements Serializable {
    @PrimaryKey
    @NonNull
    @ColumnInfo(name = "nim")
    private String nim;
    @ColumnInfo(name = "nama")
    private String nama;
    @ColumnInfo(name = "email")
    private String email;
    @ColumnInfo(name = "nomorhp")
    private String nomorhp;

    public Mahasiswa(String nim, String nama, String email,
                     String nomorhp){
        this.nim = nim;
        this.nama = nama;
        this.email = email;
        this.nomorhp = nomorhp;
    }

    public void setNim(String nim)      {this.nim      = nim;  }
    public void setNama(String nama)   {this.nama     = nama; }
    public void setEmail(String email) {this.email    = email; }
    public void setNomorhp(String nohp){this.nomorhp = nohp; }

    public String getNim()      { return this.nim;  }
    public String getNama()     { return this.nama; }
    public String getEmail()   { return this.email; }
    public String getNomorhp() { return this.nomorhp; }
}
```

4. Membuat DAO (Data Access Object) **Interface** dengan nama **MahasiswaDao**, dengan cara klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**. Isikan Name: **MahasiswaDAO** dan Kind: **Interface** pada jendela “**Create New Class**” lalu klik “**OK**”. Deklarasikan method-method untuk menjalankan query pada SQLite database seperti **SELECT, INSERT, UPDATE, dan DELETE** dengan menggunakan anotasi yang sesuai. Berikut kode lengkap interface **MahasiswaDAO**:

```

@Dao
public interface MahasiswaDAO {
    @Query("SELECT * FROM tblMahasiswa")
    LiveData<List<Mahasiswa>> getAllMahasiswa();

    @Insert(onConflict = OnConflictStrategy.IGNORE)
    void insert(Mahasiswa mhs);

    @Delete
    void delete(Mahasiswa mhs);

    @Update
    void update(Mahasiswa mhs);

    @Query("DELETE FROM tblMahasiswa")
    void deleteAll();
}

```

Dengan menggunakan DAO dan Room Database, Android system akan membuatkan skrip SQLite secara otomatis untuk insert, delete, dan update. Sementara untuk kustom query dapat kita definisikan sendiri dengan **@Query(...)** yang berasosiasi dengan abstract method di bawah anotasi tersebut seperti **getAllMahasiswa()** dan **deleteAll()** yang berasosiasi dengan query di atasnya.

5. Selanjutnya membuat Room database. Room adalah lapisan (layer) di atas SQLite database. Room menggunakan DAO untuk menjalankan query ke database. RoomDatabase berupa sebuah abstract class yang meng-extends **RoomDatabase**. Untuk membuat Room database, klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**. Beri nama “**MahasiswaRoomDatabase**”, Superclass: “**android.room.RoomDatabase**”, Modifier: **Abstract**. Gunakan anotasi **@Database** dengan parameter **entities, version, dan exportSchema**. Tambahkan deklarasi method abstract **daoMahasiswa()**, serta object **MahasiswaRoomDatabase** dengan nama **INSTANCE** yang akan dibuat dengan menggunakan **builder**. Tambahkan callback method untuk

memanggil (meng-override) `onOpen()` method. Berikut kode lengkap dari abstract class **MahasiswaRoomDatabase**:

```

@Database(entities = {Mahasiswa.class}, version = 1,
          exportSchema = false)
public abstract class MahasiswaRoomDatabase extends RoomDatabase {
    public abstract MahasiswaDAO daoMahasiswa();

    private static MahasiswaRoomDatabase INSTANCE;

    static MahasiswaRoomDatabase getDatabase(final Context context) {
        if (INSTANCE == null) {
            synchronized (MahasiswaRoomDatabase.class) {
                if (INSTANCE == null) {
                    INSTANCE = Room.databaseBuilder(
                        context.getApplicationContext(),
                        MahasiswaRoomDatabase.class, "dbMahasiswa")
                        .addCallback(sRoomDatabaseCallback)
                        .build();
                }
            }
        }
        return INSTANCE;
    }

    private static RoomDatabase.Callback sRoomDatabaseCallback =
        new RoomDatabase.Callback() {
            @Override
            public void onOpen(@NonNull SupportSQLiteDatabase db) {
                super.onOpen(db);
            }
        };
}

```

6. Membuat Repository yang berupa sebuah class yang memiliki member DAO dan isi dari table pada database. Klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**. Beri nama class "**MahasiswaRepository**" yang berisi private member variable **MahasiswaDAO** dan daftar Mahasiswa dengan tipe data **LiveData<List<Mahasiswa>>**, konstruktor dengan input parameter **Application** class, serta implementasi semua abstract method yang ada pada **MahasiswaDAO**. Untuk mengimplementasikan method yang membutuhkan adanya perubahan pada database seperti insert, update, dan delete, sangat dianjurkan untuk dijalankan pada Thread terpisah dengan Main Thread. Pada tutorial ini menggunakan **Asynchronous Task**. Berikut ini adalah kode lengkap dari class **MahasiswaRepository**:

```

public class MahasiswaRepository {
    private MahasiswaDAO daoMahasiswa;
    private LiveData<List<Mahasiswa>> daftarMahasiswa;

    MahasiswaRepository(Application app) {
        MahasiswaRoomDatabase db =
            MahasiswaRoomDatabase.getDatabase(app);
        daoMahasiswa = db.daoMahasiswa();
        daftarMahasiswa = daoMahasiswa.getAllMahasiswa();
    }

    LiveData<List<Mahasiswa>> getDaftarMahasiswa() {
        return this.daftarMahasiswa;
    }

    public void insert(Mahasiswa mhs) {
        new insertAsyncTask(daoMahasiswa).execute(mhs);
    }

    public void deleteAll() {
        new deleteAllAsyncTask(daoMahasiswa).execute();
    }

    public void delete(Mahasiswa mhs) {
        new deleteAsyncTask(daoMahasiswa).execute(mhs);
    }

    public void update(Mahasiswa mhs) {
        new updateAsyncTask(daoMahasiswa).execute(mhs);
    }

    private static class insertAsyncTask extends
        AsyncTask<Mahasiswa, Void, Void> {
        private MahasiswaDAO asyncDaoMahasiswa;
        insertAsyncTask(MahasiswaDAO dao) {
            this.asyncDaoMahasiswa = dao;
        }
        @Override
        protected Void doInBackground(final Mahasiswa... mahasiswa) {
            asyncDaoMahasiswa.insert(mahasiswa[0]);
            return null;
        }
    }

    private static class deleteAllAsyncTask extends
        AsyncTask<Void, Void, Void> {
        private MahasiswaDAO asyncDaoMahasiswa;
        deleteAllAsyncTask(MahasiswaDAO dao) {
            this.asyncDaoMahasiswa = dao;
        }
    }
}

```

```

@Override
protected Void doInBackground(final Void... voids) {
    asyncDaoMahasiswa.deleteAll();
    return null;
}

private static class deleteAsyncTask extends
        AsyncTask<Mahasiswa, Void, Void>{
    private MahasiswaDAO asyncDaoMahasiswa;
    deleteAsyncTask(MahasiswaDAO dao){
        this.asyncDaoMahasiswa = dao;
    }

    @Override
    protected Void doInBackground(final Mahasiswa... mahasiswas){
        asyncDaoMahasiswa.delete(mahasiswas[0]);
        return null;
    }
}

private static class updateAsyncTask extends
        AsyncTask<Mahasiswa, Void, Void>{
    private MahasiswaDAO asyncDaoMahasiswa;
    updateAsyncTask(MahasiswaDAO dao){
        this.asyncDaoMahasiswa = dao;
    }

    @Override
    protected Void doInBackground(final Mahasiswa... mahasiswas){
        asyncDaoMahasiswa.update(mahasiswas[0]);
        return null;
    }
}
}

```

7. Berikutnya adalah membuat ViewModel dengan cara klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**, beri nama class **MahasiswaViewModel** dengan superclass: **android.lifecycle.AndroidViewModel**. Class ini membutuhkan konstruktor wajib sehingga otomatis akan dinyatakan error oleh Android Studio. Tekan ALT+ENTER untuk memperbaikinya dan pilih implementasi constructor tersebut. Class ini berisi dua buah private member variable **mhsRepository** dengan tipe data **MahasiswaRepository** dan **daftarMahasiswa** dengan tipe data **LiveData<List<Mahasiswa>>**. Inisialisasi member ini dilakukan pada konstruktor wajib tadi dan wrap (deklarasi ulang) method-method yang ada pada repository. Di bawah ini adalah kode lengkap dari class **MahasiswaViewModel**:

```

public class MahasiswaViewModel extends AndroidViewModel {

    private MahasiswaRepository mhsRepository;

    private LiveData<List<Mahasiswa>> daftarMahasiswa;

    public MahasiswaViewModel(@NonNull Application application) {
        super(application);
        mhsRepository = new MahasiswaRepository(application);
        daftarMahasiswa = mhsRepository.getDaftarMahasiswa();
    }

    LiveData<List<Mahasiswa>> getDaftarMahasiswa(){
        return this.daftarMahasiswa;
    }

    public void insert(Mahasiswa mhs) {
        mhsRepository.insert(mhs);
    }

    public void deleteAll() {
        mhsRepository.deleteAll();
    }

    public void delete(Mahasiswa mhs) {
        mhsRepository.delete(mhs);
    }

    public void update(Mahasiswa mhs) {
        mhsRepository.update(mhs);
    }
}

```

8. Langkah berikutnya adalah membuat GUI (Graphical User Interface) agar pengguna dapat berinteraksi dengan database menggunakan back-end yang telah disiapkan. Pada tutorial ini akan menggunakan dua buah Activity MainActivity dan DetilActivity. DetilActivity ini yang akan melakukan penambahan dan peng-editan detil data mahasiswa yang akan dikirim dari MainActivity melalui RecyclerView object.
9. Buka dan edit layout XML file **activity_main.xml** untuk mengubah icon email dengan icon "+" pada Floating Action Button. Ganti:

```
app:srcCompat="@android:drawable/ic_dialog_email"
```

dengan:

```
app:srcCompat="@android:drawable/ic_input_add" />
```

10. Buka dan edit layout XML file `content_main.xml` untuk mengganti `TextView` bawaan dengan `RecyclerView` object. Gunakan full constraint (top, bottom, start, dan end) ke parent dan beri id dengan `rvMahasiswa`, berikut skrip XML untuk object `RecyclerView`:

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvMahasiswa"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

11. Membuat DetilActivity, klik **File > New > Activity > Empty Activity**. Beri nama **DetilActivity** dan layout **activity_detil**. Buka dan edit file layout `activity_detil.xml` untuk dapat mengisi dan mengubah data mahasiswa berupa NIM, Nama, Alamat Email, dan Nomor HP. Untuk kesederhanaan gunakan Vertikal LinearLayout sebagai root Layout. Serta tambahkan tombol “**Simpan**” dan “**Batal**”. Berikut Script lengkap layout file `activity_detil.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".DetilActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nim:" />
    <EditText
        android:id="@+id/etNim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nama Lengkap:" />
```

```

<EditText
    android:id="@+id/etNama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Alamat Email:" />
<EditText
    android:id="@+id/etEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textEmailAddress" />
<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Nomor Hp:" />
<EditText
    android:id="@+id/etNomorHp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="phone" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="horizontal">
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:onClick="simpanData"
        android:text="Simpan" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:onClick="batal"
        android:text="Batal" />
</LinearLayout>
</LinearLayout>

```

12. Buka dan lengkapi **DetilActivity.java** agar error message pada layout hilang dengan menambahkan dua method yang disebutkan pada onClick attribute yaitu **simpanData** dan **batal** serta

menghubungkan objek-objek pada layout dengan activity, berikut kode awal dari **DetilActivity.java**:

```

public class DetilActivity extends AppCompatActivity {
    private EditText etNim, etNama, etEmail, etNomorHP;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detil);
        etNim      = findViewById(R.id.etNim);
        etNama     = findViewById(R.id.etNama);
        etEmail    = findViewById(R.id.etEmail);
        etNomorHP = findViewById(R.id.etNomorHp);
    }
    public void simpanData(View view) {}

    public void batal(View view) {}
}

```

13. Selanjutnya adalah menyiapkan layout file untuk item yang akan ditampilkan pada RecyclerView object dengan cara klik kanan **app > res > layout** pilih **New > Layout resource file**, beri nama **mhs_item_layout**. Tambahkan dua buah **TextView** yang akan menampilkan Nim dan Nama. Berikut skrip XML dan layout ini:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tvItemNim"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:text="Nim"
        android:textSize="20dp" />

    <TextView
        android:id="@+id/tvItemNama"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:text="Nama" />

</LinearLayout>

```

14. Selanjutnya membuat class **Adapter** untuk mengendalikan **RecyclerView** object seperti pada tutorial-tutorial praktikum sebelumnya dengan membuat class baru yang meng-extends **RecyclerView.Adapter<ViewHolder>**. Klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**, beri nama **MahasiswaListAdapter**. Lengkapi class

ini dengan tambahkan extends **RecyclerView.Adapter<MahasiswaListAdapter.MahasiswaViewHolder>** lalu klik ALT+ENTER untuk memperbaiki error implementasi abstract method wajib untuk adapter dan buat inner class dengan nama **MahasiswaViewHolder** yang meng-extends class **RecyclerView.ViewHolder**, serta klik ALT+ENTER untuk memperbaiki error penambahan constructor wajib. Lengkapi inner class dengan membuat dua member **TextView** agar terhubung dengan **TextView** yang ada pada **mhs_item_layout.xml**. Lengkapi seluruh method-method yang ada pada class outer. **MahasiswaListAdapter** seperti tutorial dan tambahkan method **getMahasiswaAtPosition()** yang mengembalikan object **Mahasiswa** pada posisi tertentu di **RecyclerView** dan method setter **setDaftarMahasiswa()**. Berikut kode java lengkap dari class **MahasiswaListAdapter**:

```

public class MahasiswaListAdapter extends
  RecyclerView.Adapter<MahasiswaListAdapter.MahasiswaViewHolder> {
    private final LayoutInflater mInflater;
    private List<Mahasiswa> daftarMahasiswa;
    MahasiswaListAdapter(Context context) {
        mInflater = LayoutInflater.from(context);
    }

    @NonNull
    @Override
    public MahasiswaViewHolder onCreateViewHolder(@NonNull
        ViewGroup parent, int viewType) {
        View itemView = mInflater.inflate(R.layout.mhs_item_layout,
            parent,false);
        return new MahasiswaViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(@NonNull MahasiswaViewHolder holder,
        int position) {
        if(daftarMahasiswa != null){
            Mahasiswa mhs = daftarMahasiswa.get(position);
            holder.tvNim.setText(mhs.getNim());
            holder.tvNama.setText(mhs.getNama());
        } else {
            holder.tvNim.setText("Belum ada Mahasiswa");
        }
    }
    @Override
    public int getItemCount() {
        if(daftarMahasiswa != null){
            return daftarMahasiswa.size();
        } else {
            return 0;
        }
    }
}

```

```

public Mahasiswa getMahasiswaAtPosition(int posisi) {
    return daftarMahasiswa.get(posisi);
}

void setDaftarMahasiswa(List<Mahasiswa> mhss) {
    daftarMahasiswa = mhss;
    notifyDataSetChanged();
}

class MahasiswaViewHolder extends RecyclerView.ViewHolder{
    private final TextView tvNim;
    private final TextView tvNama;

    public MahasiswaViewHolder(@NonNull View itemView) {
        super(itemView);
        tvNim = itemView.findViewById(R.id.tvItemNim);
        tvNama = itemView.findViewById(R.id.tvItemNama);
    }
}
}

```

15. Selanjutnya mengubah Teks menu pada action bar dari “**Setting**” menjadi “**Hapus Semua**” dengan membuka dan mengedit file **strings.xml** pada folder **app > res > values** seperti skrip berikut:

```

<resources>
    <string name="app_name">Week09_067806</string>
    <string name="action_settings">Hapus Semua</string>
</resources>

```

16. Langkah berikutnya adalah membuka dan mengedit file **MainActivity.java** untuk:

- Menghubungkan dengan object **RecyclerView** dan konfigurasi object tersebut dengan menggunakan **MahasiswaListAdapter** serta mengisi dan mengasosiasikan adapter ini dengan object **MahasiswaViewModel**.
- Melengkapi kode di dalam **FloatingActionButton** untuk menangani klik user pada FAB tersebut yang memanggil **DetilActivity** untuk menambahkan data mahasiswa.
- Melengkapi kode penanganan klik user pada menu action bar **Hapus Semua** dengan memnggil method **deleteAll()**.
- Menambahkan interaksi user untuk dapat menghapus dan mengedit data mahasiswa. Pada tutorial ini untuk menghapus dengan menggeser (swipe) item di recyclerView ke kiri dan mengedit dengan menggeser (swipe) item ke kanan dengan memanfaatkan class **ItemTouchHelper**.

Berikut kode java lengkap dari class **MainActivity.java**:

```

public class MainActivity extends AppCompatActivity {
    private RecyclerView rv;
    private MahasiswaViewModel mhsVM;
    private static final int REQUEST_TAMBAH = 1;
    private static final int REQUEST_EDIT = 2;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent addIntent = new Intent(MainActivity.this,
                DetilActivity.class);
            startActivityForResult(addIntent, REQUEST_TAMBAH);
        }
    });

    rv = (RecyclerView) findViewById(R.id.rvMahasiswa);
    final MahasiswaListAdapter adapter = new MahasiswaListAdapter(this);
    rv.setAdapter(adapter);

    rv.setLayoutManager(new LinearLayoutManager(this));
    mhsVM = ViewModelProviders.of(this).get(MahasiswaViewModel.class);
    mhsVM.getDaftarMahasiswa().observe(this,
        new Observer<List<Mahasiswa>>() {

            @Override
            public void onChanged(List<Mahasiswa> mhss) {
                adapter.setDaftarMahasiswa(mhss);
            }
        });
}

ItemTouchHelper helper = new ItemTouchHelper(
    new ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT |
        ItemTouchHelper.RIGHT) {

        @Override
        public boolean onMove(@NonNull RecyclerView recyclerView,
            @NonNull RecyclerView.ViewHolder viewHolder,
            @NonNull RecyclerView.ViewHolder target) {
            return false;
        }

        @Override
        public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder,
            int direction) {
            int posisi = viewHolder.getAdapterPosition();

```

```

    Mahasiswa mhs = adapter.getMahasiswaAtPosition(posisi);
    if(direction == ItemTouchHelper.LEFT) {
        Toast.makeText(MainActivity.this,
            "Mahasiswa dengan NIM = "+mhs.getNim()+
            " dihapus",Toast.LENGTH_LONG).show();
        mhsVM.delete(mhs);
    }
    else if (direction == ItemTouchHelper.RIGHT) {
        Intent editIntet = new Intent (MainActivity.this,
            DetilActivity.class);
        editIntet.putExtra("MAHASISWA", mhs);
        startActivityForResult(editIntet,REQUEST_EDIT);
    }
}
);

helper.attachToRecyclerView(rv);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data){
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == RESULT_OK) {
        Mahasiswa mhs = (Mahasiswa)
            data.getSerializableExtra("MAHASISWA");
        if(requestCode == REQUEST_TAMBAH ) {
            mhsVM.insert(mhs);
        }
        else if (requestCode == REQUEST_EDIT) {
            mhsVM.update(mhs);
        }
    }
    rv.getAdapter().notifyDataSetChanged();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        mhsVM.deleteAll();
        return true;
    }
}

```

```

        return super.onOptionsItemSelected(item);
    }

}

```

17. Langkah selanjutnya yaitu melengkapi class **DetilActivity.java** untuk menangani permintaan penambahan dan pengeditan data. Untuk pengeditan data tambahkan penerimaan data mahasiswa dari **MainActivity.java** yang mengirim melalui **Intent**. Lengkapi method **simpanData()** yang akan mengirimkan data mahasiswa ke **MainActivity.java** dengan kode hasil **RESULT_OK** serta lengkapi method **batal()** dengan hanya mengirim kode hasil **RESULT_CANCELED**. Berikut kode lengkap dari **DetilActivity.java**:

```

public class DetilActivity extends AppCompatActivity {
    private EditText etNim, etNama, etEmail, etNomorHP;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detil);
        etNim      = findViewById(R.id.etNim);
        etNama     = findViewById(R.id.etNama);
        etEmail    = findViewById(R.id.etEmail);
        etNomorHP = findViewById(R.id.etNomorHp);
        Intent intent = getIntent();
        if(intent.hasExtra("MAHASISWA")) {
            Mahasiswa mhs = (Mahasiswa)
                intent.getSerializableExtra("MAHASISWA");
            etNim.setText(mhs.getNim());
            etNama.setText(mhs.getNama());
            etEmail.setText(mhs.getEmail());
            etNomorHP.setText(mhs.getNomorhp());
            etNim.setEnabled(false);
        } else {
            etNim.setEnabled(true);
        }
    }
    public void simpanData(View view) {
        String mNIM  = etNim.getText().toString();
        String mName = etNama.getText().toString();
        String mEmail = etEmail.getText().toString();
        String mNoHp = etNomorHP.getText().toString();
        if(mNIM.length() <= 0 || mName.length() <= 0 ||
           mEmail.length() <= 0 || mNoHp.length() <= 0) {
            Toast.makeText(this,"Semua harus Diisi",
                         Toast.LENGTH_LONG).show();
        } else {
            Intent intentJawab = new Intent();
            Mahasiswa mhs = new Mahasiswa(mNIM, mName, mEmail, mNoHp);
            intentJawab.putExtra("MAHASISWA",mhs);
            setResult(RESULT_OK,intentJawab);
            finish();
        }
    }
}

```

```
        }
    }
public void batal(View view){
    Intent intentJawab = new Intent();
    setResult(RESULT_CANCELED,intentJawab);
    finish();
}
}
```

18. Jalankan (run) project anda serta coba anda lakukan beberapa percobaan untuk menambah data mahasiswa, menambahkan data mahasiswa dengan NIM yang sudah ada, menghapus satu data mahasiswa dengan swipe kiri, mengedit data mahasiswa dengan swipe kanan dan menghapus semua dengan klik titik tidak pada action bar di atas da klik “Hapus Semua”. Perhatikan dan catat bagaimana behavior dari app anda.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning.
2. Lakukan pengujian terhadap app anda sesuai petunjuk tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 10

(BACKGROUND PROCESSES)

DESKRIPSI TEMA

- Asynchronous Task
 - AsyncTask
 - AsyncTaskLoader
- Services
 - Intent Services
 - Services
 - Bound Services

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat memanfaatkan background processes or threads such as: AsyncTask, AsyncTask Loader, Intent Services, Services, and Bound Services.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: AsyncTask

Salah satu kunci dalam pengembangan aplikasi Android adalah tidak diperbolehkannya melakukan operasi / aksi yang akan memakan waktu yang banyak pada thread utama pada aplikasi.

Aturan Kedua, kode yang menggunakan thread yang terpisah tidak diperbolehkan untuk mengubah aspek UI yang ada. Setiap perubahan terhadap aspek UI harus dilakukan pada thread utama. Hal ini dikarenakan Android UI Toolkit memiliki konsep yang non-thread-safe.

Apabila ada aksi yang memakan waktu lama yang perlu dijalankan di balik layar dan juga perlu untuk mengubah aspek UI, Android menyediakan class yang bernama AsyncTask.

AsyncTask memerlukan 3 tipe generik yang diperlukan yaitu:

- **Params** → parameter yang dikirimkan pada saat aksi dieksekusi.
- **Progress** → tipe dari unit progress pada saat komputasi di balik layar.
- **Result** → hasil yang digunakan setelah komputasi tersebut selesai.

Contoh:

```
private class TestingAsyncTask extends AsyncTask<String, int, Object> {
}
```

String akan mewakili Params.

Int akan mewakili Progress.

Object akan mewakili Result.

Selain itu, pada class `AsyncTask<Params, Progress, Result>` memiliki beberapa method yang dapat diimplementasi, yaitu:

- `doInBackground()` → Method ini akan memiliki kode / logic yang akan dieksekusi pada background. Pada method ini kita juga bisa mengirimkan hasil ke UI Thread dengan menggunakan method `publishProgress()`.
- `onPreExecute()` → Method ini akan memiliki kode / logic yang akan dijalankan sebelum method `doInBackground()` dijalankan.
- `onPostExecute()` → Method ini akan memiliki kode / logic yang akan dijalankan setelah `doInBackground()` selesai.
- `onProgressUpdate()` → Method ini menerima update progress dari `doInBackground()`, yang dipublikasikan dari method `publishProgress()`. Method ini dapat digunakan untuk mengupdate UI Thread.

Pada tutorial ini kita akan menggunakan `AsyncTask`:

1. Buat Android Studio Project yang baru dengan nama: **Week1oa_[NIM]**, domain: **umn.ac.id**, minimum android: **API 25: Android 7.1.1 (Nougat)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Buka layout file **activity_main.xml** dan ganti text "Halo" Pada `TextView` dengan teks "Siap untuk mulai", tambahkan sebuah `Button` di bawah `TextView` dengan teks "Mulai Task" yang memiliki attribut `android:onClick = "startTask"`, serta sebuah `ProgressBar` horizontal yang diletakkan berada di bawah tombol. Berikut skrip lengkap dari `activity_main.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp"
    tools:context=".MainActivity">
```

```

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Siap untuk mulai"
    android:textSize="24sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:onClick="startTask"
    android:text="Mulai Task"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView1" />
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

3. Buka file MainActivity.java untuk menambahkan object TextView, ProgressBar, dan mengimplementasikan method startTask. Tambahkan pula kode untuk menyimpan SavedInstanceState seperti tutorial sebelumnya. Setelah ini buatlah sebuah Inner Class dengan modifier private yang baru pada MainActivity, dengan nama AsyncTaskSederhana yang merupakan turunan dari class AsyncTask yang menerima 3 parameter berupa Integer, Integer, dan String. Pada class AsyncTaskSederhana tersebut, implement method onPreExecute(), doInBackground(), onPostExecute(), onProgressUpdate(). Pada onPreExecute(), tampilan ProgressBar dan set nilai progress ke 0 (nol). Pada doInBackground(), melakukan pengulangan sebanyak n kali, dimana n adalah nilai input yang berupa angka acak antara 10 sampai 60, dengan masing-masing iterasi akan berhenti sementara (sleep) selama 200 milidetik dan mengirimkan nilai progress berupa nilai presentasi dari iterasi yang sudah dijalankan dengan memanggil method publishProgress(). Method doInBackground() menghasilkan sebuah String yang berupa kalimat pemberitahuan berapa lama background processes ini berjalan. Pada method onProgressUpdate(), menerima nilai progress dari method doInBackground dan menampilkannya pada TextView dan ProgressBar. Pada method onPostExecute(), menampilkan hasil teks dan menyembunyikan ProgressBar. Sementara pemanggilan AsyncTaskSederhana dilakukan pada method startTask() yang aktif ketika pengguna meng-klik tombol "Mulai Task" dengan menggunakan method execute() disertai input parameter

sebuah nilai integer acak antara 10 sampai dengan 60. Berikut kode java lengkap pada MainActivity.java:

```

public class MainActivity extends AppCompatActivity {
    private static final String TEXT_STATE = "currentText";
    private TextView mTextView;
    private ProgressBar mProgressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mTextView = findViewById(R.id.textView1);
        mProgressBar = findViewById(R.id.progressBar);
        mProgressBar.setMax(100);
        if(savedInstanceState!=null) {
            mTextView.setText(savedInstanceState.getString(TEXT_STATE));
        }
    }

    public void startTask(View view) {
        mTextView.setText("Siap untuk mulai");
        new AsyncTaskSederhana().execute((int)(Math.random()*50)+10);
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putString(TEXT_STATE, mTextView.getText().toString());
    }

    private class AsyncTaskSederhana extends AsyncTask<Integer,
                                                Integer, String> {
        @Override
        protected void onPreExecute() {
            mProgressBar.setVisibility(View.VISIBLE);
            mProgressBar.setProgress(0);
        }

        @Override
        protected String doInBackground(Integer... integers) {
            int n = integers[0].intValue();
            int s = 0;
            try {
                s = n * 200;
                for(int i = 0; i < n; i++) {
                    Thread.sleep(200);
                    publishProgress(((int)((100 * i)/(float) n)));
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            return "Aktif lagi setelah tidur selama "+ s + " milidetik";
        }
    }
}

```

```
    @Override
    protected void onProgressUpdate(Integer ... progress) {
        mTextView.setText("progress = "+progress[0]+" person");
        mProgressBar.setProgress(progress[0]);
    }

    @Override
    protected void onPostExecute(String result) {
        mTextView.setText(result);
        mProgressBar.setVisibility(View.INVISIBLE);
    }
}
```

4. Yakinkan Project anda tidak ada kesalahan lalu jalankan project anda serta catat dan perhatikan apa yang terjadi App anda.

Tutorial: AsyncTaskLoader

App yang akan dibuat pada tutorial ini adalah sebuah App yang memiliki behavior sama dengan tutorial sebelumnya (`AsyncTask`) namun diimplementasikan menggunakan `AsyncTaskLoader` dan `LoaderManager`.

1. Buat Android Studio Project yang baru dengan nama: **Week1ob_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 25: Android 7.1.1 (Nougat)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
 2. Layout yang digunakan sama dengan layout pada tutorial AsyncTask di atas.
 3. Buka dan edit MainActivity.java untuk menambahkan object TextView dan ProgressBar serta pembuatan inner class AsyncTaskLoader beserta implementasi LoaderManager. Berikut Langkah detilnya:
 - a. Buat static inner class dengan nama ContohLoader yang merupakan turunan dari class AsyncTaskLoader dengan parameter Void.
 - b. Tambahkan member variable mCounter dan static variable mActivity dengan tipe data WeakReference<MainActivity>. Note: penggunaan WeakReference untuk menghindari leaking context dari Activity. Weak references mencegah kebocoran memory dengan memperbolehkan object dipegang oleh (held by) variable untuk di garbage-collected.
 - c. Buatlah konstruktor yang menerima parameter masukan Context, Int, dan MainActivity yang memanggil super konstruktor dan menginisialisai ke dua variable membernya.
 - d. Implementasikan method loadInBackground yang melakukan proses counting dan menunggu selama 200 milidetik seperti tutorial sebelumnya dengan perbedaan update nilai dan progress

yang ditampilkan pada TextView, dan ProgressBar harus melalui mActivity dan dijalankan dengan menggunakan method runOnUiThread().

- e. Modifikasi class MainActivity dengan menambahkan implements LoaderManager.LoaderCallbacks<Void> dan mengimplementasikan method-method abstract pada interface tersebut yaitu onCreateLoader(...), onLoadFinished(...), dan onLoadReset(). Pada method onCreateLoader() isi dengan membuat object dari class ContohLoader dan panggil method forceLoad() lalu kembalikan object tersebut. pada onLoadFinished, hanya memanggil method destroyLoader() dari class LoaderManager (menggunakan method getSupportFragmentManager()). Untuk method onLoadReset() dikosongkan.

Berikut kode lengkap dari MainActivity.java:

```

public class MainActivity extends AppCompatActivity implements
    LoaderManager.LoaderCallbacks<Void> {

    private TextView mTextView;
    private ProgressBar mProgressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mTextView = findViewById(R.id.textView1);
        mProgressBar = findViewById(R.id.progressBar);
        mProgressBar.setMax(100);
    }

    public void startTask(View view){
        if(!getSupportFragmentManager().hasRunningLoaders())
            getSupportFragmentManager().initLoader(0, (Bundle) null,
                this);
    }

    @NonNull
    @Override
    public Loader<Void> onCreateLoader(int id, @Nullable Bundle args) {
        AsyncTaskLoader<Void> asyncTaskLoader =
            new ContohLoader(this, (int)(Math.random()*50)+10, this);
        asyncTaskLoader.forceLoad();
        return asyncTaskLoader;
    }

    @Override
    public void onLoadFinished(@NonNull Loader<Void> loader, Void data)
    {
        getSupportFragmentManager().destroyLoader(0);
    }
}

```

```

@Override
public void onLoaderReset(@NonNull Loader<Void> loader) {
}

static class ContohLoader extends AsyncTaskLoader<Void> {
    static WeakReference<MainActivity> mActivity;
    int mCounter = 0;

    public ContohLoader(@NonNull Context context, int n,
                        MainActivity main) {
        super(context);
        mCounter = n;
        mActivity = new WeakReference<MainActivity>(main);
    }

    @Nullable
    @Override
    public Void loadInBackground() {
        try {
            for(int i = 0; i <= mCounter; i++) {
                Thread.sleep(200);
                final int progress = ((int)((100 * i) /
                    (float) mCounter));
                if (mActivity.get() != null) {
                    mActivity.get().runOnUiThread(new
                        Runnable() {
                            @Override
                            public void run() {
                                if(progress < 100) {
                                    mActivity.get().mProgressBar
                                        .setProgress(progress);
                                    mActivity.get().mTextView
                                        .setText("progress = " +
                                            progress + " persen");
                                } else {
                                    mActivity.get().mProgressBar
                                        .setProgress(100);
                                    mActivity.get().mTextView.
                                        setText("Selesai selama "
                                            + mCounter * 200 +
                                            " milidetik..."));
                                }
                            }
                        });
                }
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        return null;
    }
}

```

4. Yakinkan Project anda tidak ada kesalahan lalu jalankan project anda serta catat dan perhatikan apa yang terjadi pada App anda.

Services

App yang akan dibuat pada tutorial ini adalah sebuah App yang memiliki behavior sama dengan tutorial sebelumnya (`AsyncTask` dan `AsyncTaskLoader`) namun diimplementasikan menggunakan Services seperti Intent Service, Unbound Service, dan Bound Services.

Service merupakan komponen pada Android yang bekerja di balik layar (background) tanpa memerlukan adanya interaksi dengan user. Karena sebuah Service tidak memiliki user interface, Service tidak terikat pada Lifecycle Activity dalam Android.

Apa perbedaan antara Service dengan `AsyncTask`?

Pada Android, `AsyncTask` didesain hanya untuk melakukan pekerjaan sekali jalan yang membutuhkan waktu sangat lama yang tidak dapat dijalankan pada UI Thread, sedangkan Service didesain untuk bekerja terus menerus di balik layar.

Apabila kita hanya membutuhkan pengambilan data dari internet apabila sebuah button ditekan atau activity berjalan sekali saja, kemudian selesai, maka sebaiknya menggunakan `AsyncTask`, namun apabila digunakan untuk pemutaran music (e.g. Google Play Music / Spotify), Pengambilan daya baru secara terus menerus di balik layar, dsb, maka sebaiknya menggunakan Service.

Service juga dapat digunakan / dipanggil sekalipun Activity nya sedang tidak berjalan.

Service sendiri beberapa tipe, yang akan kita pelajari pada minggu ini adalah started service (Unbound Service) dan Bound Service.

Bagian-bagian (Method) dari Service:

- `onStartCommand()`

Method ini dipanggil Ketika service dijalankan oleh komponen Android via method `startService()`.

Method ini tidak perlu digunakan apabila kita membuat bound services.

- `onBind()`

Method ini dipanggil ketika komponen mengikat ke service via method `bindService()`. Ketika method ini digunakan untuk bound services, method ini harus me-return object berupa `IBinder` untuk memfasilitasi komunikasi dengan klien. Pada started services, method ini diimplementasikan untuk me-return nilai `NLL`.

- `onCreate()`

Method ini digunakan untuk melakukan task awal. Method ini dipanggil sebelum method `onStartCommand()` dipanggil atau `onBind()`.

- `onDestroy()`

Dipanggil ketika Services akan di-destroy.

- `onHandleIntent()`

Hanya digunakan pada subclass IntentService. Method ini digunakan untuk menangani pemrosesan pada Services. Dieksekusi pada thread yang berbeda dari main thread.

Method `onStartCommand()` memiliki beberapa konstanta return type integer, diantaranya adalah:

- `START_STICKY`
- `START_NOT_STICKY`
- `START_REDELIVER_INTENT`

Tutorial: Started Services (Unbound Service)

Started service merupakan Service yang dijalankan oleh komponen aplikasi Android seperti Activity atau `BroadcastReceiver()` dan berjalan indefinite di balik layar sampai servicesnya dihentikan, atau di-destroy oleh Android runtime untuk membersihkan resources pada saat penuh.

Untuk membuat Service ini ada 2 class yang dapat digunakan untuk membuat Started Service:

- Service

Merupakan class dasar untuk semua Service. Ketika akan membuat Service (extend Service), diwajibkan untuk membuat sebuah thread baru dimana Service dapat menyelesaikan seluruh task yang dibuat. By default Service akan menggunakan main thread dari aplikasi / komponen yang memanggil Service, yang apabila tidak ditangani dengan baik, dapat menghambat performa dari aplikasi yang dibuat.

- Intent Service

Pada dasarnya Service dijalankan secara default pada main thread yang sama dengan komponen Android (e.g. Activity) dimana mereka dipanggil, sehingga setiap CPU Intensive Task yang perlu dilakukan oleh service harus dilakukan pada sebuah thread yang baru, untuk menghindari hit pada performa.

`IntentService` merupakan sebuah class yang merupakan turunan dari class `Service` yang akan membuat sebuah worker thread yang digunakan untuk menangani task di balik layar dan menangani setiap request secara asynchronous. Yang SANGAT dibutuhkan pada `IntentService` adalah implementasi dari method `onHandleIntent()` untuk menjalankan setiap request yang ada.

Pada tutorial ini kita akan mencoba untuk membuat Started Service sederhana pada Android menggunakan `IntentService` dan `Service`.

Started Service – IntentService

1. Buat Android Studio Project yang baru dengan nama: **Week1oc_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 25: Android 7.1.1 (Nougat)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Membuat sebuah class dengan nama SimpleIntentService yang merupakan turunan dari IntentService. Lalu tambahkan kode ini pada method onHandleIntent untuk mengetahui bahwa Service yang kita buat sudah dimulai atau belum dan dimasukkan pada Logcat dan melakukan process yang sama seperti pada tutorial sebelumnya.

```

public class SimpleIntentService extends IntentService {

    public SimpleIntentService() {
        super("SimpleIntentService");
    }

    @Override
    protected void onHandleIntent(@Nullable Intent intent) {
        Log.i("INTENTSERVICE", "onHandleIntent: IntentService dimulai !");
        int n =(int) (Math.random()*50)+10;
        try {
            for(int i = 0; i < n; i++) {
                Thread.sleep(200);
                Log.i("INTENTSERVICE", "onHandleIntent: berjalan "+
                    ((int)((100 * i)/(float) n)) + " persen");
            }
            Log.i("INTENTSERVICE", "onHandleIntent: Selesai");
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

3. Tambahkan permission untuk menggunakan internet pada AndroidManifest.xml, menambahkan Service ke dalam AndroidManifest.xml. Service merupakan bagian dari Aplikasi Android yang terdapat di luar Activity, hal ini juga tercemin pada AndroidManifest.xml yang akan kita modifikasi menjadi seperti berikut:

```
<service android:name=".SimpleIntentService" />
```

di dalam block “Application”.

4. Setelah service sudah teregister pada AndroidManifest.xml, sekarang kita akan memanggil service tersebut pada MainActivity.java. Cara pemanggilannya mirip seperti kita memanggil Activity di dalam method onCreate(...):

```
Intent servIntent = new Intent(this, SimpleIntentService.class);
startService(servIntent);
```

- Jalankan Aplikasi Android yang sudah Anda buat dan cobalah lihat pada Logcat aplikasi Anda.

Started Service – Service

- Masih menggunakan Project yang sama, buatlah sebuah Service [**File > New > Service > service**] dengan nama CustomService. Override method onCreate pada CustomService dan tambahkan kode sebagai berikut. Kode ini hanya digunakan untuk memonitor pembuatan service pada logcat.

```
@Override
public void onCreate() {
    super.onCreate();
    Log.i("CUSTOMSERVICE", "onCreate: CustomService");
}
```

- Override method onStartCommand pada CustomService dan tambahkan kode sebagai berikut. Kode ini berfungsi untuk menampilkan tulisan bahwa Service sedang berjalan pada logcat dan melakukan proses seperti tutorial sebelumnya.

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Log.i("CUSTOMSERVICE", "onStartCommand: "+startId);
    int n =(int) (Math.random()*50)+10;
    try {
        for(int i = 0; i < n; i++) {
            Thread.sleep(200);
            Log.i("CUSTOMSERVICE", "onStartCommand: " + startId +
                  " berjalan "+ ((int)((100 * i)/(float) n)) + " persen");
        }
        Log.i("CUSTOMSERVICE", "onStartCommand: " + startId +
              " Selesai");
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    return Service.START_STICKY;
}
```

- Modifikasi method onBind pada CustomService. Kode ini berfungsi untuk menampilkan tulisan pada Service sudah di-bind pada Aplikasi dan Override method onDestroy pada CustomService dan tambahkan kode sebagai berikut. Kode ini berfungsi untuk menampilkan tulisan bahwa Service sudah berhasil dihapus pada logcat.

```
@Override
public IBinder onBind(Intent intent) {
    Log.i("CUSTOM SERVICE", "onBind: Service Bind");
    return null;
}
@Override
public void onDestroy() {
    super.onDestroy();
    Log.i("CUSTOMSERVICE", "onDestroy: Service Destroyed");
}
```

4. Modifikasi AndroidManifest.xml untuk menambahkan Service yang sudah kita buat, apabila sudah dibuat secara otomatis pada saat membuat Service, abaikan langkah ini.

```
<service
    android:name=".CustomService"
    android:enabled="true"
    android:exported="true" />
```

5. Pada layout activity_main.xml, tambahkan 1 buah button dengan id "main_button_startservice" dengan text "Mulai Service", serta ada MainActivity.java, modifikasi kode menjadi seperti berikut.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Intent servIntent = new Intent(this, SimpleIntentService.class);
    startService(servIntent);

    Button btnStartService = findViewById(R.id.main_button_startservice);
    btnStartService.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this,
                CustomService.class);
            startService(intent);
        }
    });
}
```

6. Jalankan kode tersebut dan tekan tombol Start sebanyak 3x, setelah 20 detik dan tunggu hingga selesai. Apakah ada error yang terjadi? Error ini terjadi karena pada method onStartCommand, seluruh task yang diketik, akan dijalankan pada main thread, dan solusi nya adalah dengan menggunakan AsyncTask untuk menyelesaiakannya.
7. Buatlah class baru bernama CustomServiceTask yang merupakan extends dari AsyncTask dengan Params Integer, Progress Integer, dan Results Integer.

Keterangan:

Method doInBackground pada CustomServiceTask merupakan modifikasi dari onStartCommand pada CustomService.

```
public class CustomServiceTask extends AsyncTask<Integer, Integer,
Integer> {
    @Override
    protected void onProgressUpdate(Integer ... integers) {
        Log.i("CUSTOMSERVICE", "onStartCommand: " + integers[0] +
        " berjalan " + integers[1] + " persen");
    }
}
```

```

@Override
protected void onPostExecute(Integer result) {
    Log.i("CUSTOMSERVICE", "onStartCommand: " + result + " Selesai");
}

@Override
protected Integer doInBackground(Integer... integers) {
    int startId = integers[0];
    int n = (int) (Math.random() * 50) + 10;
    try {
        for(int i = 0; i < n; i++) {
            Thread.sleep(200);
            publishProgress(startId, ((int)((100 * i) / (float) n)));
        }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    return startId;
}
}

```

8. Modifikasi CustomService method onStartCommand menjadi seperti berikut.

```

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Log.i("CUSTOMSERVICE", "onStartCommand: "+startId);
    //      int n = (int) (Math.random() * 50) + 10;
    //      try {
    //          for(int i = 0; i < n; i++) {
    //              Thread.sleep(200);
    //              Log.i("CUSTOMSERVICE", "onStartCommand: " + startId +
    //                  " berjalan " + ((int)((100 * i) / (float) n)) + "
    persen");
    //      }
    //      Log.i("CUSTOMSERVICE", "onStartCommand: " + startId +
    //          " Selesai");
    //  } catch (InterruptedException e) {
    //      e.printStackTrace();
    //  }

    AsyncTask customServiceTask = new CustomServiceTask()
        .executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, startId);
    return Service.START_STICKY;
}

```

9. Jalankan kembali Aplikasi yang sudah Anda buat, kemudian tekan tombol Startnya 3x, apakah masih error?

Tutorial: Started Service (Unbound Service)

Bound service hamper sama seperti Started service, hanya saja pada Started service pada umumnya tidak memiliki return result (NULL) atau tidak memperbolehkan adanya interaksi dengan komponen yang menjalankan Service tersebut. Bound Service memperbolehkan komponen yang menjalankan Service untuk berinteraksi dan menerima result dari / ke Service. Melalui implementasi dari InterProcess Communication (IPC).

Pada tutorial ini kita akan mencoba untuk membuat Bound Service sederhana yang berguna untuk mengambil waktu dari service kemudian mengembalikannya ke aplikasi berdasarkan Project sebelumnya.

1. Masih pada project yang sama. Buatlah sebuah Service [**File > New > Service > Service**] dengan nama CustomBoundService. Pada Bound Service, kita akan memuat “pengikat” (Binder) antara Service dengan komponen Android melalui class Binder, dan selanjutnya kita akan membuat method `getCurrentTime()` yang berguna untuk mengembalikan Jam dan Tanggal pada saat ini, kemudian mengikatnya object customBinder pada method `onBind()`. Berikut kode lengkap dari class CustomBoundService:

```
public class CustomBoundService extends Service {
    public IBinder customBinder = new CustomLocalBinder();

    public CustomBoundService() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        return customBinder;
    }

    public class CustomLocalBinder extends Binder {
        CustomBoundService getService(){
            return CustomBoundService.this;
        }
    }

    public String getCurrentTime(){
        SimpleDateFormat dateFormat = new SimpleDateFormat(
            "HH:mm:ss MM/dd/yyyy", Locale.US);
        return (dateFormat.format(new Date()));
    }
}
```

2. Setelah itu kita akan me-register Bound Service yang sudah dibuat pada `AndroidManifest.xml` jika belum dilakukan secara otomatis:

```
<service
    android:name=".CustomBoundService"
    android:enabled="true"
    android:exported="true"></service>
```

3. Selanjutnya kita akan mendefinisikan Bound Service yang sudah dibuat pada Aplikasi Android. Untuk itu selanjutnya kita akan membuka MainActivity.java kemudian menambahkan sebuah Object ServiceConnection pada MainActivity. ServiceConnection berguna untuk menyambungkan / mengikat Bound Service dengan Activity / komponen Android yang ada. Modifikasilah MainActivity sebagai berikut:

```
public class MainActivity extends AppCompatActivity {
    CustomBoundService customBoundService;
    boolean isBound = false;

    private ServiceConnection serviceConnection =
    new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName name,
                                       IBinder service) {
            CustomBoundService.CustomLocalBinder binder =
            (CustomBoundService.CustomLocalBinder) service;
            customBoundService = binder.getService();
            isBound = true;
        }

        @Override
        public void onServiceDisconnected(ComponentName name) {
            isBound = false;
        }
    };
}
```

4. Selanjutnya, kita akan mengikat Service yang sudah didefinisikan pada method onCreate. Modifikasilah method onCreate pada MainActivity menjadi seperti berikut:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Intent servIntent = new Intent(this, SimpleIntentService.class);
    startService(servIntent);

    Button btnStartService = findViewById(R.id.main_button_startservice);
    btnStartService.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this,
                                         CustomService.class);
            startService(intent);
        }
    });

    Intent intent2 = new Intent(this, CustomBoundService.class);
    bindService(intent2, serviceConnection, Context.BIND_AUTO_CREATE);
}
```

5. Selanjutnya kita akan membuat sebuah button untuk menampilkan Toast yang memberikan informasi berupa jam yang diambil dari Bound Service yang sudah dibuat. Bukalah activity_main.xml dan Tambahkan sebuah button dengan id “main_button_showtime” dan text “Tampilkan Waktu”.
6. Pada MainActivity.java tambahkan kode berikut pada onCreate untuk menampilkan jam pada Toast berdasarkan data service yang ada.

```
Button btnShowTime = findViewById(R.id.main_button_showtime);
btnShowTime.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String currentTime = customBoundService.getCurrentTime();
        Toast.makeText(getApplicationContext(),
                currentTime,Toast.LENGTH_LONG).show();
    }
});
```

7. Jalankan dan cobalah kode yang sudah Anda ketik tersebut.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning.
2. Lakukan pengujian terhadap app anda sesuai petunjuk tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 11

(WEB SERVICE)

DESKRIPSI TEMA

- Network and Internet
 - Memeriksa koneksi jaringan dan Internet
- Web Service
 - Akses ke web page dengan protocol Http dan/atauHttps
 - Kirim data menggunakan GET dan POST
 - Parsing JSON data

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat memeriksa status jaringan dan koneksi internet serta dapat memanggil dan mengakses halaman web dinas (seperti .php), mengirim parameter serta mengelola hasil dan jawaban dari halaman web tersebut dengan menggunakan format JSON (Javascript Object Notation).

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Web Service

App yang akan dibuat pada tutorial ini adalah sebuah App yang dapat memeriksa apakah device terhubung ke jaringan (atau Internet) dengan memeriksa konektivitas ke WiFi dan/atau ke mobile data (tidak sampai mendeteksi terkoneksi dengan internet atau tidak), serta dapat melakukan akses ke web service (dalam tutorial ini menggunakan pemrograman server PHP yang dapat menerima parameter dengan metode GET maupun POST dan mengirimkan hasilnya berupa JSON kode) dengan dan tanpa mengirim parameter. App juga akan melakukan parsing JSON yang didapatkan dari web service tersebut untuk kemudian dikelola lebih lanjut. Behavior dari App ini sangat mirip sekali dengan Tutorial minggu ke 09 yaitu tentang SQLite yang menampilkan isi dari sebuah table pada sebuah database (MySQL database di server) yang ditampilkan pada RecyclerView. Selanjutnya pengguna dapat melihat, mengubah, menghapus data tersebut menggunakan mobile device. Perubahan data yang terjadi akan bersifat global karena data tersebut tersedia di cloud yang dapat diakses oleh banyak device yang menggunakan App yang sama. Berikut Langkah-langkahnya:

1. Buat Android Studio Project yang baru dengan nama: **Week11_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Basic Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Project yang dibuat menggunakan fitur jaringan (network dan internet) serta koneksi menggunakan protocol HTTP dan/atau HTTPS. Untuk akses peralatan jaringan dan koneksi ke Internet dibutuhkan **user-permissions** dan untuk melakukan request dengan HTTP (HTTPURLConnection) dibutuhkan setting yang membolehkan pengiriman clear text (tidak menggunakan enkripsi). Berikut kode XML pada AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="id.ac.umn.week11_067806">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name=
                    "android.permission.ACCESS_NETWORK_STATE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:usesCleartextTraffic="true">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name=
                          "android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

3. Seperti pada tutorial SQLite, kita harus menyediakan class Mahasiswa yang menjadi perantara antara MySQL database (di server / cloud) dengan object pada User Interface, dengan cara klik kanan pada folder **app > java > id.ac.umn.week11_[NIM]** pilih **New > Java Class**, beri nama "**Mahasiswa**". Agar class Mahasiswa ini dapat dengan mudah dijadikan object yang ditambahkan ke Intent, maka class Mahasiswa ini harus mengimplementasi interface **Serializable**. Buat konstruktor dan method-

method getter dan setter untuk mempermudah penggunaan class ini. Berikut kode lengkap (selain import) class **Mahasiswa**:

```

public MahasiswaViewModel(@NonNull Application application) {
    super(application);
    mhsRepository = new MahasiswaRepository(application);
    daftarMahasiswa = mhsRepository.getDaftarMahasiswa();
}

LiveData<List<Mahasiswa>> getDaftarMahasiswa() {
    return this.daftarMahasiswa;
}

public void insert(Mahasiswa mhs) {
    mhsRepository.insert(mhs);
}

public void deleteAll() {
    mhsRepository.deleteAll();
}

public void delete(Mahasiswa mhs) {
    mhsRepository.delete(mhs);
}

public void update(Mahasiswa mhs) {
    mhsRepository.update(mhs);
}
}

```

4. Langkah berikutnya adalah membuat GUI (Graphical User Interface) agar pengguna dapat berinteraksi dengan web service yang telah disiapkan. Pada tutorial ini akan menggunakan empat buah Activity: MainActivity, DetilActivity (menambah dan meng-edit data mahasiswa), CariActivity (mencari daftar mahasiswa sesuai input nim dan/atau nama mahasiswa), dan SettingActivity (melakukan setting URL dari folder tempat action web page service disimpan).
5. Buka dan edit layout XML file **activity_main.xml** untuk mengubah icon email dengan icon "+" pada Floating Action Button. Ganti:

```
app:srcCompat="@android:drawable/ic_dialog_email"
```

dengan:

```
app:srcCompat="@android:drawable/ic_input_add" />
```

6. Buka dan edit layout XML file `content_main.xml` untuk mengganti `TextView` bawaan dengan `RecyclerView` object. Gunakan full constraint (top, bottom, start, dan end) parent dan beri id dengan `rvMahasiswa`, berikut skrip XML untuk object `RecyclerView`:

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvMahasiswa"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

7. Membuat DetilActivity, klik File > New > Activity > Empty Activity. Beri nama `DetilActivity` dan layout `activity_detil`. Buka dan edit file layout `activity_detil.xml` untuk dapat mengisi dan mengubah data mahasiswa berupa NIM, Nama, Alamat Email, dan Nomor HP. Untuk kesederhanaan gunakan Vertikal LinearLayout sebagai root Layout. Serta tambahkan tombol "Simpan" dan "Batal". Berikut Script lengkap layout file `activity_detil.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".DetilActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nim:" />
    <EditText
        android:id="@+id/etNim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nama Lengkap:" />
```

```

<EditText
    android:id="@+id/etNama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Alamat Email:" />
<EditText
    android:id="@+id/etEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textEmailAddress" />
<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Nomor Hp:" />
<EditText
    android:id="@+id/etNomorHp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="phone" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="horizontal">
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:onClick="simpanData"
        android:text="Simpan" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:onClick="batal"
        android:text="Batal" />
</LinearLayout>
</LinearLayout>

```

Buka dan lengkapi **DetilActivity.java** agar error message pada layout hilang dengan menambahkan dua method yang disebutkan pada onClick attribute yaitu **simpanData** dan **batal** serta menghubungkan objek-objek pada layout dengan activity, berikut kode awal dari **DetilActivity.java**:

```

public class DetilActivity extends AppCompatActivity {
    private EditText etNim, etNama, etEmail, etNomorHP;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detil);
        etNim      = findViewById(R.id.etNim);
        etNama     = findViewById(R.id.etNama);
        etEmail    = findViewById(R.id.etEmail);
        etNomorHP = findViewById(R.id.etNomorHp);
    }
    public void simpanData(View view){}

    public void batal(View view){}
}

```

8. Selanjutnya adalah menyiapkan layout file untuk item yang akan ditampilkan pada RecyclerView object dengan cara klik kanan **app > res > layout** pilih **New > Layout resource file**, beri nama **mhs_item_layout**. Tambahkan dua buah **TextView** yang akan menampilkan Nim dan Nama. Berikut skrip XML dari layout ini:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tvItemNim"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:text="Nim"
        android:textSize="20dp" />

    <TextView
        android:id="@+id/tvItemNama"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:text="Nama" />

</LinearLayout>

```

9. Selanjutnya membuat class **Adapter** untuk mengendalikan **RecyclerView** object seperti pada tutorial-tutorial praktikum sebelumnya dengan membuat class baru yang meng-extends **RecyclerView.Adapter<ViewHolder>**. Klik kanan pada folder **app > java > id.ac.umn.week09_[NIM]** pilih **New > Java Class**, beri nama **MahasiswaListAdapter**. Lengkapi class ini dengan tambahkan **extends RecyclerView.Adapter<MahasiswaListAdapter.MahasiswaViewHolder>** lalu klik **ALT+ENTER** untuk memperbaiki error implementasi abstract method wajib untuk adapter dan buat inner class dengan nama **MahasiswaViewHolder** yang meng-extends class **RecyclerView.ViewHolder**, serta klik **ALT+ENTER** untuk memperbaiki error penambahan constructor wajib. Lengkapi inner class

dengan membuat dua member `TextView` agar terhubung dengan `TextView` yang ada pada `mhs_item_layout.xml`. Lengkapi seluruh method-method yang ada pada class outer `MahasiswaListAdapter` seperti tutorial sebelumnya dan tambahkan method `getMahasiswaAtPosition()` yang mengembalikan object `Mahasiswa` pada posisi tertentu di `RecyclerView` dan methode setter `setDaftarMahasiswa()`. Serta tambahkan method-method yang terkait dengan perubahan isi dari data mahasiswa yang di web server (di cloud) dengan daftar terkait dengan perubahan isi dari data mahasiswa yang digunakan pada adapter ini seperti: `hapusData(int posisi)` yang akan menghapus item pada daftarMahasiswa pada posisi tertentu, `tambahData(Mahasiswa mhs)` yang mengubah data pada posisi tertentu di daftarMahasiswa sesuai dengan data baru yang diterima. Berikut kode java lengkap dari class `MahasiswaListAdapter`:

```
public class MahasiswaListAdapter extends
    RecyclerView.Adapter<MahasiswaListAdapter.MahasiswaViewHolder> {
    private final LayoutInflater mInflater;
    private List<Mahasiswa> daftarMahasiswa;
    MahasiswaListAdapter(Context context) {
        mInflater = LayoutInflater.from(context);
    }

    @NonNull
    @Override
    public MahasiswaViewHolder onCreateViewHolder(@NonNull
                                                ViewGroup parent, int viewType) {
        View itemView = mInflater.inflate(R.layout.mhs_item_layout,
                                         parent, false);
        return new MahasiswaViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(@NonNull MahasiswaViewHolder holder,
                               int position) {
        if(daftarMahasiswa != null){
            Mahasiswa mhs = daftarMahasiswa.get(position);
            holder.tvNim.setText(mhs.getNim());
            holder.tvNama.setText(mhs.getNama());
        } else {
            holder.tvNim.setText("Belum ada Mahasiswa");
        }
    }
}
```

```

@Override
public int getItemCount() {
    if(daftarMahasiswa != null){
        return daftarMahasiswa.size();
    } else {
        return 0;
    }
}

public Mahasiswa getMahasiswaAtPosition(int posisi){
    return daftarMahasiswa.get(posisi);
}

void setDaftarMahasiswa(List<Mahasiswa> mhss) {
    daftarMahasiswa = mhss;
    notifyDataSetChanged();
}

void hapusData(int posisi){
    daftarMahasiswa.remove(posisi);
    notifyDataSetChanged();
}

void tambahData(Mahasiswa mhs) {
    daftarMahasiswa.add(mhs);
    notifyDataSetChanged();
}

void updateData(int posisi, Mahasiswa mhs) {
    daftarMahasiswa.set(posisi,mhs);
    notifyDataSetChanged();
    notifyItemChanged(posisi);
}

class MahasiswaViewHolder extends RecyclerView.ViewHolder{
    private final TextView tvNim;
    private final TextView tvNama;

    public MahasiswaViewHolder(@NonNull View itemView) {
        super(itemView);
        tvNim = itemView.findViewById(R.id.tvItemNim);
        tvNama = itemView.findViewById(R.id.tvItemNama);
    }
}
}

```

10. Selanjutnya menambah menu “Cari Mahasiswa” pada action bar dengan membuka dan mengedit **menu_main.xml** pada folder **app > res > menu**, dengan menambahkan XML skrip berikut setelah item Setting:

```
<item
    android:id="@+id/action_cari"
    android:orderInCategory="100"
    android:title="Cari Mahasiswa"
    app:showAsAction="never"
/>>
```

11. Membuat **SettingActivity** dengan cara **File > New > Activity > Empty Activity**, beri nama dengan **SettingActivity** dan Layout file dengan nama **activity_setting**. Buka dan edit layout file tersebut dengan menggunakan LinearLayout vertikal yang berisi Judul, label, isian untuk URL dari folder tempat menyimpan web service API, serta sebuah Button yang akan menyimpan data tersebut ke dalam **SharedPreference**. Berikut script dari layout **activity_setting.xml**:

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".SettingActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pengaturan"
        android:textSize="30sp" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Http Server (URL)" />
    <EditText
        android:id="@+id/httpServer"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:ems="10"
        android:inputType="textPersonName" />
    <Button
        android:id="@+id/btnSimpanSetting"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SIMPAN" />
</LinearLayout>
```

12. Selanjutnya lengkapi **SettingActivity.java** yang membuat object untuk link dengan object isian (EditText) dan tombol pada layout file, serta mengimplementasi onClick method pada saat tombol “Simpan” diclick dengan mengisikan variable statis **MainActivity.server** dan membalas panggilan dari **MainActivity** (yang memanggil dengan **startActivityForResult()**) dengan mengirimkan **RESULT_OK** pada method **setResult()**. Berikut kode dari **SettingActivity.java**:

```

public class SettingActivity extends AppCompatActivity {
    private EditText etServer;
    private Button btnSimpan;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_setting);
        etServer = (EditText) findViewById(R.id.httpServer);
        etServer.setText(MainActivity.server);
        btnSimpan = (Button) findViewById(R.id.btnSimpanSetting);
        btnSimpan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                MainActivity.server = etServer.getText().toString();
                setResult(RESULT_OK);
                finish();
            }
        });
    }
}

```

13. Membuat activity baru **CariActivity** yang akan menangani kegiatan pencarian data mahasiswa berdasarkan nim dan/atau nama. Klik **File > New > Activity > Empty Activity**, beri nama CariActivity dan **activity_cari** untuk layout filenya. Layout ini berisi dua isian untuk Nim dan Nama, sebuah Button untuk mengaktifkan pencarian data mahasiswa, dan sebuah multi-line EditText (text area) yang menampung hasil dari pencarian data mahasiswa pada server. Berikut skrip XML dari layout **activity_cari.xml**:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".CariActivity">

```

```
<EditText
    android:id="@+id/etNimCari"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:hint="Nim"
    android:inputType="number"
    app:layout_constraintEnd_toStartOf="@+id	btnCari"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<EditText
    android:id="@+id/etNamaCari"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:ems="10"
    android:hint="Nama"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toStartOf="@+id	btnCari"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etNimCari" />
<Button
    android:id="@+id	btnCari"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginTop="10dp"
    android:text="CARI"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<ScrollView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:scrollbarAlwaysDrawHorizontalTrack="false"
    android:scrollbarAlwaysDrawVerticalTrack="false"
    android:scrollbarStyle="insideOverlay"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etNamaCari">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <EditText
            android:id="@+id/etHasilCari"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="start|top"
            android:inputType="textMultiLine" />
    </LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

14. Selanjutnya lengkapi **CariActivity.java** yang dapat menerima input nim dan/atau nama dari data mahasiswa yang dicari dengan menggunakan web service API “**cari_mahasiswa.php**” menggunakan method “**GET**”. Perlu diperhatikan inputan dari user apakah user mengisi kedua-duanya atau hanya satu atau tidak diisi dengan menggunakan “**if**” yang tepat, karena ini digunakan untuk membangun teks parameter yang akan dikirim ke server. Gunakan worker thread seperti **AsyncTask** untuk mengakses dan me-request ke web server. Berikut kode java pada **CariActivity.java**:

```

public class CariActivity extends AppCompatActivity {
    EditText etNimCari, etNamaCari, etHasilCari;
    Button btnCari;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cari);
        etNimCari = (EditText) findViewById(R.id.etNimCari);
        etNamaCari = (EditText) findViewById(R.id.etNamaCari);
        etHasilCari = (EditText) findViewById(R.id.etHasilCari);
        etHasilCari.setEnabled(false);
        btnCari = (Button) findViewById(R.id.btnCari);
        btnCari.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                etHasilCari.setText("");
                String nim = etNimCari.getText().toString().trim();
                String nama = etNamaCari.getText().toString().trim();
                if(!nim.isEmpty() || !nama.isEmpty()) {
                    String actinWebPage = MainActivity.server+
                        "cari_mahasiswa.php?";
                    Uri builtURI = null;
                    if(!nim.isEmpty() && !nama.isEmpty()){
                        builtURI = Uri.parse(actinWebPage).buildUpon()
                            .appendQueryParameter("nim",nim)
                            .appendQueryParameter("nama",nama)
                            .build();
                    } else if (!nim.isEmpty() && nama.isEmpty()){
                        builtURI = Uri.parse(actinWebPage).buildUpon()
                            .appendQueryParameter("nim",nim)
                            .build();
                    } else if (nim.isEmpty() && !nama.isEmpty()){
                        builtURI = Uri.parse(actinWebPage).buildUpon()
                            .appendQueryParameter("nama",nama)
                            .build();
                    }
                    new AsynCariMahasiswa().execute(builtURI.toString());
                }else{
                    etHasilCari.setText(
                        "NIM dan/atau NAMA harus ada yang diisi");
                }
            }
        });
    }
}

```

```
class AsynCariMahasiswa extends AsyncTask<String, Void, String>{
    @Override
    protected String doInBackground(String... strings) {
        HttpURLConnection con = null;
        BufferedReader reader = null;
        String mhsJSONString = null;
        try {
            URL urlRequest = new URL(strings[0]);
            con = (HttpURLConnection) urlRequest.openConnection();
            con.setRequestMethod("GET");
            con.connect();
            InputStream inputStream = con.getInputStream();
            reader = new BufferedReader(new
                InputStreamReader(inputStream));
            StringBuilder builder = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
                builder.append(line);
                builder.append("\n");
            }
            if (builder.length() == 0) {
                return null;
            }
            mhsJSONString = builder.toString();
            JSONObject jsonObject = new JSONObject(mhsJSONString);
            String hasilText = "";
            int sukses = (int) jsonObject.getInt("success");
            if(sukses == 1){
                JSONArray mhsJSONArray = (JSONArray)
                    jsonObject.get("mahasiswa");
                for(int i = 0 ; i < mhsJSONArray.length(); i++){
                    JSONObject mhsJSONObj = (JSONObject)
                        mhsJSONArray.get(i);
                    String nim = mhsJSONObj.getString("nim");
                    String nama = mhsJSONObj.getString("nama");
                    String email = mhsJSONObj.getString("email");
                    String nHp = mhsJSONObj.getString("nomorHp");
                    hasilText += nim + " | " + nama + " | " + email +
                        " | " + nHp + "\n";
                }
                return hasilText;
            } else {
                String pesan = jsonObject.getString("message");
                return pesan;
            }
        } catch (Exception e){
            Log.e("CARI_ACTIVITY",
                "Error = "+e.getMessage().toString());
        }
        return null;
    }
}
```

```

        finally {
            if (con != null) {
                con.disconnect();
            }
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    Log.e("WEBSERVICE",
                            "Exception: "+e.getMessage().toString());
                }
            }
        }

    @Override
    protected void
    onPostExecute(String s) {
        super.onPostExecute(s);
        etHasilCari.setText(s);
    }
}
}

```

15. Berikutnya adalah membuka dan mengedit file **MainActivity.java** untuk:
- Menghubungkan dengan object **RecyclerView** dan konfigurasi object tersebut dengan menggunakan **MahasiswaListAdapter** serta mengisi dan mengasosiasikan adapter ini dengan **daftarMahasiswa** dengan tipe data **List<Mahasiswa>**. Untuk mengisi dataMahasiswa, menggunakan worker thread (**AsyncTask**) yang merupakan inner class dengan nama **AsyncAmbilSemuaMahasiswa** yang akan mengakses **ambil_Semua_mahasiswa.php** tanpa mengirim input parameter.
 - Melengkapi kode di dalam **FloatingActionButton** untuk menangani klik user pada FAB tersebut yang memanggil **DetilActivity** untuk menambahkan data mahasiswa. Pemanggilan activity ini menggunakan method **startActivityForResult()** yang mengharapkan balasan dari activity ini berupa object class Mahasiswa.
 - Melengkapi kode penanganan klik user pada menu action bar “Setting” dengan memanggil activity **SettingActivity** yang digunakan untuk mengedit folder web service API, dan “Cari Mahasiswa” dengan memanggil activity **CariActivity** yang digunakan untuk mencari data mahasiswa dengan kata kunci nim dan/atau nama. Pemanggilan SettingActivity menggunakan method **startActivityForResult()** yang mengharapkan balasan apakah user klik Simpan Button atau klik Back button bawaan berupa **resultCode**.

- d. Menambahkan interaksi user untuk dapat menghapus dan mengedit data mahasiswa. Pada tutorial ini untuk menghapus dengan menggeser (swipe) item ke kanan dengan memanfaatkan class **ItemTouchHelper**. Untuk mengedit item yang digeser kanan akan memanggil activity **DetailActivity** dengan mengirimkan data object Mahasiswa pada item yang digeser tersebut pada Intentnya. Sementara untuk menghapus data mahasiswa akan memanggil inner class **AsyncHapusMahasiswa**.

Berikut kode java lengkap dari class **MainActivity.java**:

```
public class MainActivity extends AppCompatActivity {
    private final String KEY = "server";
    private final int SETTING_REQUEST = 1;
    private SharedPreferences mPreferences;
    private String sharedPrefFile;
    private Context context;
    protected static String server;
    private RecyclerView rv;
    static MahasiswaListAdapter adapter;
    private static final int REQUEST_TAMBAH = 2;
    private static final int REQUEST_EDIT = 3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        context = this;
        sharedPrefFile = context.getPackageName();
        mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);
        server = mPreferences.getString(KEY,
                "https://if633.mobdevumn.com/api/");
        rv = (RecyclerView) findViewById(R.id.recyclerView);
        adapter = new MahasiswaListAdapter(this);
        rv.setAdapter(adapter);
        rv.setLayoutManager(new LinearLayoutManager(this));
    }
}
```

```
// Check the status of the network connection.
ConnectivityManager connMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = null;
if (connMgr != null) {
    networkInfo = connMgr.getActiveNetworkInfo();
}
if (networkInfo != null && networkInfo.isConnected()){
    new AsyncAmbilSemuaMahasiswa().execute();
} else {
    Toast.makeText(MainActivity.this,
        "Tidak ada Koneksi Jaringan", Toast.LENGTH_LONG).show();
}

Toolbar toolbar = findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent addIntent = new Intent(MainActivity.this,
            DetilActivity.class);
        startActivityForResult(addIntent, REQUEST_TAMBAH);
    }
});

ItemTouchHelper helper = new ItemTouchHelper(
    new ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT
        | ItemTouchHelper.RIGHT) {
        @Override
        public boolean onMove(@NonNull RecyclerView recyclerView,
            @NonNull RecyclerView.ViewHolder viewHolder,
            @NonNull RecyclerView.ViewHolder target) {
            return false;
        }
        @Override
        public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder,
            int direction) {
            int posisi = viewHolder.getAdapterPosition();
            Mahasiswa mhs =
                adapter.getMahasiswaAtPosition(posisi);
            if(direction == ItemTouchHelper.LEFT){
                new AsyncHapusMahasiswa(posisi).
                    execute(mhs.getNim());
            }
        }
    }
);
```

```
        else if(direction == ItemTouchHelper.RIGHT) {
            Intent editIntet = new Intent(MainActivity.this,
                DetilActivity.class);
            editIntet.putExtra("MAHASISWA", mhs);
            editIntet.putExtra("POSISI", posisi);
            startActivityForResult(editIntet, REQUEST_EDIT);
        }
    }
}
};

helper.attachToRecyclerView(rv);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        Intent settingIntent = new Intent(MainActivity.this,
            SettingActivity.class);
        startActivityForResult(settingIntent, SETTING_REQUEST);
        return true;
    } else if(id == R.id.action_cari){
        Intent cariIntent = new Intent(MainActivity.this,
            CariActivity.class);
        startActivity(cariIntent);
        return true;
    }
    return super.onOptionsItemSelected(item);
}

@Override
protected void onPause(){
    super.onPause();
    SharedPreferences.Editor preferencesEditor = mPreferences.edit();
    preferencesEditor.putString(KEY, server);
    preferencesEditor.apply();
}
```

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == RESULT_OK){
        if(requestCode == SETTING_REQUEST){
            Sharedpreferences.Editor preferencesEditor =
                mPreferences.edit();
            preferencesEditor.putString(KEY, server);
            preferencesEditor.apply();
            Toast.makeText(context,
                "Setting telah disimpan",Toast.LENGTH_LONG).show();
        } else {
            Mahasiswa mhs = (Mahasiswa)
                data.getSerializableExtra("MAHASISWA");
            if (requestCode == REQUEST_TAMBAH) {
                new AsyncTambahMahasiswa().execute(mhs);
            } else if (requestCode == REQUEST_EDIT) {
                int posisi = data.getIntExtra("POSISI",-1);
                new AsyncUpdateMahasiswa(posisi).execute(mhs);
            }
        }
    } else {
        adapter.notifyDataSetChanged();
    }
}

void tutupKoneksi(BufferedReader reader, HttpsURLConnection con){
    if (con != null) {
        con.disconnect();
    }
    if (reader != null) {
        try {
            reader.close();
        } catch (IOException e) {
            Log.e("WEBSERVICE",
                "Exception: "+e.getMessage().toString());
        }
    }
}
String sambungKeServer(String actionPage, String postString){
    HttpsURLConnection con = null;
    StringBuilder sb = null;
    BufferedReader reader = null;
    String urlParameters = postString;
    byte[] postData = urlParameters.getBytes(
        StandardCharsets.UTF_8 );
    int postDataLength = postData.length;
```

```
try {
    sb = new StringBuilder();
    String urlString = MainActivity.server+actionPage;
    URL url = new URL(urlString);
    con = (HttpsURLConnection) url.openConnection();
    con.setReadTimeout(10000);
    con.setConnectTimeout(15000);
    con.setRequestMethod("POST");
    con.setDoOutput(true);
    con.setDoInput(true);
    con.setInstanceFollowRedirects( false );
    con.setRequestProperty("Content-Type",
                           "application/x-www-form-urlencoded");
    con.setRequestProperty("charset", "utf-8");
    con.setRequestProperty("Content-Length",
                           Integer.toString( postDataLength ));
    con.setUseCaches( false );
    con.connect();
    DataOutputStream wr = new DataOutputStream(
                           con.getOutputStream());
    wr.write(postData);
    wr.flush();
    wr.close();
    InputStream is = con.getInputStream();
    reader = new BufferedReader(new InputStreamReader(is));
    String line;
    while ((line = reader.readLine()) != null) {
        sb.append(line);
        sb.append("\n");
    }
    if (sb.length() == 0) {
        return null;
    }
    String mhsJSONString = sb.toString();
    JSONObject jsonObject = new JSONObject(mhsJSONString);
    int sukses = (int) jsonObject.getInt("success");
    if(sukses == 1){
        return "sukses";
    } else {
        return null;
    }
} catch (Exception e){
    Log.e("WEBSERVICE","Exception: "+e.getMessage().toString());
} finally {
    tutupKoneksi(reader, con);
}
return null;
}
```

```

class AsyncAmbilSemuaMahasiswa extends AsyncTask<Void, Void,
                                         List<Mahasiswa>> {
    @Override
    protected List<Mahasiswa> doInBackground(Void... voids) {
        HttpsURLConnection con = null;
        StringBuilder sb = null;
        BufferedReader reader = null;
        List<Mahasiswa> mahasiswas = new ArrayList<>();
        try {
            sb = new StringBuilder();
            String urlString = MainActivity.server +
                               "ambil_semua_mahasiswa.php";
            URL url = new URL(urlString);
            con = (HttpsURLConnection) url.openConnection();
            con.connect();
            InputStream is = con.getInputStream();
            reader = new BufferedReader(new InputStreamReader(is));
            String line;
            while ((line = reader.readLine()) != null) {
                sb.append(line);
                sb.append("\n");
            }
            if (sb.length() == 0) {
                return null;
            }
            String mhsJSONString = sb.toString();
            JSONObject jsonObject = new JSONObject(mhsJSONString);
            int sukses = (int) jsonObject.getInt("success");
            if(sukses == 1){
                JSONArray mhsJSONArray = (JSONArray)
                    jsonObject.get("mahasiswas");
                for(int i = 0 ; i < mhsJSONArray.length(); i++){
                    JSONObject mhsJSONObj = (JSONObject)
                        mhsJSONArray.get(i);
                    String nim = mhsJSONObj.getString("nim");
                    String nama = mhsJSONObj.getString("nama");
                    String email = mhsJSONObj.getString("email");
                    String nomorHp = mhsJSONObj.getString("nomorHp");
                    Mahasiswa mhs = new Mahasiswa(nim, nama, email,
                                                   nomorHp);
                    mahasiswas.add(mhs);
                }
                return mahasiswas;
            } else {
                return null;
            }
        } catch (Exception e){
            Log.e("WEBSERVICE",
                  "Exception: "+e.getMessage().toString());
        } finally {
            tutupKoneksi(reader, con);
        }
        return null;
    }
}

```

```

@Override
protected void onPostExecute(List<Mahasiswa> mahasiswas) {
    super.onPostExecute(mahasiswas);
    adapter.setDaftarMahasiswa(mahasiswas);
}
}

class AsyncHapusMahasiswa extends AsyncTask<String, Void, String>{
    private int mPosisi;
    AsyncHapusMahasiswa(int posisi){
        mPosisi = posisi;
    }
    @Override
    protected String doInBackground(String... strings) {
        return sambungKeServer("hapus_mahasiswa.php",
                               "nim=" + strings[0]);
    }
    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        if( (s != null) && (s.equalsIgnoreCase("sukses")) ){
            adapter.hapusData(mPosisi);
            Toast.makeText(MainActivity.this,
                           "Mahasiswa Berhasil dihapus",Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(MainActivity.this,
                           "Mahasiswa gagal dihapus",Toast.LENGTH_LONG).show();
        }
    }
}

class AsyncTambahMahasiswa extends AsyncTask<Mahasiswa, Void, String>{
    Mahasiswa mhs;
    @Override
    protected String doInBackground(Mahasiswa... mahasiswas) {
        mhs = mahasiswas[0];
        String urlParameters = "nim="+mhs.getNim()+"&" +
                               "nama="+mhs.getNama()+"&" +
                               "email="+mhs.getEmail()+"&" +
                               "nomorHp="+mhs.getNomorhp();
        return sambungKeServer("tambah_mahasiswa.php", urlParameters);
    }
    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        if( (s != null) && (s.equalsIgnoreCase("sukses")) ){
            adapter.tambahData(mhs);
            Toast.makeText(MainActivity.this,
                           "Mahasiswa Berhasil ditambahkan",
                           Toast.LENGTH_LONG).show();
        }
    }
}

```

```

        else {
            Toast.makeText(MainActivity.this,
                "Mahasiswa gagal ditambahkan", Toast.LENGTH_LONG).show();
        }
    }

class AsyncUpdateMahasiswa extends AsyncTask<Mahasiswa, Void, String>{
    Mahasiswa mhs;
    int mPosisi;
    AsyncUpdateMahasiswa(int posisi){
        mPosisi = posisi;
    }
    @Override
    protected String doInBackground(Mahasiswa... mahasiswas) {
        mhs = mahasiswas[0];
        String urlParameters = "nim="+mhs.getNim()+"&" +
            "nama="+mhs.getNama()+"&" +
            "email="+mhs.getEmail()+"&" +
            "nomorHp="+mhs.getNomorhp();
        return sambungKeServer("update_mahasiswa.php",
            urlParameters);
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        if( (s != null) && (s.equalsIgnoreCase("sukses")) ){
            adapter.updateData(mPosisi, mhs);
            Toast.makeText(MainActivity.this,
                "Data Mahasiswa Berhasil diupdate",
                Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(MainActivity.this,
                "Data Mahasiswa gagal diupdate",
                Toast.LENGTH_LONG).show();
        }
    }
}

```

16. Yakinkan device atau emulator yang digunakan dapat mengakses Internet sehingga saat menjalankan App ini dapat berjalan dengan baik. Jalankan (run) project anda serta coba lakukan beberapa percobaan untuk menambah data mahasiswa, menambahkan data mahasiswa dengan NIM yang sudah ada, menghapus satu data mahasiswa dengan swipe kiri, mengedit data mahasiswa dengan swipe kanan. Lakukan juga percobaan dengan mengubah setting pada device atau emulator yang digunakan seperti menyalakan dan mematikan WiFi, serta menyalakan dan mematikan Mobile data. Perhatikan dan catat bagaimana behavior dari app anda.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning.
2. Lakukan pengujian terhadap app anda sesuai petunjuk tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 12

(SENSOR)

DESKRIPSI TEMA

- Sensor
 - Daftar Sensor yang ada pada device
 - Register Listener untuk data sensor
 - Menangani data yang datang dari sensor

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat mengetahui daftar sensor yang ada pada device, melakukan registrasi listener data sensor serta menangani data yang datang dari sensor.

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Deteksi Sensor

1. Buat Android Studio Project yang baru dengan nama: **Week12a_[NIM]**, domain:**umn.ac.id**, minimum Android: **API 25: Android 7.1.1 (Nougat)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
2. Buka layout file **activity_main.xml** ganti TextView bawaan dengan **ScrollView**, serta tambahkan sebuah **TextView** dengan nama **daftarSensor**, berikut skrip XML pada **activity_main.xml**:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/daftarSensor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="(daftar Sensor)" />
</LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

3. Buka file **MainActivity.java** untuk menambahkan object **TextView** yang ada pada layout file di atas dengan nama **tvDaftarSensor**, serta menambahkan member variable object **SensorManager** dengan nama **mSensorManager** serta dapatkan default sensor manager dengan memanggil method **getSystemService()** dengan parameter **Context.SENSOR_SERVICE**. Lalu gunakan method **getSensorList()** yang ada pada sensor manager dengan parameter **Sensor.TYPE_ALL** untuk mendapatkan daftar dari semua Sensor yang ada pada device atau emulator yang menjalankan App ini. Selanjutnya gunakan loop untuk melihat dan mengubah daftar sensor tersebut menjadi String dan tampilkan String tersebut ke **tvDaftarSensor**. Berikut kode Java dari **MainActivity.java**:

```

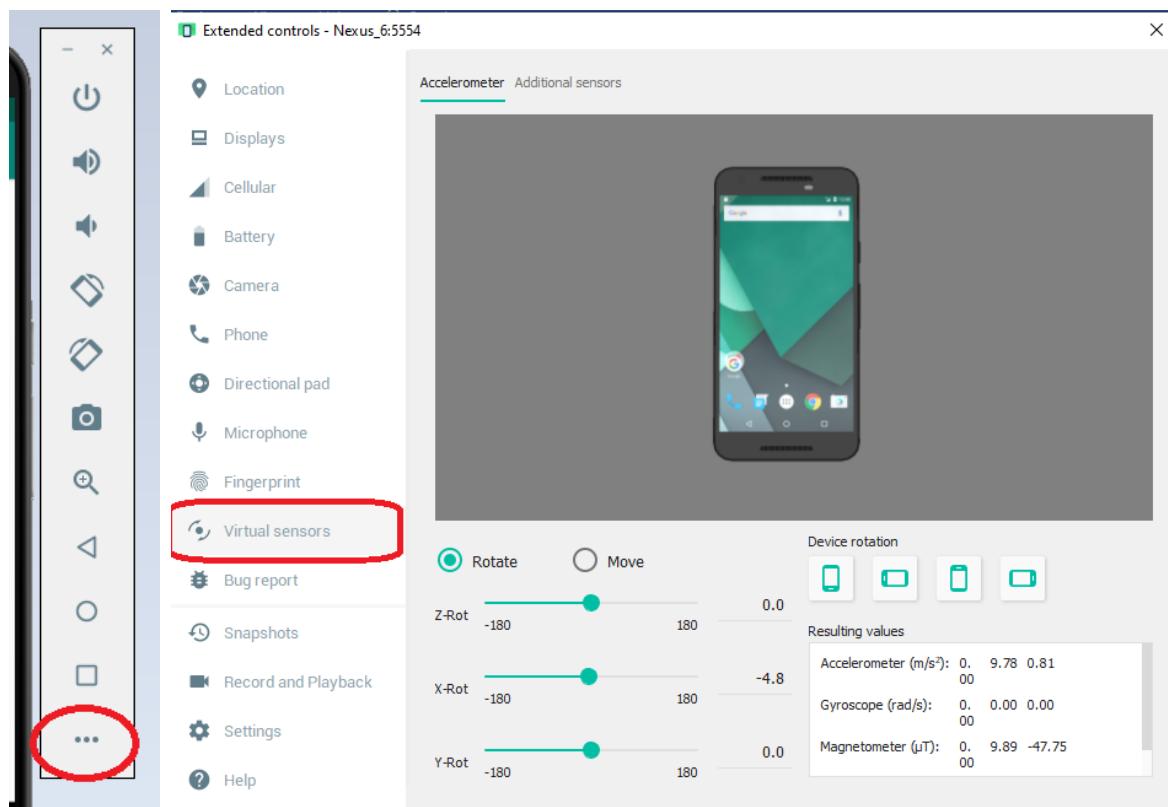
public class MainActivity extends AppCompatActivity {
    private TextView tvDaftarSensor;
    private SensorManager mSensorManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvDaftarSensor = findViewById(R.id.daftarSensor);
        mSensorManager = (SensorManager)
            getSystemService(Context.SENSOR_SERVICE);
        List<Sensor> daftarSensor =
            mSensorManager.getSensorList(Sensor.TYPE_ALL);
        StringBuilder sensorText = new StringBuilder();
        for (Sensor currentSensor : daftarSensor ) {
            sensorText.append(currentSensor.getName()).append(
                System.getProperty("line.separator"));
        }
        tvDaftarSensor.setText(sensorText);
    }
}

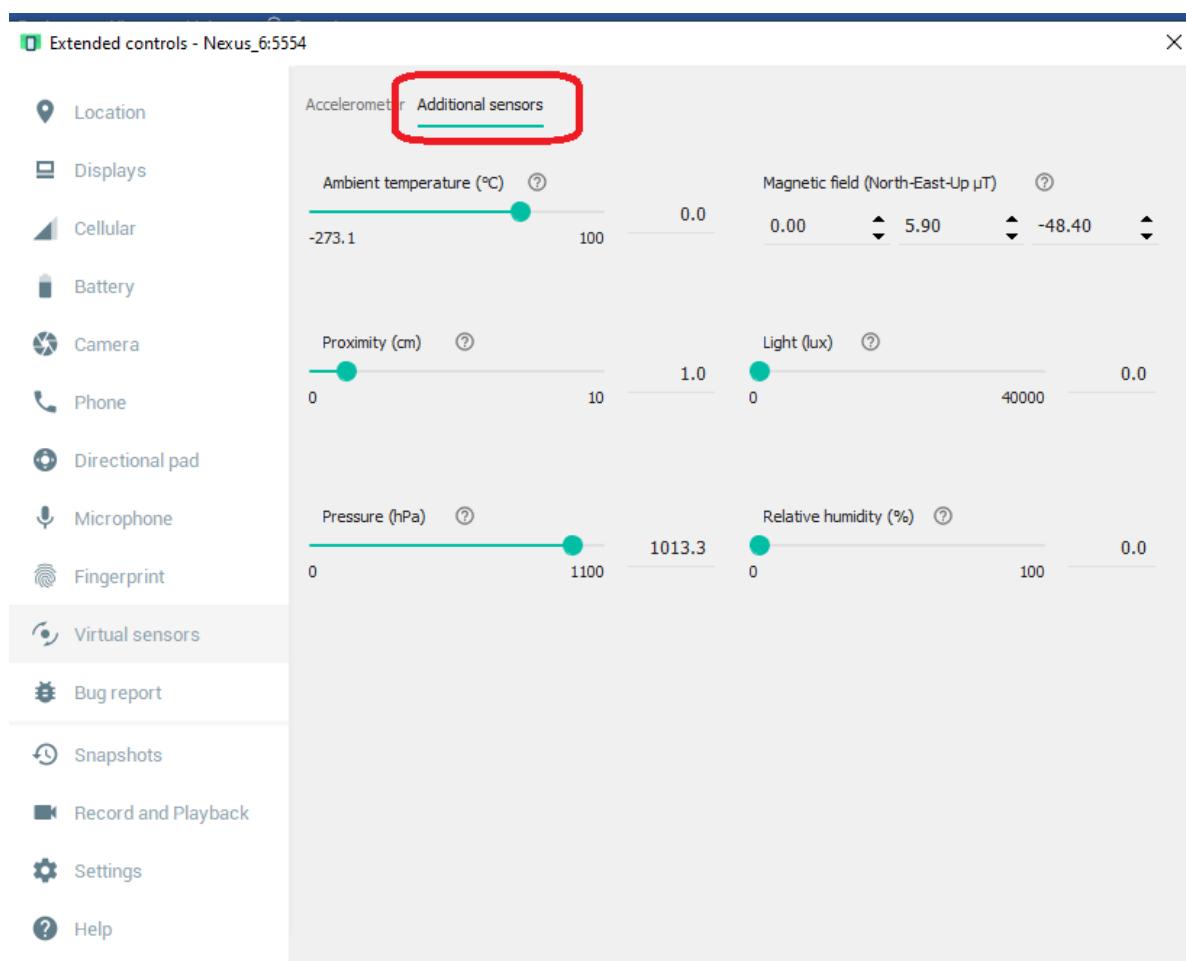
```

4. Jalankan App ini pada device anda dan perhatikan daftar dari sensor yang ada pada device anda. Satu device dengan device yang lain tentu saja berbeda daftar dari sensor tersebut tergantung dari sensor yang ada. Pada daftar sensor, sensor diawali dengan sebuah kode huruf atau angka

merepresentasikan fisikal hardware yang ada pada device. Huruf-huruf dan angka-angka mengidentifikasi pabrik yang membuat sensor tersebut dan nomor modelnya. Umumnya accelerometer, gyroscope, dan magnetometer adalah sensor fisik. Sementara daftar sensor yang tanpa kode huruf/angka adalah sensor virtual atau komposit yang disimulaskan pada software. Sensor-sensor ini menggunakan data dari satu atau lebih sensor fisik. Seperti sensor gravity menggunakan data dari accelerometer, gyroscope, dan magnetometer untuk menyediakan arah dan besaran gravitasi pada sistem koordinat perangkat smartphone.

5. Jalankan App ini pada emulator, karena Android Emulator adalah perangkat simulasi, maka semua sensor yang ada adalah sensor virtual. "Goldfish" adalah nama dari emulator pada kernel Linux.
6. Klik **more** button (tiga titik horizontal), lalu klik **Virtual Sensor** akan menampilkan setting dan nilai-nilai dari virtual sensor tersebut. Kemudian klik juga "**Additional Sensors**" untuk melihat daftar virtual sensor lainnya beserta setting dan nilai-nilainya.





Tutorial: Pembacaan Data Sensor

- Buat Android Studio Project yang baru dengan nama: **Week12b_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity_main**.
- Buka dan edit layout XML file **activity_main.xml**, hapus TextView bawaan dan ganti dengan dua buah TextView dengan nama **lightSensor** dan **proximitySensor**. Berikut skrip **activity_main.xml**:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/lightSensor"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/proximitySensor"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginTop="10dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/lightSensor" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

3. Langkah berikutnya adalah membuka dan mengedit file **MainActivity.java** untuk:
 - a. Menghubungkan dengan dua buah object TextView pada file layout dengan nama **tvLight** dan **tvProximity**.
 - b. Menambahkan dua variable member Sensor untuk sensor Light dan Proximity dengan nama **mSensorProximity** dan **mSensorLight**, dan sebuah **SensorManager** dengan nama **mSensorManager**. Assign **mSensorManager** dengan mengambil sensor bawaan seperti pada tutorial sebelumnya, lalu gunakan method **getDefaultSensor()** dengan parameter **Sensor.TYPE.LIGHT** dan **Sensor.TYPE.PROXIMITY** untuk mendapatkan sensor bawaan light dan proximity yang selanjutnya diassign ke dua variable sensor di atas.
 - c. Mendaftarkan listener untuk Sensor tersebut dengan mengimplement interface **SensorEventListener** dan meng-override callback method **onSensorChanged()** dan **onAccuracyChanged()**. Pada **onSensorChanged()** gunakan **event.sensor.getType()** untuk mendapatkan jenis sensor dan **event.value[0]** untuk mendapatkan nilai yang datang dari sensor tersebut. Gunakan switch untuk menampilkan nilai sensor sesuai dengan tipe sensornya.
 - d. Override **onStart()** method untuk mendaftarkan kedua buah sensor tadi.
 - e. Override **onStop()** method untuk me-unregister semua sensor dengan memanggil **unregisterListener()** method pada sensor manager.

Berikut kode java lengkap dari class **MainActivity.java**:

```

public class MainActivity extends AppCompatActivity implements
                                SensorEventListener {
    private TextView tvLight;
    private TextView tvProximity;
    private Sensor sensorLight;
    private Sensor sensorProximity;
    private SensorManager mSensorManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvLight = findViewById(R.id.lightSensor);
        tvProximity = findViewById(R.id.proximitySensor);
        mSensorManager = (SensorManager)
            getSystemService(Context.SENSOR_SERVICE);
        sensorProximity =
            mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
        sensorLight =
            mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        int sensorType = event.sensor.getType();
        float currentValue = event.values[0];
        switch(sensorType){
            case Sensor.TYPE_LIGHT:
                tvLight.setText("Light Sensor : " +
                    String.format("%.2f",currentValue));
                break;
            case Sensor.TYPE_PROXIMITY:
                tvProximity.setText("Proximity Sensor : " +
                    String.format("%.2f",currentValue));
                break;
        }
    }
    @Override
    protected void onStart() {
        super.onStart();
        if (sensorProximity != null) {
            mSensorManager.registerListener(this, sensorProximity,
                SensorManagerSENSOR_DELAY_NORMAL);
        }
        if (sensorLight != null) {
            mSensorManager.registerListener(this, sensorLight,
                SensorManagerSENSOR_DELAY_NORMAL);
        }
    }
}

```

```
@Override
protected void onStop() {
    super.onStop();
    mSensorManager.unregisterListener(this);
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {

}
}
```

4. Jalankan (run) project anda pada device anda dan coba lakukan perubahan nilai dengan mendekatkan dan menjauhkan object dari sensor Light dan sensor Proximity. Lokasi dari sensor tersebut biasanya berada pada bagian atas layar device anda.
5. Jalankan juga dengan menggunakan emulator. Coba buka Additional Sensor tab pada emulator anda dan ubahlah nilai proximity dan light, dan periksa apakah terjadi perubahan pada App anda.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning.
2. Lakukan pengujian terhadap app anda sesuai petunjuk tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI

MODUL 13

(GOOGLE MAP API)

DESKRIPSI TEMA

- Google Map API
 - Menampilkan Google Map di App
 - Menambahkan Marker dan Shape di Google Map

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat memanfaatkan Google Map untuk menunjukkan lokasi obyek tertentu yang direpresentasikan dengan Marker dan Shape (Circle, PolyLine, dan Polygon).

PENUNJANG PRAKTIKUM

1. Software Android Studio

LANGKAH-LANGKAH PRAKTIKUM

Tutorial: Google Map API

1. Sebelum membuat project yang memanfaatkan Google Location dan atau Google Map API, kita perlu menambahkan Google Play pada project kita dengan cara pilih menu **Tools > SDK Manager** lalu pilih **SDK Tools** Tab kemudian check **Google Play Services** selanjutnya klik "OK" atau "Apply".
2. Buat Android Studio Project yang baru dengan nama: **Week13_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 25: Android 7.1.1 (Nougat)**, Activity Template: **Google Map Activity**, Activity Name: **MapsActivity**, dan Layout Name: **activity_maps**.
3. Android Studio akan menampilkan file XML **google_maps_api.xml**. Buka URL yang ada pada file tersebut untuk membuat **Google Map API key**. Gunakan google account untuk masuk ke layanan ini, setelah mengikuti proses pembuatan API key. Setelah mendapatkan API key copy API key tersebut dan gantikan "**ENTER KEY HERE**" pada file **google_maps_api.xml**.
4. Buka dan edit layout file **activity_maps.xml** untuk mengganti root Layout menjadi **RelativeLayout** dan tambahkan tombol "**Tambah**" di atas Google Map fragment untuk dapat membuka Activity Map lain sehingga dapat menambahkan **marker** dan/atau **shape** object pada google map object. Berikut skrip XML dari layout file **activity_maps.xml**:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity">
    <fragment
        android:id="@+id/map"

        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <Button
        android:id="@+id/tambah"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#359c5e"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true"
        android:padding="8dp"
        android:layout_margin="5dp"
        android:text="TAMBAH"
        android:textColor="#ffffff" />
</RelativeLayout>

```

5. Buat sebuah Class dengan nama "**MarkerDanShape**" yang digunakan untuk merepresentasikan sebuah Marker atau Shape (Circle, PolyLine, dan Polygon). Class ini memiliki member variable String **jenis** (yang berisi "Marker", "Circle", "PolyLine", atau "Polygon"), String **keterangan**, double **radius**, List<LatLng> **titiks** yang berisi daftar titik-titik pada google map object, dan Object **objekPeta**.
Buat konstruktor, method getter dan method setter seperti berikut:

```

public class MarkerDanShape{
    private String jenis = "Marker";
    private String keterangan = "";
    private double radius = 0.0;
    private List<LatLng> titiks = new ArrayList<LatLng>();
    private Object objekPeta = null;
    public MarkerDanShape(String jenis, String keterangan,
                          double radius, List<LatLng> titiks){
        this.jenis = jenis;
        this.keterangan = keterangan;
        this.radius = radius;
        this.titiks = new ArrayList<LatLng>();
        this.titiks.addAll(titiks);
    }
}

```

```

public String getJenis(){ return this.jenis;}
public String getKeterangan() { return this.keterangan;}
public double getRadius() { return this.radius;}
public List<LatLng> getTitiks() { return this.titiks;}
public int getJumlahTitiks() {return this.titiks.size();}
public Object getObjekPeta(){return this.objekPeta;}

public void setKeterangan(String keterangan) {
    this.keterangan = keterangan;
}
public void setRadius(double radius) { this.radius = radius;}
public void setTitiks(List<LatLng> titiks) { this.titiks =
titiks;}
public void setObjekPeta(Object objekPeta) {
    this.objekPeta = objekPeta;
}
}

```

6. Buka dan modifikasi file **MapsActivity.java** untuk menambahkan member variable **mUiSetting** (**UiSetting**), **currentLocation** (**LatLng**), **daftarMarkerDanShape** (**List<MarkerDanShape>**), **lokalTitiks** (**List<LatLng>**), **btnTambah** (**Button**), dan constant **REQUEST_TAMBAH**. Lengkapi **onCreate()** dengan mereferensi **btnTambah** dan menambahkan **onClickListener** yang memanggil Activity Map lain untuk penambahan Marker dan Shape baru. Isikan **currentLocation** secara manual dengan posisi kampus UMN yaitu latitude: -6.2574591 longitude: 106.6183484. Isikan **daftarMarkerDanShape** dengan initial marker, circle, polyline, dan polygon object sebagai contoh masing-masing marker/shape. Buat sebuah method **gambarMarkerDanShape** ke google map object **mMap**. Ganti isi cari callback method **onMapReady()** dengan menambahkan **UiSetting** yang menampilkan Compas dan Zoom Control, panggil **moveCamera()** dengan titik tengah, **currentLocation** dengan zoom level **16**, dan panggil method **gambarMarkerDanShape()**. Berikut ini kode java sementara dari **MapsActivity.java**:

```

public class MapsActivity extends FragmentActivity
    implements OnMapReadyCallback {
    private UiSettings mUiSettings;
    private GoogleMap mMap;
    static LatLng currentLocation;
    private List<MarkerDanShape> daftarMarkerDanShape =
        new ArrayList<MarkerDanShape>();
    private List<LatLng> lokalTitiks = new ArrayList<LatLng>();
    private Button btnTambah;
    static final int REQUEST_TAMBAH = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
}

```

```

btnTambah = findViewById(R.id.tambah);
btnTambah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    }
});
currentLocation = new LatLng(-6.2574591,106.6183484);
lokalTitiks.add(currentLocation);
daftarMarkerDanShape.add(new MarkerDanShape("Marker",
    "Universitas Multimedia Nusantara", 0.0,lokalTitiks));
daftarMarkerDanShape.add(new MarkerDanShape("Circle",
    "Area 500 m dari UMN",500.0, lokalTitiks));
lokalTitiks.clear();
lokalTitiks.add(new LatLng(-6.256718, 106.618209));
lokalTitiks.add(new LatLng(-6.255982, 106.618434));
lokalTitiks.add(new LatLng(-6.256061, 106.621020));
lokalTitiks.add(new LatLng(-6.254611, 106.622085));
lokalTitiks.add(new LatLng(-6.254752, 106.622383));
daftarMarkerDanShape.add(new MarkerDanShape("PolyLine",
    "Jalur UMN ke Bethsaida",0.0, lokalTitiks));
lokalTitiks.clear();
lokalTitiks.add(new LatLng(-6.256302, 106.617534));
lokalTitiks.add(new LatLng(-6.256099, 106.619744));
lokalTitiks.add(new LatLng(-6.256558, 106.619851));
lokalTitiks.add(new LatLng(-6.259374, 106.618639));
lokalTitiks.add(new LatLng(-6.258659, 106.616740));
daftarMarkerDanShape.add(new MarkerDanShape("Polygon",
    "Area Kampus UMN",0.0, lokalTitiks));
lokalTitiks.clear();
}
protected void gambarMarkerDanShape(){
for(int i = 0 ; i < daftarMarkerDanShape.size(); i++){
    MarkerDanShape ms = daftarMarkerDanShape.get(i);
    if(ms.getJenis().equalsIgnoreCase("Marker")){
        ms.setObjekPeta( mMap.addMarker(new MarkerOptions()
            .position(ms.getTitiks().get(0))
            .title(ms.getKeterangan()))
    );
    }else if(ms.getJenis().equalsIgnoreCase("Circle")){
        ms.setObjekPeta( mMap.addCircle(new CircleOptions()
            .center(ms.getTitiks().get(0))
            .radius(ms.getRadius())
            .strokeColor(Color.YELLOW)
            .fillColor(Color.argb(30,255,255,0)))
    );
    }else if(ms.getJenis().equalsIgnoreCase("PolyLine")){
        PolylineOptions pl = new PolylineOptions();
        for(int j = 0; j < ms.getJumlahTitiks(); j++){
            pl.add(ms.getTitiks().get(j));
        }
        pl.color(Color.RED).width(10.0f);
        ms.setObjekPeta( mMap.addPolyline(pl));
    }
}
}

```

```

    }else if(ms.getJenis().equalsIgnoreCase("Polygon")){
        PolygonOptions pg = new PolygonOptions();
        for(int j = 0; j < ms.getJumlahTitiks(); j++){
            pg.add(ms.getTitiks().get(j));
        }
        pg.add(ms.getTitiks().get(0))
            .strokeColor(Color.BLUE)
            .strokeWidth(10.0f)
            .fillColor(Color.argb(20,0,255,255));
        ms.setObjekPeta( mMap.addPolygon(pg));
    }
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mUiSettings = mMap.getUiSettings();
    mUiSettings.setZoomControlsEnabled(true);
    mUiSettings.setCompassEnabled(true);
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
        currentLocation,16));
    gambarMarkerDanShape();
}
}

```

7. Selanjutnya buatlah Map Activity lain dengan cara klik kanan pada folder **app > java > id.ac.umn.week13_[NIM]** pilih **New > Google > Google Maps Activity** isi nama **TambahObject**, Title: Tambah Object, Hierarchical Parent: **id.umn.ac.week13_[NIM].MapsActivity**, serta layout: **activity_tambah_object**, lalu klik "Finish".
8. Buka dan edit layout file **activity_tambah_object.xml** untuk menambahkan **RadioButtonGroup** dengan nama **rgShape** yang berisi empat buah **RadioButton** (**marker**, **circle**, **polyline**, **polygon**), Dua buah **EditText** (**etKeterangan** dan **etRadius**). Sebuah **Button** dengan nama **simpan**, serta sebuah **MapFragment** dengan nama **tambahmap**. Berikut skrip XML dari **activity_tambah_object.xml**:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:map="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:name="com.google.android.gms.maps.SupportMapFragment"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".TambahObject" >
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="80dp"

```

```
        android:orientation="vertical"
        map:layout_constraintEnd_toEndOf="parent"
        map:layout_constraintStart_toStartOf="parent"
        map:layout_constraintTop_toTopOf="parent">
    <RadioGroup
        android:id="@+id/rgShape"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/marker"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:checked="true"
            android:text="Marker" />
        <RadioButton
            android:id="@+id/circle"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="Circe" />
        <RadioButton
            android:id="@+id/polyline"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="PolyLine" />
        <RadioButton
            android:id="@+id/polygon"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="Polygon" />
    </RadioGroup>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/etKeterangan"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="Keterangan"
            android:inputType="text" />
        <EditText
            android:id="@+id/etRadius"
            android:layout_width="100dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:ems="10"
```

```

        android:enabled="false"
        android:hint="radius"
        android:inputType="number" />
    <Button
        android:id="@+id/simpan"
        android:layout_width="70dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:enabled="false"
        android:text="Simpan" />
    </LinearLayout>
</LinearLayout>
<fragment
    android:id="@+id/tambahmap"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="0dp"
    android:layout_height="0dp"
    map:layout_constraintBottom_toBottomOf="parent"
    map:layout_constraintEnd_toEndOf="parent"
    map:layout_constraintStart_toStartOf="parent"
    map:layout_constraintTop_toBottomOf="@+id/linearLayout"
/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

9. Buka dan modifikasi file **TambahObject.java** yang mereferensi dan membuat objectobject View yang ada pada layout file di atas seperti: RadioGroup, EditText, Button dan Fragment Map. Tambahkan beberapa member variable bantu seperti int **pilihan**, List<LatLng> **titiks**, serta **MarkerDanShape**. Tambahkan **onCheckedChangeListener** pada **RadioGroup rgShape** dengan penanganan yang tepat untuk masing-masing **RadioButton** anggotanya. Tambahkan **OnClickListener** pada **Button simpan** yang mengirim balik data berupa detil dari object **MarkerDanShape** object. Note: **LatLng** adalah class yang non-serializable sehingga tidak dapat dikirim sebagai data pada Intent sehingga diganti dengan array of double.Tambahkan juga **OnMapClickListener** pada object google map **mMap** untuk menangani pengguna melakukan click pada object google map, lakukan ini didalam method **onMapReady()** karena object **mMap** di generate pada method callback ini. Berikut adalah kode java dari **TambahObject.java**:

```

public class TambahObject extends FragmentActivity
    implements OnMapReadyCallback {
    private int pilihan = 0;
    // 0 = Marker; 1 = Circle; 2 = Polyline; 3 = Polygon
    private GoogleMap mMap;
    private Button btnSimpan;
    private EditText etRadius;
    private EditText etKeterangan;
    private RadioGroup rgShape;
    private List<LatLng> titiks;
    private UiSettings mUiSettings;
```

```

private static MarkerDanShape objMS;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tambah_object);
    SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager()
        .findFragmentById(R.id.tambahmap);
    mapFragment.getMapAsync(this);
    btnSimpan = findViewById(R.id.simpan);
    etRadius = findViewById(R.id.etRadius);
    etKeterangan = findViewById(R.id.etKeterangan);
    rgShape = findViewById(R.id.rgShape);
    titiks = new ArrayList<LatLng>();
    rgShape.setOnCheckedChangeListener(new RadioGroup.
        OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                titiks.clear();
                btnSimpan.setEnabled(false);
                if(mMap != null) {
                    mMap.clear();
                }
                switch(checkedId){
                    case R.id.marker:
                        pilihan = 0;
                        etRadius.setEnabled(false);
                        break;
                    case R.id.circle:
                        pilihan = 1;
                        etRadius.setEnabled(true);
                        break;
                    case R.id.polyline:
                        pilihan = 2;
                        etRadius.setEnabled(false);
                        break;
                    case R.id.polygon:
                        pilihan = 3;
                        etRadius.setEnabled(false);
                        break;
                }
            }
        });
    btnSimpan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(!objMS.getKeterangan().equalsIgnoreCase(
                etKeterangan.getText().toString())){
                objMS.setKeterangan(etKeterangan.getText().toString());
            }
            Intent jawabIntent = new Intent();
            jawabIntent.putExtra("jenis",objMS.getJenis());
        }
    });
}

```

```

        jawabIntent.putExtra("keterangan", objMS.getKeterangan());
        jawabIntent.putExtra("radius", objMS.getRadius());
        List<LatLng> mTitiks = objMS.getTitiks();
        double[] dTitiks = new double[mTitiks.size()*2];
        for(int i = 0; i < mTitiks.size(); i++){
            dTitiks[2*i] = mTitiks.get(i).latitude;
            dTitiks[2*i+1] = mTitiks.get(i).longitude;
        }
        jawabIntent.putExtra("titiks",dTitiks);
        setResult(RESULT_OK,jawabIntent);
        finish();
    }
}
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mUiSettings = mMap.getUiSettings();
    mUiSettings.setZoomControlsEnabled(true);
    mUiSettings.setCompassEnabled(true);
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
        MapsActivity.currentLocation,15));
    mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
        @Override
        public void onMapClick(LatLng latLng) {
            switch(pilihan){
                case 0:
                    if(titiks.size()==0){
                        titiks.add(latLng);
                    } else {
                        titiks.set(0,latLng);
                    }
                    if(mMap != null){
                        mMap.clear();
                        mMap.addMarker(new
                            MarkerOptions().position(titiks.get(0))
                            .title(etKeterangan.getText().toString()));
                        objMS = new MarkerDanShape("Marker",
                            etKeterangan.getText().toString(),0.0,titiks);
                        btnSimpan.setEnabled(true);
                    }
                    break;
                case 1:
                    if(titiks.size()==0){
                        titiks.add(latLng);
                    } else {
                        titiks.set(0,latLng);
                    }
                    if(mMap != null){
                        mMap.clear();
                        mMap.addMarker(new MarkerOptions()

```

```
.position(titiks.get(0))
.title(etKeterangan.getText().toString()));
String rad = etRadius.getText().toString()
.trim();
if(rad.length() > 0
    && Double.parseDouble(rad) > 0.0){
    double radius = Double.parseDouble(rad);
    mMap.addCircle(new CircleOptions()
        .radius(radius).center(titiks.get(0))
        .strokeColor(Color.YELLOW)
        .fillColor(Color.argb(30,255,255,0)));
    objMS = new MarkerDanShape("Circle",
        etKeterangan.getText().toString(),
        radius, titiks);
    btnSimpan.setEnabled(true);
}
}
break;
case 2:
titiks.add(latLng);
if(mMap != null) {
    mMap.clear();
}
if(titiks.size()==1){
    mMap.addMarker(new MarkerOptions()
        .position(titiks.get(0)));
} else {
    PolylineOptions pl = new PolylineOptions()
        .color(Color.RED).width(10.0f);
    for(int i = 0; i < titiks.size(); i++){
        mMap.addMarker(new MarkerOptions()
            .position(titiks.get(i)));
        pl.add(titiks.get(i));
    }
    mMap.addPolyline(pl);
    objMS = new MarkerDanShape("PolyLine",
        etKeterangan.getText().toString(),
        0, titiks);
    btnSimpan.setEnabled(true);
}
break;
case 3:
titiks.add(latLng);
if(mMap != null) {
    mMap.clear();
}
if(titiks.size() == 1) {
    mMap.addMarker(new MarkerOptions()
        .position(titiks.get(0)));
} else if (titiks.size() == 2){
    PolylineOptions pl = new PolylineOptions()
        .color(Color.RED).width(10.0f);
    mMap.addPolyline(pl);
    objMS = new MarkerDanShape("PolyLine",
        etKeterangan.getText().toString(),
        0, titiks);
    btnSimpan.setEnabled(true);
}
```

```
        for(int i = 0; i < titiks.size(); i++){
            mMap.addMarker(new MarkerOptions()
                .position(titiks.get(i)));
            pl.add(titiks.get(i));
        }
        mMap.addPolyline(pl);
    } else if (titiks.size() > 2){
        PolygonOptions pg = new PolygonOptions()
            .strokeColor(Color.BLUE)
            .strokeWidth(10.0f)
            .fillColor(Color.argb(20,0,255,255));
        for(int i = 0; i < titiks.size(); i++){
            mMap.addMarker(new MarkerOptions()
                .position(titiks.get(i)));
            pg.add(titiks.get(i));
        }
        pg.add(titiks.get(0));
        mMap.addPolygon(pg);
        objMS = new MarkerDanShape("Polygon",
            etKeterangan.getText().toString(),
            0, titiks);
        btnSimpan.setEnabled(true);
    }
    break;
}
});
```

10. Selanjutnya adalah melengkapi **MapsActivity.java** dengan melengkapi method **onClick()** pada **btnTambah** seperti berikut:

```
btnTambah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent tambahIntent = new Intent(MapsActivity.this,
                                         TambahObject.class);
        startActivityForResult(tambahIntent, REQUEST_TAMBAH);
    }
});
```

Lalu tambahkan implementasi method callback **onActivityResult()** dengan menerima kiriman data object baru dari **TambahObject** Activity dan tambahkan ke **daftarMarkerDanShape** lalu gambar ulang google map berikut seluruh isi dari daftar di atas. Berikut implementasi dari method **onActivityResult** pada **MapsActivity.java**:

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
```

```
if(resultCode == RESULT_OK && requestCode == REQUEST_TAMBAH) {  
    String jenis = data.getStringExtra("jenis");  
    String keterangan = data.getStringExtra("keterangan");  
    double radius = data.getDoubleExtra("radius",0.0);  
    double[] dTitiks = data.getDoubleArrayExtra("titiks");  
    List<LatLng> mTitiks = new ArrayList<LatLng>();  
    for(int i = 0; i < dTitiks.length/2; i++){  
        mTitiks.add(new LatLng(dTitiks[2*i],dTitiks[2*i + 1]));  
    }  
    MarkerDanShape mds = new MarkerDanShape(jenis,keterangan,  
                                              radius,mTitiks);  
    daftarMarkerDanShape.add(mds);  
    if(mMap != null){  
        mMap.clear();  
        gambarMarkerDanShape();  
    }  
}  
}
```

11. Jalankan App ini pada device anda atau pada emulator dan lakukan percobaan dengan menggunakan zoom button, geser google map dan lain-lain. Lakukan juga penambahan Marker, Circle, PolyLine, dan/atau Polygon. Perhatikan apa yang terjadi dan catat.

Latihan, Tugas, dan Eksperimen:

1. Ikuti langkah-langkah di atas sampai project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning.
2. Lakukan pengujian terhadap app anda sesuai petunjuk tutorial di atas. Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.

REFERENSI