

# **LAPORAN UJIAN AKHIR SEMESTER**

**KAJIAN.ID**

Disusun untuk memenuhi tugas akhir daring Pemrograman Mobile 2  
oleh Bapak Nandang Hermanto, M.Kom.



## **DISUSUN OLEH**

Dwi Candra Permana	18.11.0004 / IF 2018 A
Mar'Atun Solehah	18.11.0017 / IF 2018 A
Muhammad Rizky Bayu Pratama	18.11.0258 / IF 2018 A
Fendi Riawan	18.11.0268 / IF 2018 A
Diyah Primasari	18.11.0289 / IF 2018 A

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS AMIKOM PURWOKERTO**

**PURWOKERTO**

**2020**

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>1</b>
<b>BAB I: PENDAHULUAN</b>	<b>2</b>
A. LATAR BELAKANG .....	2
B. RUMUSAN MASALAH .....	2
C. TUJUAN.....	3
<b>BAB II: PEMBAHASAN .....</b>	<b>4</b>
A. KAJIAN.ID USER .....	4
B. KAJIAN.ID USTADZ.....	8
<b>BAB III: <i>SOURCE CODE</i> .....</b>	<b>23</b>
A. ALAMAT <i>REPOSITORY</i> .....	23
B. KAJIAN.ID USER .....	23
C. KAJIAN.ID USTADZ.....	143
<b>BAB IV: KESIMPULAN DAN SARAN .....</b>	<b>284</b>
A. KESIMPULAN .....	284
B. SARAN.....	284

## **BAB I PENDAHULUAN**

### **A. LATAR BELAKANG**

Di masa pandemi ini, kegiatan sehari-hari yang dilakukan masyarakat terbatas. Masyarakat hanya boleh keluar rumah jika ada kepentingan yang mendesak. Jika keluar rumah, masyarakat tidak boleh berkerumun atau harus jaga jarak untuk mencegah penyebaran virus COVID-19.

Kajian adalah kegiatan menyampaikan ilmu agama Islam yang dilakukan oleh pembicara ke masyarakat. Dalam situasi normal, masyarakat tentunya akan datang ke lokasi di mana diadakannya kajian. Setiap hari, di suatu daerah bisa ada satu atau beberapa tempat yang mengadakan kajian. Jadwal kajian bisa didapat oleh masyarakat dari media sosial. Hal tersebut menjadi kurang efektif, karena masyarakat akan berulang kali melihat jadwal kajian dengan membuka dan menutup informasi yang didapatkan dari media sosial. Belum lagi jika masyarakat lupa menyimpan informasi jadwal kajian.

Dalam membantu pemerintah untuk mengurangi rantai penyebaran virus corona, semua kegiatan yang menimbulkan kerumunan massa sebaiknya dihindari, seperti dalam pelaksanaan kajian Yang selalu dihadiri oleh Massa Yang berkumpul dalam jumlah banyak. Hal tersebut tentunya sangat bertentangan dengan program pemerintah untuk memutus rantai penyebaran COVID-19. Dengan kondisi tersebut, kajian perlu dilaksanakan secara online. Masyarakat tetap bisa mengikuti kajian Yang diinginkan tanpa rasa khawatir akan penyebaran COVID-19. Untuk menghindari kerumunan di masa pandemi seperti saat ini, kajian dilakukan dengan bersifat online. hal tersebut cukup efektif untuk mengurangi penyebaran virus corona dan tetap melaksanakan kajian yang sudah dijadwalkan.

### **B. RUMUSAN MASALAH**

- a. Bagaimana cara efektif agar masyarakat tetap melakukan kajian rutin dikala pandemi?
- b. Bagaimana cara masyarakat mendapatkan informasi jadwal kajian secara online tanpa harus melihat sosial media berkali-kali?
- c. Bagaimana cara masyarakat membaca artikel islami secara offline dengan mudah?
- d. Apa saja fitur-fitur dalam aplikasi Kajian.ID User?
- e. Bagaimana cara menggunakan aplikasi Kajian.ID User?
- f. Bagaimana cara ustadz/ustadzah menyampaikan kajian yang dilaksanakan dikala pandemi?
- g. Bagaimana cara ustadz/ustadzah menulis artikel islami dengan mudah?
- h. Apa saja fitur-fitur dalam aplikasi Kajian.ID Ustadz?
- i. Bagaimana cara menggunakan aplikasi Kajian.ID Ustadz?

### **C. TUJUAN**

- a. Memudahkan masyarakat tetap melakukan kajian rutin dikala pandemi
- b. Memudahkan masyarakat mendapatkan informasi jadwal kajian secara online tanpa harus melihat sosial media berkali-kali
- c. Memudahkan masyarakat membaca Artikel Islami secara offline dengan mudah
- d. Memudahkan masyarakat memahami fitur-fitur dalam aplikasi Kajian.ID User
- e. Memudahkan masyarakat memahami aplikasi Kajian.ID User
- f. Memudahkan ustadz/ustadzah menyampaikan kajian yang dilaksanakan ketika pandemi
- g. Memudahkan ustadz/ustadzah menulis Artikel Islami dengan mudah
- h. Memudahkan ustadz/udtazah memahami fitur-fitur dalam aplikasi Kajian.ID Ustadz
- i. Memudahkan ustadz/udtazah menggunakan aplikasi Kajian.ID Ustadz

## **BAB II PEMBAHASAN**

### **A. KAJIAN.ID USER**

Aplikasi Kajian.ID ditujukan seluruh kalangan anak muda sampai orang tua untuk mengetahui info kajian yang telah di jadwal kan agar sebagai umat islam bisa lebih tanggap mengikuti kajian tanpa bingung mencari jadwal kajian secara manual dan dapat lebih memperdalam wawasan agama islam dalam kajian yang tersedia . Aplikasi Kajian.ID ini dapat melihat artikel ataupun kajian yang diadakan atau akan dilaksanakan.

Berikut fitur-fitur yang dibawakan di Kajian.ID:

- Menyimpan artikel atau dan pengingat kajian Islami,
- menghapus artikel atau kajian yang telah disimpan,
- Melihat masjid atau lokasi yang biasanya diadakan kajian.
- Tema terang/gelap yang membantu pengguna aplikasi ini mengurangi cahaya yang ditangkap oleh mata, dan
- Membagikan artikel atau kajian ke media sosial.

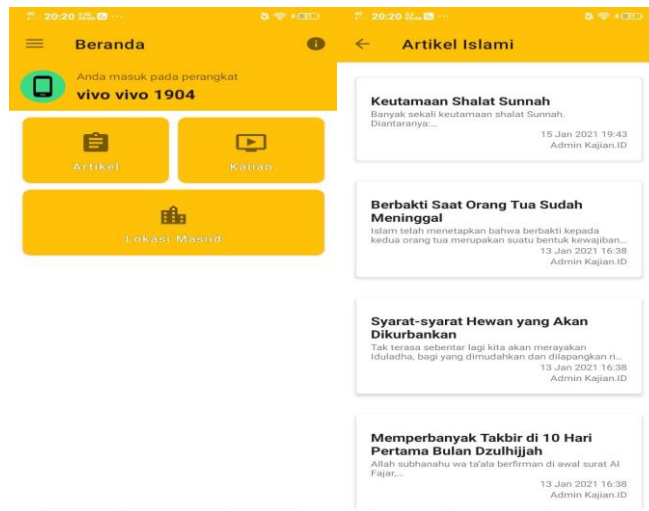
Berikut adalah tata cara penggunaan aplikasi ini.


#### **1. Langkah awal sebelum menggunakan aplikasi**

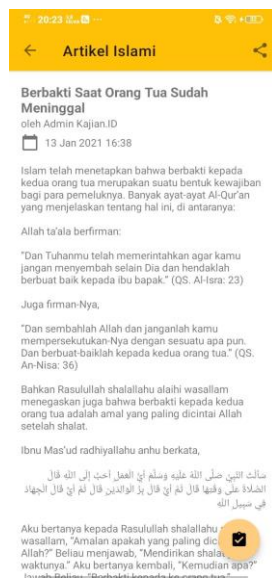
Saat pertama kali menginstal aplikasi ini, Anda akan dikenali sekilas mengenai aplikasi ini. Tunggu loading tampilan awal aplikasi membuka halaman selanjutnya, sudah selesai.



2. Cara membaca artikel, menyimpan artikel dan menyebarluaskan artikel Anda. Di layar utama, pilih tab “Artikel” lalu pilih artikel yang akan dipilih dan tekan tombol “Simpan Post” di pojok kanan bawah layar.

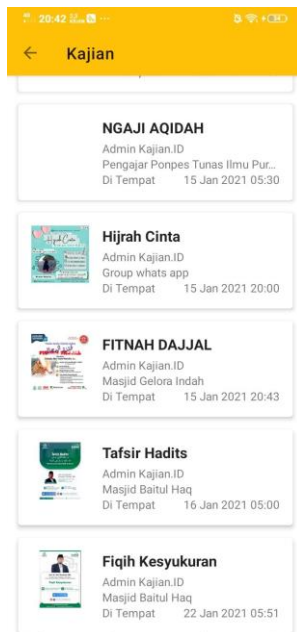



Untuk Menyimpan artikel yang akan disimpan tekan tombol “Simpan Post” di pojok kanan bawah layar. dan untuk membagikan atau menyebarluaskan artikel yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca artikel dan pilih ke aplikasi mana artikel Anda akan dibagikan.



3. Cara membaca kajian, menyimpan/mengingat kajian dan menyebarluaskan kajian Anda.

Di layar utama, pilih tab “Kajian” lalu pilih kajian yang akan dipilih.



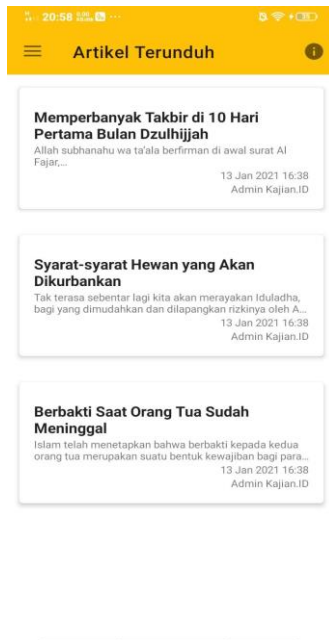
Untuk pengingat kajian yang sudah dipilih dan tekan tombol “Lonceng Post” di pojok kanan bawah layar. dan untuk membagikan atau menyebarkan artikel yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca artikel dan pilih ke aplikasi mana artikel Anda akan dibagikan.



Untuk melihat masjid di sekitar kita, Di layar utama, pilih tab “Lokasi Masjid” lalu pilih Lokasi.

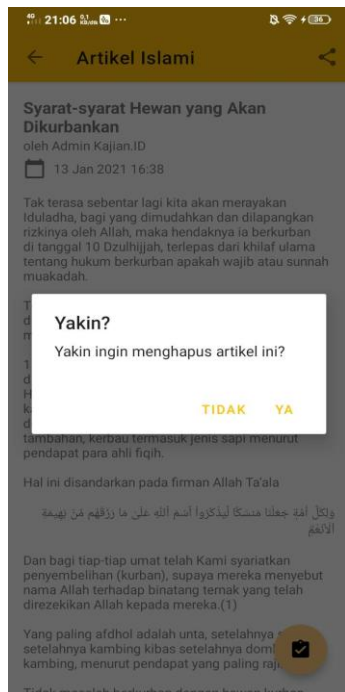


4. Jika kalian ingin melihat Artikel yang sudah tersimpan, silahkan tekan tombol pada layar utama pojok kiri atas, lalu pilih “Artikel Terunduh”



5. Jika kalian ingin menghapus Artikel yang sudah dibaca, silahkan pilih artikel tersebut lalu tekan tombol pada pojok kanan bawah. Lalu tekan “YA”





6. Jika kalian ingin melihat pengingat Kajian yang sudah tersimpan, silahkan tekan tombol pojok kiri atas, lalu pilih “Kajian Tersimpan”



7. Jika kalian ingin menghapus Pengingat Kajian yang sudah dibaca, silahkan pilih Kajian tersebut lalu tekan tombol pada pojok kanan bawah. Lalu tekan “YA”

## B. KAJIAN.ID USTADZ

Aplikasi Kajian.ID Ustadz ditujukan untuk para ulama, ustadz/ustadzah, *event organizer* keagamaan, maupun orang yang mengelola masjid tersebut untuk menyebarluaskan kajian yang mereka adakan. Aplikasi Kajian.ID Ustadz ini dapat

mengelola artikel ataupun kajian yang diadakan atau akan dilaksanakan. Berikut fitur-fitur yang dibawakan di Kajian.ID Ustadz:

- Menambah, mengubah, atau menghapus artikel Islami,
- Menambah, mengubah, atau menghapus kajian yang akan atau telah dilaksanakan,
- Mengelola masjid atau lokasi yang biasanya diadakan kajian.
- Tema terang/gelap yang membantu pengguna aplikasi ini mengurangi cahaya yang ditangkap oleh mata,
- Mengunggah alamat Video atau Live Streaming dari YouTube, dan
- Membagikan artikel atau kajian ke media sosial.

Berikut adalah tata cara penggunaan aplikasi ini.

1) Langkah awal sebelum menggunakan aplikasi

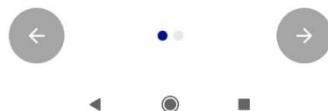
- Saat pertama kali menginstal aplikasi ini, Anda akan dikenali sekilas mengenai aplikasi ini. Tekan tombol → untuk membuka halaman selanjutnya, atau tekan tombol ✓ .bila layar pengenalan sudah selesai.



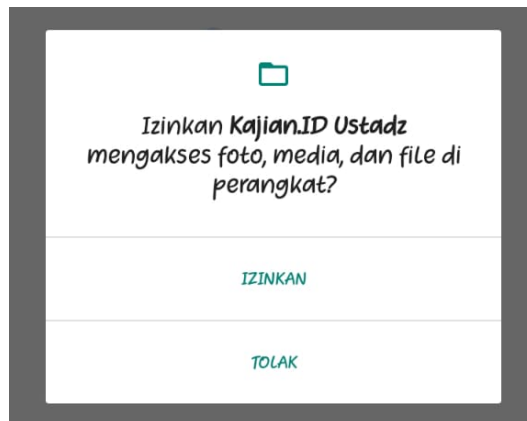
*Jalin hubungan bersama  
masyarakat*



*Buatlah pengumuman dan informasi terkait kajian  
yang akan Anda adakan!*



- Berikan izin pengaksesan media penyimpanan di *smartphone* Anda. Izin ini hanya untuk membaca data foto yang akan Anda unggah untuk keperluan menyebarluaskan informasi kajian di aplikasi ini.



- Setelah layar pengenalan selesai, anda akan diarahkan menuju layar Masuk sebagai Ustadz. Silakan masukkan Username dan Kata Sandi (*password*) dalam kolom yang sudah disediakan. Apabila Anda belum memiliki akun, Anda bisa menekan tombol “Daftar” untuk menuju aplikasi WhatsApp dan kirim pesan serta isi sesuai format yang disediakan. Tim kami akan membuatkan akun untuk Anda. Setelah selesai, tekan tombol “Login”.



Masuk sebagai Ustadz

Username

Password

☐ Lihat Password

Daftar

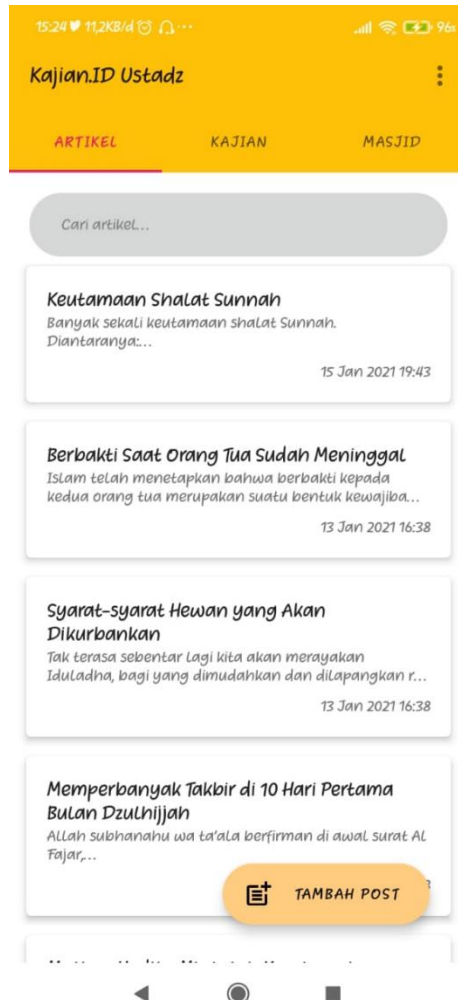
Login



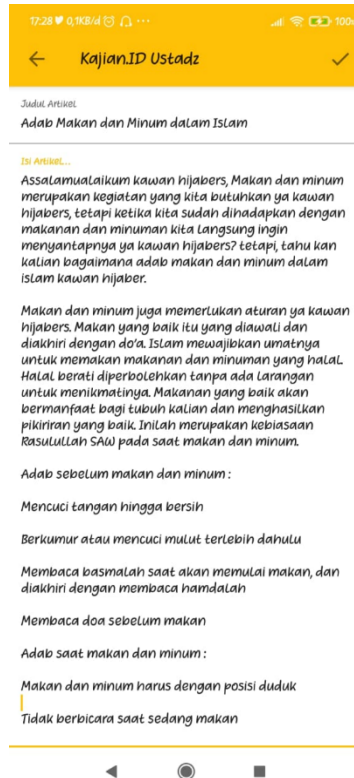
- Aplikasi akan menyimpan status login terakhir Anda dan akan selalu diingat sampai Anda melakukan *Logout* dari aplikasi.

2) Cara mengunggah artikel baru, membaca artikel, dan menyebarluaskan artikel Anda

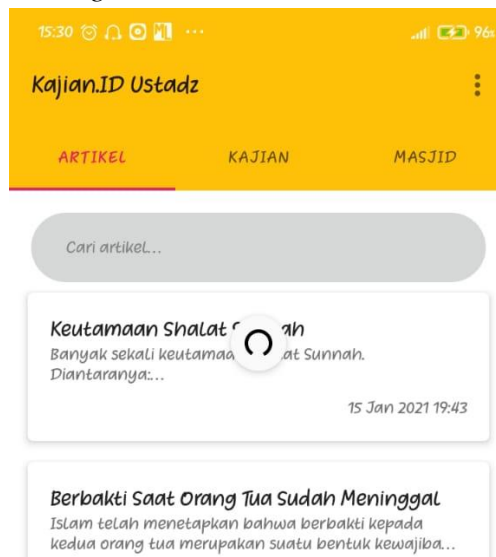
- Di layar utama, pilih tab “Artikel” dan tekan tombol “Tambah Post” di pojok kanan bawah layar.




- Silahkan isi judul dan isi artikel pada halaman yang sudah disediakan. Setelah selesai mengisi, tekan tombol ✓ di pojok kanan atas layar.



- Artikel berhasil diunggah. Untuk memperbarui daftar artikel, tarik daftar artikel dari atas ke bawah layar sampai muncul animasi indikator *loading*.

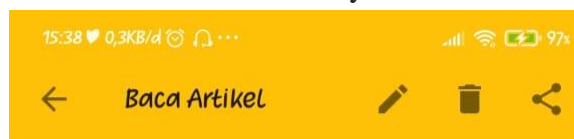


- Untuk melihat isi artikel yang sudah atau pernah Anda buat, cukup pilih artikel mana yang akan Anda baca.
- Untuk membagikan atau menyebarkan artikel yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca artikel dan pilih ke aplikasi mana artikel Anda akan dibagikan.

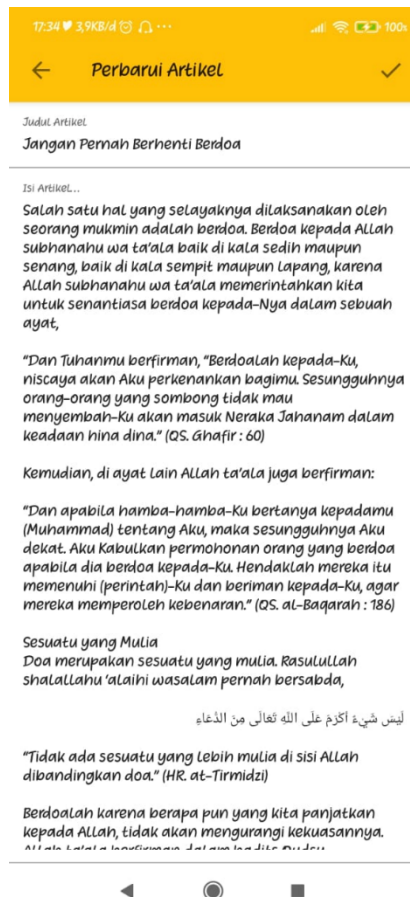


### 3) Cara mengubah judul atau isi artikel

- Di layar utama, pilih tab “Artikel” dan pilih artikel mana yang ingin Anda ubah judul atau isinya.
- Tekan tombol ✎ di atas layar.

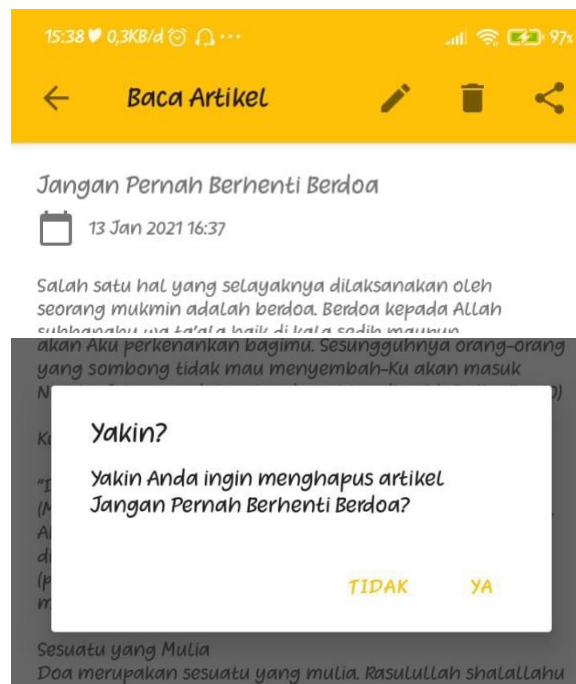


- Silahkan ubah judul atau isi artikel tersebut. Tekan tombol ✓ di pojok kanan atas layar apabila sudah selesai mengubah artikel.

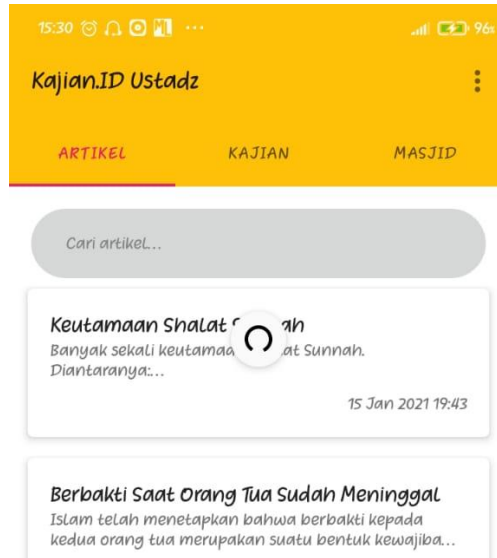


#### 4) Cara menghapus artikel

- Di layar utama, pilih tab “Artikel” dan pilih artikel mana yang ingin Anda hapus.
- Tekan tombol 🗑 di atas layar dan tekan tombol “Ya” untuk menghapus artikel.

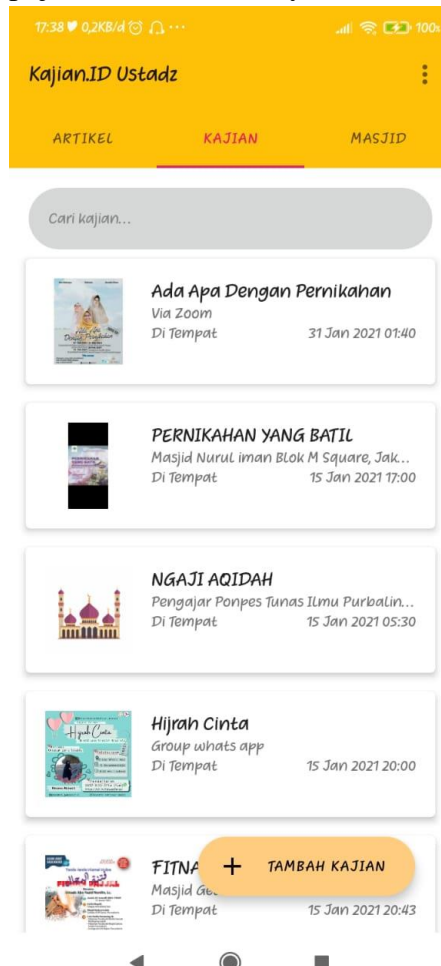


- Artikel berhasil dihapus. Untuk memperbarui daftar artikel, tarik daftar artikel dari atas ke bawah layar sampai muncul animasi indikator loading.



##### 5) Cara mengunggah dan menyebarluaskan informasi kajian

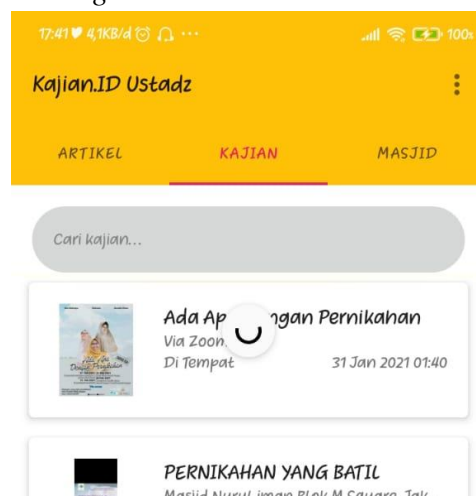
- Di layar utama, pilih tab “Kajian” dan tekan tombol “Tambah Kajian” di pojok kanan bawah layar.






- Silahkan isi informasi mengenai kajian pada halaman yang sudah disediakan. Setelah selesai mengisi, tekan tombol ✓ di pojok kanan atas layar.

- Kajian berhasil diunggah. Untuk memperbarui daftar kajian, tarik daftar kajian dari atas ke bawah layar sampai muncul animasi indikator *loading*.



- Untuk melihat isi kajian yang sudah atau pernah Anda buat, cukup pilih kajian mana yang akan Anda baca.

- Untuk membagikan atau menyebarluaskan kajian yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca informasi kajian dan pilih ke aplikasi mana kajian Anda akan dibagikan.





*Ada Apa Dengan Pernikahan*  
Via Zoom  
oleh Admin Kajian.ID

*Di Tempat*  
Waktu Pengumuman 15 Jan 2021 19:24  
Waktu Pelaksanaan 31 Jan 2021 01:40

*Deskripsi:*  
~Alhamdulillah~

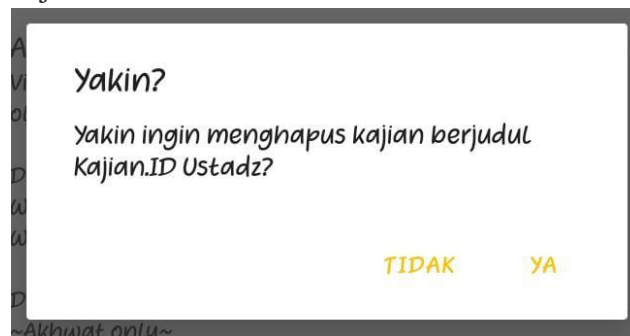
#### 6) Cara mengubah atau memperbarui informasi kajian

- Di layar utama, pilih tab “Kajian” dan pilih kajian mana yang ingin Anda ubah.
- Tekan tombol  di atas layar.
- Silahkan ubah data informasi kajian tersebut. Tekan tombol  di pojok kanan atas layar apabila sudah selesai mengubah informasi kajian.



#### 7) Cara menghapus informasi kajian

- Di layar utama, pilih tab “Kajian” dan pilih kajian mana yang ingin Anda hapus.
- Tekan tombol 🗑 di atas layar dan tekan tombol “Ya” untuk menghapus kajian.



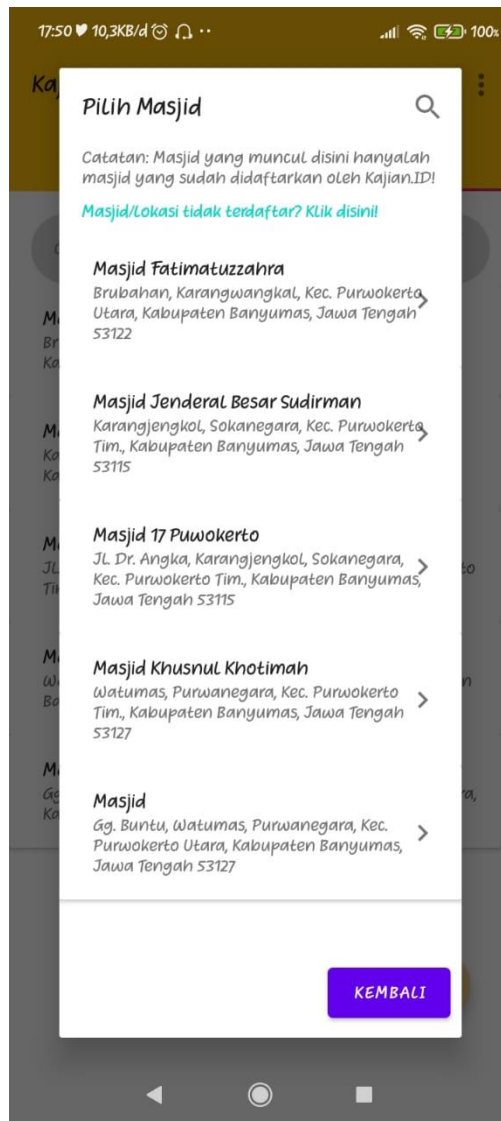
- Kajian berhasil dihapus. Untuk memperbarui daftar kajian, tarik daftar kajian dari atas ke bawah layar sampai muncul animasi indikator *loading*.

#### 8) Cara menambah masjid atau lokasi untuk dikelola dalam kajian

- Di layar utama, pilih tab “Masjid” dan tekan tombol “Tambah Masjid”

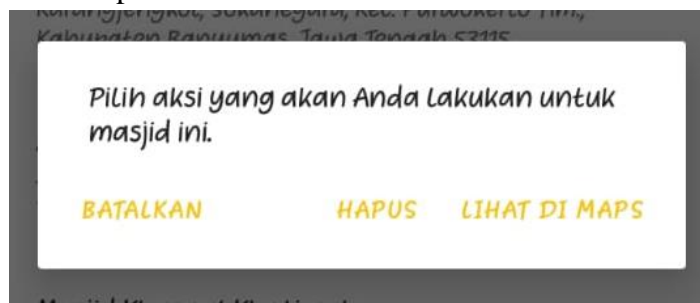


- Silahkan pilih masjid mana yang ingin Anda kelola. Masjid yang ada pada daftar ini hanya masjid yang sudah didaftarkan di server Kajian.ID. Untuk mendaftarkan lokasi baru, ikuti instruksi yang terdapat pada layar.

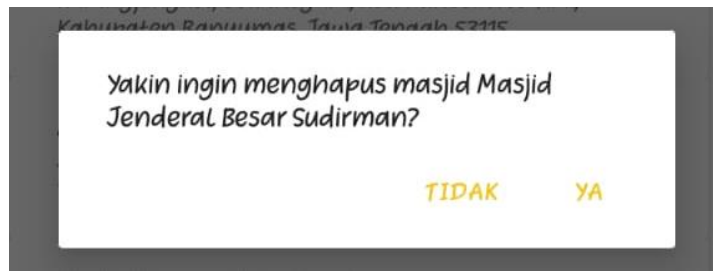


9) Cara menghapus masjid yang dikelola

- Di layar utama, pilih tab “Masjid” dan pilih masjid mana yang ingin Anda hapus.



- Pilih “Hapus” dan tekan “Ya” untuk menghapus.

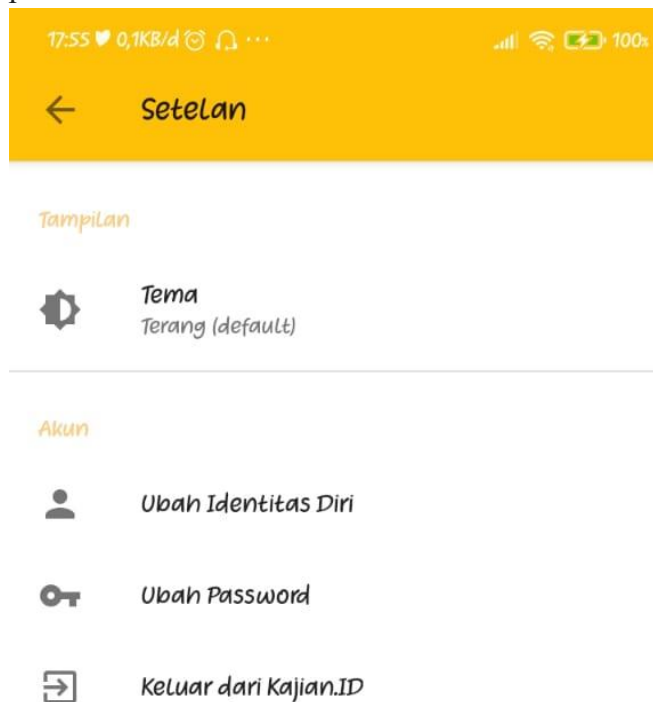


- Masjid telah dihapus. Perlu diingat, kajian yang pernah diadakan dan disimpan di masjid ini juga ikut terhapus. Namun, masjid tersebut tetap ada dalam server Kajian.ID.

#### 10) Cara mengubah tema

Catatan: Untuk versi Android 9 “Pie” keatas, fitur ini secara otomatis akan mengikuti konfigurasi sistem apakah menggunakan tema gelap atau terang.

- Di layar utama, tekan tombol tiga titik (⋮) di pojok kanan atas layar dan pilih “Setelan”



- Pilih “Tema” dan pilih tema mana yang ingin Anda terapkan di aplikasi Kajian.ID Ustadz.

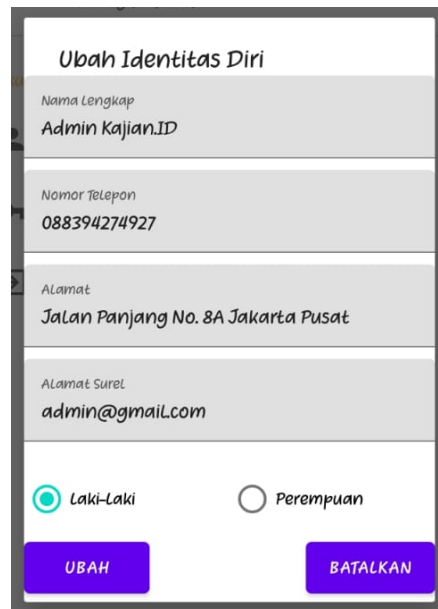


11) Cara mengubah identitas diri

- Di layar utama, tekan tombol tiga titik ( : ) di pojok kanan atas layar dan pilih “Setelan”
- Pilih “Ubah Identitas Diri” dan silahkan Anda ubah data diri Anda. Jika sudah, klik tombol “Ubah”.

12) Cara mengubah kata sandi (*password*) akun Kajian.ID Ustadz

- Di layar utama, tekan tombol tiga titik ( : ) di pojok kanan atas layar dan pilih “Setelan”
- Pilih “Ubah Password” dan silahkan ubah kata sandi Anda. Jika sudah, klik tombol “Ubah”.



Ubah Identitas Diri

Nama Lengkap  
Admin Kajian.ID

Nomor Telepon  
088394274927

Alamat  
Jalan Panjang No. 8A Jakarta Pusat

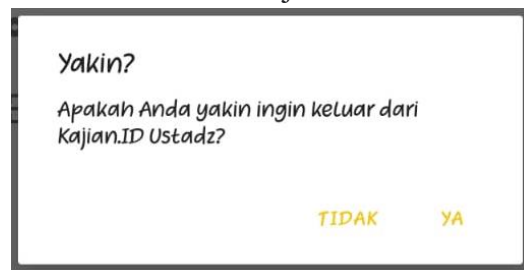
Alamat Surel  
admin@gmail.com

☒ Laki-Laki ☐ Perempuan

UBAH BATALKAN

13) Cara mengeluarkan akun (*logout*) dari Kajian.ID Ustadz

- Di layar utama, tekan tombol tiga titik ( : ) di pojok kanan atas layar dan pilih “Setelan”
- Pilih “Keluar dari Kajian.ID” dan tekan “Ya” untuk keluar.



Yakin?

Apakah Anda yakin ingin keluar dari Kajian.ID Ustadz?

TIDAK YA

## BAB III

### SOURCE CODE

#### A. ALAMAT REPOSITORY

- Alamat repository untuk source code Kajian.ID User dan Kajian.ID Ustadz ada di [https://github.com/IF18A-Kajian-ID/kajianid\\_android](https://github.com/IF18A-Kajian-ID/kajianid_android).
- Alamat repository untuk source code Kajian.ID Server ada di <https://github.com/IF18A-Kajian-ID/kajianid-webserver>.

#### B. KAJIAN.ID USER

##### a. Splash Screen

##### a) Layout

```
<?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=".activities.SplashActivity">

    <ImageView
        android:id="@+id/nama_slogan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/logo"
        android:src="@drawable/icon"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/version"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nama_slogan" />

    <TextView
        android:id="@+id/version"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="40dp"
        android:text="@string/versinya"
        android:textColor="?attr/editTextColor"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```



```

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/copyright"
            android:textColor="?attr/editTextColor"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/version"
            app:layout_constraintVertical_bias="0.3" />
    </androidx.constraintlayout.widget.ConstraintLayout>

```

## b) Program

```

package com.kajianid.android.activities

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import com.kajianid.android.R
import com.kajianid.android.databinding.ActivitySplashBinding

class SplashActivity : AppCompatActivity() {
    private var binding: ActivitySplashBinding? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivitySplashBinding.inflate(layoutInflater)
        setContentView(binding?.root)

        binding?.version?.text = "Version
${getString(R.string.app_version)}"

        Handler(Looper.getMainLooper()).postDelayed({
            startActivity(Intent(this@SplashActivity,
MainActivity::class.java))
            finishAffinity()
        }, 2000)
    }
}

```

## b. Tampilan Utama

### a) Layout

- activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    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:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        tools:openDrawer="start"
        tools:context=".activities.MainActivity">

        <include
            android:id="@+id/app_bar_main"
            layout="@layout/app_bar_main"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <com.google.android.material.navigation.NavigationView
            android:id="@+id/nav_main"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            app:itemBackground="@android:color/transparent"
            app:itemTextColor="?android:textColorPrimary"
            app:itemIconTint="?android:textColorPrimary"
            android:fitsSystemWindows="true"
            app:headerLayout="@layout/nav_header_main"
            app:menu="@menu/menu_main_drawer" />

    </androidx.drawerlayout.widget.DrawerLayout>

```

- app\_bar\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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=".activities.MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:elevation="0dp">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

- nav\_header\_main.xml

```
<?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="@dimen/nav_header_height"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="?attr/colorPrimary"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".activities.MainActivity">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/nav_header_desc"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:src="@mipmap/ic_launcher" />

    <TextView
        android:id="@+id/txtNamaUser2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="%username%"

        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
    />

    <TextView
        android:id="@+id/txtEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="%email%" />

</LinearLayout>
```

- menu\_main\_drawer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="all">
        <item
```

```

        android:id="@+id/nav_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="@string/menu_home" />
    <item
        android:id="@+id/nav_downloaded_articles"
        android:icon="@drawable/ic_baseline_library_books_24"
        android:title="@string/menu_downloaded_articles" />
    <item
        android:id="@+id/nav_saved_kajian"
        android:icon="@drawable/ic_baseline_bookmarks_24"
        android:title="@string/menu_saved_kajian" />
</group>
</menu>

```

- content\_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"

    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_main">

    <fragment
        android:id="@+id/nav_host_fragment"

        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

- mobile\_navigation.xml

```

<?xml version="1.0" encoding="utf-8"?>
<navigation
    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:id="@+id/mobile_navigation"
    app:startDestination="@+id/nav_home">

    <fragment
        android:id="@+id/nav_home"

        android:name="com.kajianid.android.fragment.HomeFragment"
        android:label="@string/menu_home"

```

```

        tools:layout="@layout/fragment_home" />

        <fragment
            android:id="@+id/nav_downloaded_articles"

            android:name="com.kajianid.android.fragment.DownloadedArticleFr
            agment"
            android:label="@string/menu_downloaded_articles"
            tools:layout="@layout/fragment_downloaded_article" />

        <fragment
            android:id="@+id/nav_saved_kajian"

            android:name="com.kajianid.android.fragment.SavedKajianFragment
            "
            android:label="@string/menu_saved_kajian"
            tools:layout="@layout/fragment_saved_kajian" />

    </navigation>

```

- menu\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/menuAbout"
        android:title="@string/about"
        android:icon="@drawable/ic_baseline_info_24"
        app:showAsAction="always" />
</menu>

```

b) Program

```

package com.kajianid.android.activities;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import com.kajianid.android.R;
import com.kajianid.android.databinding.ActivityMainBinding;

```

```

import static
androidx.navigation.ui.NavigationUI.setupActionBarWithNavController;
import static
androidx.navigation.ui.NavigationUI.setupWithNavController;

public class MainActivity extends AppCompatActivity {
    private AppBarConfiguration mAppBarConfiguration;
    private boolean doubleBackToExitPressedOnce = false;
    private DrawerLayout drawerLayout;
    private ActivityMainBinding binding;

    @Override
    public void onBackPressed() {
        if (doubleBackToExitPressedOnce) super.onBackPressed();

        doubleBackToExitPressedOnce = true;
        Toast.makeText(this, "Tekan sekali lagi untuk keluar dari
Kajian.ID!", Toast.LENGTH_SHORT).show();
        new Handler(Looper.getMainLooper()).postDelayed(() -> {
            doubleBackToExitPressedOnce = false;
        }, 2000);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        drawerLayout = binding.drawerLayout;

        setSupportActionBar(binding.appBarMain.toolbar);
        NavController navController =
Navigation.findNavController(this, R.id.nav_host_fragment);
        mAppBarConfiguration = new AppBarConfiguration.Builder(
            R.id.nav_home,
            R.id.nav_downloaded_articles,
            R.id.nav_saved_kajian)
            .setDrawerLayout(drawerLayout)
            .build();
        setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
        setupWithNavController(binding.navMain, navController);

        View headerView = binding.navMain.getHeaderView(0);
        TextView txtUsername =
headerView.findViewById(R.id.txtNamaUser2);
        TextView txtEmail =
headerView.findViewById(R.id.txtEmail);

        txtUsername.setText(Build.MODEL);
    }
}

```

```

        txtEmail.setText(Build.MANUFACTURER);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                drawerLayout.openDrawer(GravityCompat.START);
                break;

            case R.id.menuAbout:
                Intent i = new Intent(this, AboutActivity.class);
                startActivity(i);
        }
        return true;
    }

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

```

### c. Menu Beranda

#### a) Tampilan

- fragment\_home.xml

```

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <include
            android:id="@+id/content_fragment_home"
            layout="@layout/content_fragment_home" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <LinearLayout
                android:id="@+id/layout_menu"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="10dp"

```

```

        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <Button
                android:id="@+id/btnArticle"
                android:layout_width="0dp"
                android:layout_height="100dp"
                android:layout_gravity="start"
                android:layout_margin="5dp"
                android:layout_weight=".5"

                android:background="@drawable/button_round_rec"

                android:drawableTop="@drawable/ic_baseline_assignment_24"
                android:padding="15dp"
                android:shadowColor="#000"
                android:shadowRadius="1.0"
                android:text="@string/article"
                android:textAllCaps="false" />

            <Button
                android:id="@+id/btnKajian"
                android:layout_width="0dp"
                android:layout_height="100dp"
                android:layout_gravity="start"
                android:layout_margin="5dp"
                android:layout_weight=".5"

                android:background="@drawable/button_round_rec"

                android:drawableTop="@drawable/ic_baseline_ondemand_video_24"
                android:padding="15dp"
                android:shadowColor="#000"
                android:shadowRadius="1.0"
                android:text="@string/kajian"
                android:textAllCaps="false" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <Button
                android:id="@+id/btnMosqueLocation"
                android:layout_width="0dp"
                android:layout_height="100dp"
                android:layout_gravity="start"

```



```

        android:layout_margin="5dp"
        android:layout_weight=".5"

        android:background="@drawable/button_round_rec"

        android:drawableTop="@drawable/ic_baseline_location_city_36"
        android:padding="15dp"
        android:shadowColor="#000"
        android:shadowRadius="1.0"
        android:text="@string/mosque_location"
        android:textAllCaps="false" />

    </LinearLayout>

</LinearLayout>

</LinearLayout>

</LinearLayout>

</ScrollView>

```

- content\_fragment\_home.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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:tools="http://schemas.android.com/tools">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/atas"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:padding="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <de.hdodenhof.circleimageview.CircleImageView
            android:id="@+id/imgUserView"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:src="@mipmap/ic_launcher"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/textView11"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="12dp"

```

```

        android:text="Anda masuk pada perangkat"

app:layout_constraintStart_toEndOf="@+id/imgUserView"
    app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtNamaUser1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="12dp"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="4dp"
        android:text="%username%!"
        android:textSize="20sp"
        android:textColor="?android:textColorPrimary"
        android:textStyle="bold"

app:layout_constraintStart_toEndOf="@+id/imgUserView"
app:layout_constraintTop_toBottomOf="@+id/textView11" />

    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

## b) Program

```

package com.kajianid.android.fragment;

import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.kajianid.android.activities.ArticleIslamiActivity;
import com.kajianid.android.activities.KajianActivity;
import com.kajianid.android.databinding.FragmentHomeBinding;

public class HomeFragment extends Fragment {
    private FragmentHomeBinding binding;

    public HomeFragment() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

        @Override
        public View onCreateView(@NonNull LayoutInflater inflater,
                                   ViewGroup container,
                                   Bundle savedInstanceState) {

            binding = FragmentHomeBinding.inflate(inflater);
            return binding.getRoot();

        }

        @Override
        public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
            super.onViewCreated(view, savedInstanceState);
            binding.contentFragmentHome.txtNamaUser1.setText(
                Build.MANUFACTURER + " " + Build.MODEL
            );
            binding.btnArticle.setOnClickListener(it -> {
                Intent i = new Intent(getContext(),
ArticleIslamiActivity.class);
                startActivity(i);
            });
            binding.btnKajian.setOnClickListener(it ->{
                Intent i = new Intent(getContext(),
KajianActivity.class);
                startActivity(i);
            });
            binding.btnMosqueLocation.setOnClickListener(it -> {
                Uri gmmIntentUri = Uri.parse("geo:0,0?q=masjid
terdekat");
                Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                mapIntent.setPackage("com.google.android.apps.maps");
                startActivity(mapIntent);
            });
        }
    }
}

```

#### d. Menu Artikel

##### a) Tampilan

- activity\_article\_islami.xml

```

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

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary" />

    </com.google.android.material.appbar.AppBarLayout>

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:visibility="visible"/>

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/pullToRefresh"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/appBar">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recDataArtikel"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="8dp" />

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

</RelativeLayout>

```

- activity\_list\_article.xml

```

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

</androidx.constraintlayout.widget.ConstraintLayout>

```

- list\_article.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/layout_list_artikel"
    android:padding="8dp">

    <androidx.cardview.widget.CardView
        android:id="@+id/cv_item_note"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        app:cardElevation="4dp"
        app:cardMaxElevation="4dp"
        app:cardUseCompatPadding="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp">

            <ImageView
                android:id="@+id/I_Judul"
                android:layout_width="match_parent"
                android:layout_height="180dp"
                android:src="@drawable/foto"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <TextView
                android:id="@+id/TV_Judul"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@+id/I_Judul"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginTop="8dp"
                android:text="Judul"
                android:textSize="17sp"
                android:textStyle="bold"
                android:textColor="?android:textColorPrimary"
                app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/I_Judul" />

            <TextView
                android:id="@+id/TV_Ringkasan"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/TV_Judul"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:ellipsize="end"
                android:maxLines="2"
                android:text="Berisi ringkasan artikel"
                android:textSize="12sp"
                app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/TV_Judul" />

            <TextView

```

```

        android:id="@+id/TV_Tanggal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/TV_Ringkasan"
        android:layout_alignEnd="@+id/TV_Ringkasan"
        android:text="12 desember 2019"
        android:textAlignment="textEnd"
        android:textSize="12sp"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintTop_toBottomOf="@+id/TV_Ringkasan" />

        <TextView
            android:id="@+id/TV_Ustadz"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/TV_Tanggal"
            android:layout_alignEnd="@+id/TV_Ringkasan"
            android:layout_alignParentEnd="true"
            android:gravity="end"
            android:text="Oleh K.H.Abdul Somad"
            android:textSize="12sp"
            app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintTop_toBottomOf="@+id/TV_Tanggal" />

    </RelativeLayout>

</androidx.cardview.widget.CardView>

</RelativeLayout>

```

- menu\_article\_kajian.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/share"
        android:icon="@drawable/ic_baseline_share_24"
        android:title="@string/share"
        app:showAsAction="collapseActionView|ifRoom" />

</menu>

```

- activity\_read\_article.xml

```

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

```

```

<com.google.android.material.appbar.AppBarLayout
    android:id="@+id/appBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</com.google.android.material.appbar.AppBarLayout>

<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
    android:id="@+id/pullToRefresh"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/appBar">

    <ScrollView
        android:id="@+id/svArticle"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <include
            android:id="@+id/ContentReadArticle"
            layout="@layout/activity_content_read_article"
/>

    </ScrollView>

</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

<RelativeLayout
    android:id="@+id/progressMessage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/appBar"
    android:layout_centerInParent="true"
    android:background="@android:color/white"
    android:padding="30dp">

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/progressBar"
        android:layout_centerHorizontal="true"
        android:paddingTop="16dp"

```

```

        android:text="@string/load_data"
        android:textAlignment="center" />

</RelativeLayout>

<RelativeLayout
    android:id="@+id/errorMessage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/appBar"
    android:layout_centerInParent="true"
    android:background="@android:color/white"
    android:padding="30dp"
    android:visibility="gone">

    <ImageView
        android:id="@+id/imgWarning"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_centerHorizontal="true"
        android:contentDescription="@string/warning"
        android:src="@drawable/ic_baseline_warning_24" />

    <TextView
        android:id="@+id/tvErrorMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/imgWarning"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="@string/example_error_message"
        android:textAlignment="center" />

    <Button
        android:id="@+id/btnRefresh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/tvErrorMessage"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="@string/refresh" />

</RelativeLayout>

<com.google.android.material.floatingactionbutton.FloatingAction
nButton
    android:id="@+id/fabDownload"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_margin="24dp"

```



```

        android:tint="?android:textColorPrimary"
        android:foregroundGravity="bottom"
        app:fabSize="normal"

app:srcCompat="@drawable/ic_baseline_assignment_returned_24" />

</RelativeLayout>

```

- activity\_content\_read\_article.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="wrap_content"
android:padding="16dp">

    <TextView
        android:id="@+id/tvArticleTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/example_article_title"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imgArticle"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:src="@drawable/icon"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/tv_nama_Ustad" />

    <ImageView
        android:id="@+id/imgLike"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:contentDescription="@string/likes"
        android:src="@drawable/ic_baseline_thumb_up_24"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imgArticle"
        android:visibility="gone"/>

    <TextView
        android:id="@+id/tvLikeCount"
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:text="@string/example_likes"
        app:layout_constraintBottom_toBottomOf="@+id/imgLike"
        app:layout_constraintStart_toEndOf="@+id/imgLike"
        app:layout_constraintTop_toTopOf="@+id/imgLike"
        android:visibility="gone"/>

<ImageView
    android:id="@+id/imgCalendar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="0dp"
    android:contentDescription="@string/post_date"
    android:src="@drawable/ic_baseline_calendar_today_24"

app:layout_constraintBottom_toBottomOf="@+id/tvLikeCount"
    app:layout_constraintStart_toEndOf="@+id/tvLikeCount"
    app:layout_constraintTop_toTopOf="@+id/tvLikeCount" />

<TextView
    android:id="@+id/tvPostDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="@string/example_post_date"

app:layout_constraintBottom_toBottomOf="@+id/imgCalendar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/imgCalendar"
    app:layout_constraintTop_toTopOf="@+id/imgCalendar" />

<TextView
    android:id="@+id/tv_nama_Ustad"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingBottom="16dp"
    android:gravity="start"
    android:text="@string/nama_ustadz"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/tvArticleTitle" />

<TextView
    android:id="@+id/tvPostContent"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:autoLink="all"
    android:paddingTop="16dp"
    android:paddingBottom="84dp"
    android:text="@string/example_text"
    app:layout_constraintEnd_toEndOf="parent"

```

```

        app:layout_constraintHorizontal_bias="0.812"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imgCalendar"
    />

</androidx.constraintlayout.widget.ConstraintLayout>

```

- fragment\_downloaded\_article.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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=".fragment.DownloadedArticleFragment">

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

        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        tools:listitem="@layout/list_article"/>

</RelativeLayout>

```

## b) Program

- Article.java

```

package com.kajianid.android.data;

import android.os.Parcel;
import android.os.Parcelable;

public class Article implements Parcelable {
    protected Article(Parcel in) {
        id = in.readString();
        title = in.readString();
        content = in.readString();
        postDate = in.readString();
        ustadzName = in.readString();
        hasImg = in.readString();
        imgUrl = in.readString();
        likes = in.readInt();
    }

    public Article() {
    }
}

```

```

        public static final Creator<Article> CREATOR = new
Creator<Article>() {
            @Override
            public Article createFromParcel(Parcel in) {
                return new Article(in);
            }

            @Override
            public Article[] newArray(int size) {
                return new Article[size];
            }
        };

        public String getId() {
            return id;
        }

        public void setId(String id) {
            this.id = id;
        }

        public String getTitle() {
            return title;
        }

        public void setTitle(String title) {
            this.title = title;
        }

        public String getContent() {
            return content;
        }

        public void setContent(String content) {
            this.content = content;
        }

        public String getPostDate() {
            return postDate;
        }

        public void setPostDate(String postDate) {
            this.postDate = postDate;
        }

        public String getUstadzName() {
            return ustadzName;
        }

        public void setUstadzName(String ustadzName) {
            this.ustadzName = ustadzName;
        }

```

```

    public String getHasImg() {
        return hasImg;
    }

    public void setHasImg(String hasImg) {
        this.hasImg = hasImg;
    }

    public String getImgUrl() {
        return imgUrl;
    }

    public void setImgUrl(String imgUrl) {
        this.imgUrl = imgUrl;
    }

    public int getLikes() {
        return likes;
    }

    public void setLikes(int likes) {
        this.likes = likes;
    }

    private String id;
    private String title;
    private String content;
    private String postDate;
    private String ustadzName;
    private String hasImg;
    private String imgUrl;
    private int likes;

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(id);
        dest.writeString(title);
        dest.writeString(content);
        dest.writeString(postDate);
        dest.writeString(ustadzName);
        dest.writeString(hasImg);
        dest.writeString(imgUrl);
        dest.writeInt(likes);
    }
}

```

- ArticleIslamiActivity.kt

```

package com.kajianid.android.activities

import android.os.Bundle
import android.view.MenuItem
import android.view.View
import android.widget.SearchView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.Observer
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.android.R
import com.kajianid.android.adapter.ListArticleAdapter
import
com.kajianid.android.databinding.ActivityArticleIslamiBinding
import com.kajianid.android.viewmodels.ArticleViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import org.json.JSONObject

class ArticleIslamiActivity : AppCompatActivity() {
    var id = ""
    private lateinit var articleViewModel: ArticleViewModel
    private var listArticleAdapter: ListArticleAdapter =
ListArticleAdapter()
    private var searchView: SearchView? = null
    private lateinit var binding: ActivityArticleIslamiBinding
    override fun onOptionsItemSelected(item: MenuItem): Boolean
{
        if (item.itemId == android.R.id.home) {
            super.onBackPressed()
        }
        return super.onOptionsItemSelected(item)
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityArticleIslamiBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.progressBar.visibility = View.VISIBLE

        // fungsi Toolbar dan ActionBar(back pd artikel)
        setSupportActionBar(binding.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)

supportActionBar?.setTitle(resources.getString(R.string.read_ar
ticle))

        binding.recDataArtikel.layoutManager =
LinearLayoutManager(this)

```

```

        binding.recDataArtikel.adapter = listArticleAdapter
        articleViewModel = ViewModelProvider(this,
        ViewModelProvider.NewInstanceFactory())
            .get(ArticleViewModel::class.java)
        articleViewModel.article.observe(this, Observer {
            if (it != null) {
                binding.progressBar.visibility = View.GONE
                listArticleAdapter.setArticle(it)
            }
        })
        articleViewModel.setArticleAsync(this,
        listArticleAdapter, "")

        binding.pullToRefresh.setOnRefreshListener {
            GlobalScope.launch (Dispatchers.Main) {
                val deferredArticles = async(Dispatchers.IO) {
                    articleViewModel.setArticle(this@ArticleIslamiActivity, "")
                }
                val status = deferredArticles.await()
                if (status != null) {
                    val parse = JSONObject(status)
                    Toast.makeText(this@ArticleIslamiActivity,
                    parse.getString("message"), Toast.LENGTH_SHORT).show()
                } else {
                    listArticleAdapter.notifyDataSetChanged()
                }
                binding.pullToRefresh.isRefreshing = false
            }
        }
    }
}

```

- ArticleSQLAdapter.java

```

package com.kajianid.android.adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.kajianid.android.activities.ReadArticleSQLActivity;
import com.kajianid.android.data.Article;
import com.kajianid.android.databinding.ListArticleBinding;
import java.util.ArrayList;

public class ArticleSQLAdapter extends
RecyclerView.Adapter<ArticleSQLAdapter.ArticleSQLViewHolder> {
    private final Context context;
    ArrayList<Article> mArticle = new ArrayList<>();
}

```

```

public ArticleSQLAdapter(Context context) {
    this.context = context;
}

public void setArticle(ArrayList<Article> mArticle) {
    if (mArticle != null) this.mArticle.clear();
    assert mArticle != null;
    this.mArticle.addAll(mArticle);
    notifyDataSetChanged();
}

@NonNull
@Override
public ArticleSQLAdapter.ArticleSQLViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    ListArticleBinding binding =
ListArticleBinding.inflate(LayoutInflater.from(parent.getContext()
t()));
    return new
ArticleSQLAdapter.ArticleSQLViewHolder(binding);
}

@Override
public void onBindViewHolder(@NonNull
ArticleSQLAdapter.ArticleSQLViewHolder holder, int position) {
    holder.bind(mArticle.get(position));
}

@Override
public int getItemCount() {
    return mArticle == null ? 0 : mArticle.size();
}

static class ArticleSQLViewHolder extends
RecyclerView.ViewHolder {

    private final ListArticleBinding binding;

    public ArticleSQLViewHolder(@NonNull ListArticleBinding
itemView) {
        super(itemView.getRoot());
        this.binding = itemView;
    }

    public void bind(Article article) {
        binding.TVJudul.setText(article.getTitle());
        binding.TVRingkasan.setText(article.getContent());
        binding.TVTanggal.setText(article.getPostDate());
        binding.TVUstadz.setText(article.getUstadzName());

        if (article.getHasImg().equals("0") ||
article.getHasImg().equals("false")) {
            binding.IJudul.setVisibility(View.GONE);

```



```

        } else {
            Glide.with(binding.getRoot().getContext())
                .asBitmap()
                .load(article.getImgUrl())
                .into(binding.IJudul);
        }

        binding.getRoot().setOnClickListener(view -> {
            Intent artikelIntent = new
            Intent(binding.getRoot().getContext(),
            ReadArticleSQLActivity.class);

            artikelIntent.putExtra(ReadArticleSQLActivity.EXTRA_PARCEL_ARTI
            CLES, article);

            binding.getRoot().getContext().startActivity(artikelIntent);
        });
    }
}

```

- ListSQLAdapter.java

```

package com.kajianid.android.adapter;

import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.kajianid.android.activities.ReadArticleActivity;
import com.kajianid.android.data.Article;
import com.kajianid.android.databinding.ListArticleBinding;
import java.util.ArrayList;

public class ListArticleAdapter extends
RecyclerView.Adapter<ListArticleAdapter.Myarticle> {
    ArrayList<Article> mArticle = new ArrayList<>();

    public ListArticleAdapter() {
    }

    public void setArticle(ArrayList<Article> mArticle) {
        if (mArticle != null) this.mArticle.clear();
        assert mArticle != null;
        this.mArticle.addAll(mArticle);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public Myarticle onCreateViewHolder(@NonNull ViewGroup parent,

```

```

int viewType) {
    ListArticleBinding binding =
ListArticleBinding.inflate(LayoutInflater.from(parent.getContext()));
    return new Myarticle(binding);
}

@Override
public void onBindViewHolder(@NonNull final Myarticle binding,
final int position) {
    binding.bind(mArticle.get(position));
}

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

static class Myarticle extends RecyclerView.ViewHolder {

    private final ListArticleBinding binding;

    public Myarticle(@NonNull ListArticleBinding itemView) {
        super(itemView.getRoot());
        this.binding = itemView;
    }

    public void bind(Article article) {
        binding.TVJudul.setText(article.getTitle());
        binding.TVRingkasan.setText(article.getContent());
        binding.TVTanggal.setText(article.getPostDate());
        binding.TVUstadz.setText(article.getUstadzName());

        if (article.getHasImg().equals("0")) {
            binding.IJudul.setVisibility(View.GONE);
        } else {
            Glide.with(binding.getRoot().getContext())
                .asBitmap()
                .load(article.getImgUrl())
                .into(binding.IJudul);
        }

        binding.getRoot().setOnClickListener(view -> {
            Intent artikelIntent = new
Intent(binding.getRoot().getContext(), ReadArticleActivity.class);

artikelIntent.putExtra(ReadArticleActivity.EXTRA_ARTICLE_ID,
article.getId());

binding.getRoot().getContext().startActivity(artikelIntent);
        });
    }
}

```

```

    }
}

```

## - DbArticleHelper.kt

```

package com.kajianid.android.databases.article

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteException
import
com.kajianid.android.databases.DatabaseContract.ArticleColumns.C
ompanion.ID
import
com.kajianid.android.databases.DatabaseContract.ArticleColumns.C
ompanion.TABLE_NAME_ARTICLES
import com.kajianid.android.databases.DatabaseHelper

class DbArticleHelper (context: Context){
    companion object {
        private const val DATABASE_TABLE = TABLE_NAME_ARTICLES
        private lateinit var databaseHelper: DatabaseHelper
        private var INSTANCE: DbArticleHelper? = null
        private lateinit var database: SQLiteDatabase

        fun getInstance(context: Context): DbArticleHelper =
INSTANCE ?: synchronized(this) {
    INSTANCE ?: DbArticleHelper(context)
}

    }

    init{
        databaseHelper = DatabaseHelper(context)
    }

    @Throws(SQLiteException::class)
    fun open(){
        database = databaseHelper.writableDatabase
    }

    fun close(){
        databaseHelper.close()
        if (database.isOpen) database.close()
    }

    fun queryAll(): Cursor {
        return database.query(TABLE_NAME_ARTICLES,
            null,
            null,
            null,
            null,
            null,
            null,

```

```

        null)
    }

    fun queryById(id: String): Cursor {
        return database.query(
            TABLE_NAME_ARTICLES,
            null,
            "$ID = ?",
            arrayOf(id),
            null,
            null,
            null)
    }

    fun insert(values: ContentValues?): Long {
        return database.insert(DATABASE_TABLE, null, values)
    }

    fun update(id: String, values: ContentValues?): Int {
        return database.update(DATABASE_TABLE, values, "$ID = ?", arrayOf(id))
    }

    fun deleteById(id: String): Int {
        return database.delete(DATABASE_TABLE, "$ID = $id", null)
    }
}

```

#### - MappingHelper.kt

```

package com.kajianid.android.databases.article

import android.database.Cursor
import com.kajianid.android.data.Article
import com.kajianid.android.databases.DatabaseContract

object MappingHelper {
    fun mapCursorToArrayList(articleCursor: Cursor?):
    ArrayList<Article>{
        var articles = ArrayList<Article>()
        articleCursor?.apply {
            while (moveToNext()) {
                val id =
                getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
                ID))

                val title =
                getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
                TITLE))

                val post_date =
                getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
                POST_DATE))

                val content =
                getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
                CONTENT))

                val has_img =

```

```

getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
HAS_IMG))

        val ustadz_name =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
USTADZ_NAME))

        val article = Article()
        article.id = id
        article.title = title
        article.postDate = post_date
        article.content = content
        article.hasImg = has_img
        article.ustadzName = ustadz_name
        articles.add(article)
    }
}

return articles
}

fun mapCursorToObject(articleCursor: Cursor?): Article{
    var article = Article()
    articleCursor?.apply {
        moveToFirst()
        val id =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
ID))

        val title =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
TITLE))

        val post_date =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
POST_DATE))

        val content =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
CONTENT))

        val has_img =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
HAS_IMG))

        val ustadz_name =
getString(getColumnIndexOrThrow(DatabaseContract.ArticleColumns.
USTADZ_NAME))

        article.id = id
        article.title = title
        article.postDate = post_date
        article.content = content
        article.hasImg = has_img
        article.ustadzName = ustadz_name
    }
    return article
}

```

```
}
```

#### - DatabaseContract.kt

```
package com.kajianid.android.databases

import android.provider.BaseColumns

object DatabaseContract {
    class KajianColumns : BaseColumns {
        companion object {
            const val TABLE_NAME_KAJIAN = "kajian"
            val ID = "id"
            val KAJIAN_TITLE = "kajian_title"
            val USTADZ_NAME = "ustadz_name"
            val MOSQUE_NAME = "mosque_name"
            val ADDRESS = "address"
            val PLACE = "place"
            val YOUTUBE_LINK = "youtube_link"
            val DESCRIPTION = "description"
            val IMG_RESOURCE = "img_resource"
            val DATE_ANNOUNCE = "date_announce"
            val DATE_DUE = "date_due"
        }
    }

    class ArticleColumns : BaseColumns {
        companion object {
            const val TABLE_NAME_ARTICLES = "articles"
            val ID = "id"
            val TITLE = "title"
            val POST_DATE = "post_data"
            val CONTENT = "content"
            val HAS_IMG = "has_img"
            val USTADZ_NAME = "ustadz_name"
            val IMGURL = "img_url"
        }
    }
}
```

#### - DatabaseHelper.kt

```
package com.kajianid.android.databases

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import com.kajianid.android.databases.DatabaseContract.ArticleColumns.C
import companion.TABLE_NAME_ARTICLES
import com.kajianid.android.databases.DatabaseContract.KajianColumns.C
import companion.TABLE_NAME_KAJIAN
```

```

class DatabaseHelper (context: Context):
    SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION){
        companion object {
            private const val DATABASE_NAME = "kajianid"
            private const val DATABASE_VERSION = 1
            private val SQL_CREATE_NAME_KAJIAN = """
                CREATE TABLE $TABLE_NAME_KAJIAN(
                    ${DatabaseContract.KajianColumns.ID} varchar(5) not
null primary key,
                    ${DatabaseContract.KajianColumns.KAJIAN_TITLE}
Text,
                    ${DatabaseContract.KajianColumns.USTADZ_NAME}
varchar(15),
                    ${DatabaseContract.KajianColumns.MOSQUE_NAME}
varchar(30),
                    ${DatabaseContract.KajianColumns.ADDRESS} Text,
                    ${DatabaseContract.KajianColumns.PLACE}
varchar(30),
                    ${DatabaseContract.KajianColumns.YOUTUBE_LINK}
Text,
                    ${DatabaseContract.KajianColumns.DESCRPTION} Text,
                    ${DatabaseContract.KajianColumns.IMG_RESOURCE}
Text,
                    ${DatabaseContract.KajianColumns.DATE_ANNOUNCE}
Text,
                    ${DatabaseContract.KajianColumns.DATE_DUE} Text)
                """.trimIndent()

            private val SQL_CREATE_TABLE_ARTIKEL = """
                CREATE TABLE $TABLE_NAME_ARTICLES(
                    ${DatabaseContract.ArticleColumns.ID} varchar(5) not
null primary key,
                    ${DatabaseContract.ArticleColumns.TITLE} text,
                    ${DatabaseContract.ArticleColumns.POST_DATE} Text,
                    ${DatabaseContract.ArticleColumns.CONTENT} Text,
                    ${DatabaseContract.ArticleColumns.HAS_IMG} Int,
                    ${DatabaseContract.ArticleColumns.USTADZ_NAME}
varchar(15),
                    ${DatabaseContract.ArticleColumns.IMGURL} Text)
                """.trimIndent()
        }

        override fun onCreate(db: SQLiteDatabase?) {
            db?.execSQL(SQL_CREATE_NAME_KAJIAN)
            db?.execSQL(SQL_CREATE_TABLE_ARTIKEL)
        }

        override fun onUpgrade(db: SQLiteDatabase?, oldVersion:
Int, newVersion: Int) {
            onCreate(db)
        }
    }

```

```
}
```

- ArticleViewModel.java

```
package com.kajianid.android.viewmodels;

import android.content.Context;
import android.os.Looper;
import android.util.Log;
import android.widget.Toast;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;
import com.kajianid.android.adapter.ListArticleAdapter;
import com.kajianid.android.R;
import com.kajianid.android.data.Article;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Locale;
import cz.msebera.android.httpclient.Header;

public class ArticleViewModel extends ViewModel {

    private MutableLiveData<ArrayList<Article>> listArticle =
new MutableLiveData<>();

    public String setArticle(Context context, String
searchQuery) {
        try {
            Looper.prepare();
        } catch (Exception e) {
        }
        final String[] ret = {null};
        String url =
context.getResources().getString(R.string.server) +
"api/articles";
        ArrayList<Article> listItems = new ArrayList<>();
        SyncHttpClient client = new SyncHttpClient();

        // Request Parameters
        RequestParams params = new RequestParams();
        params.put("read", "0");
        params.put("ustadz_mode", "0");
        params.put("query", searchQuery);
```



```

        client.get(url, params, new AsyncHttpResponseHandler()
{
    @Override
    public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
        try {
            String result = new String(responseBody);
            JSONObject responseObject = new
JSONObject(result);
            JSONArray list =
responseObject.getJSONArray("data");

            Log.d("Isi", result);

            for (int i = 0; i < list.length(); i++) {
                JSONObject articleJSONObject =
list.getJSONObject(i);
                Article article = new Article();

                article.setId(articleJSONObject.getString("id"));

                article.setTitle(articleJSONObject.getString("title"));

                article.setContent(articleJSONObject.getString("content"));

                article.setUstadzName(articleJSONObject.getString("ustadz_name"
));

                article.setHasImg(articleJSONObject.getString("has_img"));

                if (article.getHasImg().equals("1")) {
                    String extension =
articleJSONObject.getString("extension");

                    article.setImgUrl(context.getResources().getString(R.string.ser
ver) + "assets/articles/" + article.getId() + "." + extension);
                }

                String date =
articleJSONObject.getString("post_date");
                Date dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault()).parse(date);
                String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue);

                article.setPostDate(dateDueFormatted);
                listItems.add(article);
            }
            listArticle.postValue(listItems);
            ret[0] = null;

```

```

        } catch (JSONException | ParseException e) {
            e.printStackTrace();
        }

    }

    @Override
    public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
        error.printStackTrace();
        String expected = new String(responseBody);
        ret[0] = expected;
    }
});
return ret[0];
}

    public boolean setArticleAsync(Context context,
ListArticleAdapter adapter, String searchQuery) {
        final boolean[] ret = {false};
        String url =
context.getResources().getString(R.string.server) +
"api/articles";
        ArrayList<Article> listItems = new ArrayList<>();
        AsyncHttpClient client = new AsyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("read", "0");
        params.put("ustadz_mode", "0");
        params.put("query", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler()
{
            @Override
            public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new
JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");

                    Log.d("Isi", result);

                    for (int i = 0; i < list.length(); i++) {
                        JSONObject articleJSONObject =
list.getJSONObject(i);
                        Article article = new Article();
                        article.setId(articleJSONObject.getString("id"));

```

```

article.setTitle(articleJSONObject.getString("title"));

article.setContent(articleJSONObject.getString("content"));

article.setUstadzName(articleJSONObject.getString("ustadz_name")
));

article.setHasImg(articleJSONObject.getString("has_img"));

        if (article.getHasImg().equals("1")) {
            String extension =
articleJSONObject.getString("extension");

article.setImgUrl(context.getResources().getString(R.string.ser
ver) + "assets/articles/" + article.getId() + "." + extension);
        }

        String date =
articleJSONObject.getString("post_date");
        Date dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault()).parse(date);
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue);

        article.setPostDate(dateDueFormatted);
        listItems.add(article);
        adapter.notifyDataSetChanged();
    }
    listArticle.postValue(listItems);
    ret[0] = true;
} catch (JSONException | ParseException e) {
    e.printStackTrace();
}

}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {

Toast.makeText(context, "Failure!\n"+error.getMessage()+"\n"+res
ponseBody+"\n"+url, Toast.LENGTH_SHORT).show();
    error.printStackTrace();
    ret[0] = false;
}

});
return ret[0];
}

public LiveData<ArrayList<Article>> getArticle() {
    return listArticle;
}

```

```

    }
}

```

## - ReadArticleActivity.kt

```

package com.kajianid.android.activities

import android.content.ContentValues
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ShareCompat
import androidx.lifecycle.ViewModelProvider
import com.bumptech.glide.Glide
import com.kajianid.android.databases.DatabaseContract
import com.kajianid.android.databases.article.DbArticleHelper
import com.kajianid.android.databases.article.MappingHelper
import com.kajianid.android.R
import com.kajianid.android.data.Article
import
com.kajianid.android.databinding.ActivityReadArticleBinding
import com.kajianid.android.viewmodels.ReadArticleViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import java.util.*

class ReadArticleActivity : AppCompatActivity() {
    private lateinit var readArticleViewModel :
ReadArticleViewModel
    private var id: String = ""
    private var title: String = ""
    private var content: String = ""
    private var isError: Boolean = false
    private var downloaded = false
    private lateinit var dbArticleHelper: DbArticleHelper
    private lateinit var article: Article
    private lateinit var binding: ActivityReadArticleBinding

    companion object {
        const val EXTRA_ARTICLE_ID = "extra_article_id"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityReadArticleBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.progressBar.visibility = View.VISIBLE
    }
}

```

```

        val bundle = intent.extras
        id = bundle?.get(EXTRA_ARTICLE_ID).toString()

        setSupportActionBar(binding.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)

supportActionBar?.setTitle(resources.getString(R.string.read_article))

        readArticleViewModel = ViewModelProvider(this,
ViewModelProvider.NewInstanceFactory())
            .get(ReadArticleViewModel::class.java)
        readArticleViewModel.article.observe(this, {
            if (it["status"] == true) {
                title = it["title"].toString()
                content = it["content"].toString()
                binding.ContentReadArticle.tvArticleTitle.text
= it["title"].toString()
                if (it["hasImg"] == true) {
                    Glide.with(this)
                        .load(
resources.getString(R.string.server) +
                            "assets/articles/" + id +
                            "." +
Objects.requireNonNull(it["extension"]).toString()
                        )
                    .into(binding.ContentReadArticle.imgArticle)
                } else {
binding.ContentReadArticle.imgArticle.visibility = View.GONE
                }
                val likes = it["like"].toString() + "
${resources.getString(R.string.likes)}"
                binding.ContentReadArticle.tvLikeCount.text =
likes
                binding.ContentReadArticle.tvPostDate.text =
it["post_date"].toString()
                binding.ContentReadArticle.tvPostContent.text =
it["content"].toString()
                val ustadzName =
resources.getString(R.string.by) + " " +
it["ustadz_name"].toString()
                binding.ContentReadArticle.tvNamaUstad.text =
ustadzName
                binding.progressMessage.visibility = View.GONE
                binding.errorMessage.visibility = View.GONE

                article = Article()
                article.id = it["id"].toString()

```

```

        article.title = it["title"].toString()
        article.content = it["content"].toString()
        article.postDate= it["post_date"].toString()
        article.ustadzName =
it["ustadz_name"].toString()
        article.hasImg = it["hasImg"].toString()
        article.imgUrl =
resources.getString(R.string.server) +
"assets/articles/${id}.${it["extension"]}"
        article.likes = it["like"].toString().toInt()
    } else {
        binding.progressMessage.visibility = View.GONE
        binding.errorMessage.visibility = View.VISIBLE
        val errorMessage = ""
        Error!
        [${it["code"]}] : ${it["message"]}
        """.trimIndent()
        binding.tvErrorMessage.text = errorMessage
    }
})

readArticleViewModel.setArticleAsync(this, id)
dbArticleHelper= DbArticleHelper.getInstance(this)
dbArticleHelper.open()

GlobalScope.launch(Dispatchers.Main) {
    val deferredArticle = async(Dispatchers.IO) {
        val cursor = dbArticleHelper.queryById(id)
        MappingHelper.mapCursorToArrayList(cursor)
    }
    val isDownload = deferredArticle.await()
    if(isDownload.size==0){
        downloaded= false

binding.fabDownload.setImageResource(R.drawable.ic_baseline_ass
ignment_returned_24)
    }else{
        downloaded= true

binding.fabDownload.setImageResource(R.drawable.ic_baseline_ass
ignment_turned_in_24)
    }
    binding.fabDownload.setOnClickListener {
        if(downloaded){
            val alert = AlertDialog.Builder(this)

alert.setTitle(resources.getString(R.string.sure))

alert.setMessage(resources.getString(R.string.remove_article_co
nfirm))

alert.setPositiveButton(resources.getString(R.string.yes)) { _,

```

```

_ ->
        dbArticleHelper.deleteById(id)
        downloaded = false

binding.fabDownload.setImageResource(R.drawable.ic_baseline_assignment_returned_24)
        Toast.makeText(this, "Artikel telah dihapus!", Toast.LENGTH_LONG).show()
    }

alert.setNegativeButton(resources.getString(R.string.no)) { _,
_ ->
        /* no-op */
    }
    alert.create().show()
} else {
    val values= ContentValues()
    values.put(DatabaseContract.ArticleColumns.ID,
article.id)

values.put(DatabaseContract.ArticleColumns.TITLE, article.title)

values.put(DatabaseContract.ArticleColumns.POST_DATE,
article.postDate)

values.put(DatabaseContract.ArticleColumns.CONTENT,
article.content)

values.put(DatabaseContract.ArticleColumns.HAS_IMG,
article.hasImg)

values.put(DatabaseContract.ArticleColumns.USTADZ_NAME,
article.ustadzName)

values.put(DatabaseContract.ArticleColumns.IMGURL,
article.imgUrl)

        dbArticleHelper.insert(values)
        downloaded= true

binding.fabDownload.setImageResource(R.drawable.ic_baseline_assignment_turned_in_24)
        Toast.makeText(this, "Artikel Berhasil Diunduh", Toast.LENGTH_LONG).show()
    }
}

// pullToRefresh
binding.pullToRefresh.setOnRefreshListener {
    binding.svArticle.visibility = View.GONE
    binding.pullToRefresh.isRefreshing = false
    binding.errorMessage.visibility = View.GONE
    binding.progressMessage.visibility = View.VISIBLE
    GlobalScope.launch(Dispatchers.Main) {
        val deferredArticles = async(Dispatchers.IO) {

```

```

        id.let {
readArticleViewModel.setArticle(applicationContext, it) }
        }
        val it = deferredArticles.await()
        if (it["status"] == true) {
            title = it["title"].toString()
            content = it["content"].toString()

binding.ContentReadArticle.tvArticleTitle.text =
it["title"].toString()
            if (it["hasImg"] == true) {
                Glide.with(applicationContext)

.load(resources.getString(R.string.server) +
"assets/articles/${id}.${it["extension"]}")

.into(binding.ContentReadArticle.imgArticle)
            } else {

binding.ContentReadArticle.imgArticle.visibility = View.GONE
            }
            val likes = it["like"].toString() + "
${resources.getString(R.string.likes)}"
            binding.ContentReadArticle.tvLikeCount.text
= likes
            binding.ContentReadArticle.tvPostDate.text
= it["post_date"].toString()

binding.ContentReadArticle.tvPostContent.text =
it["content"].toString()
            binding.progressMessage.visibility =
View.GONE
            binding.svArticle.visibility = View.VISIBLE
        } else {
            binding.progressMessage.visibility =
View.GONE
            binding.errorMessage.visibility =
View.VISIBLE
            val errorMessage = ""
                Error!
                [${it["code"]}] : ${it["message"]}
                """.trimIndent()
            binding.tvErrorMessage.text = errorMessage
        }
    }
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_article_kajian, menu)
    return true
}

```



```

        override fun onOptionsItemSelected(item: MenuItem): Boolean
        {
            when (item.itemId) {
                android.R.id.home -> super.onBackPressed()
                R.id.share -> {
                    val mimeType = "text/plain"
                    ShareCompat.IntentBuilder.from(this).apply {
                        setType(mimeType)
                        setChooserTitle("Bagikan artikel ini
sekarang!")
                        setText("
*${binding.ContentReadArticle.tvArticleTitle.text}*
_${binding.ContentReadArticle.tvNamaUstad.text} pada
${binding.ContentReadArticle.tvPostDate.text}_
${binding.ContentReadArticle.tvPostContent.text}

Informasi ini disebarakan melalui aplikasi
${getString(R.string.app_name)}.
        """.trimIndent())
                        startChooser()
                    }
                }
            }
            return super.onOptionsItemSelected(item)
        }
    }
}

```

#### - ReadArticleSQLActivity.kt

```

package com.kajianid.android.activities

import android.content.ContentValues
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ShareCompat
import androidx.lifecycle.ViewModelProvider
import com.bumptech.glide.Glide
import com.kajianid.android.databases.DatabaseContract
import com.kajianid.android.databases.article.DbArticleHelper
import com.kajianid.android.databases.article.MappingHelper
import com.kajianid.android.R
import com.kajianid.android.data.Article
import
com.kajianid.android.databinding.ActivityReadArticleBinding
import com.kajianid.android.viewmodels.ReadArticleViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async

```

```

import kotlinx.coroutines.launch
import java.util.*

class ReadArticleActivity : AppCompatActivity() {
    private lateinit var readArticleViewModel :
ReadArticleViewModel
    private var id: String = ""
    private var title: String = ""
    private var content: String = ""
    private var isError: Boolean = false
    private var downloaded = false
    private lateinit var dbArticleHelper: DbArticleHelper
    private lateinit var article: Article
    private lateinit var binding: ActivityReadArticleBinding

    companion object {
        const val EXTRA_ARTICLE_ID = "extra_article_id"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityReadArticleBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.progressBar.visibility = View.VISIBLE

        val bundle = intent.extras
        id = bundle?.get(EXTRA_ARTICLE_ID).toString()

        setSupportActionBar(binding.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)

supportActionBar?.setTitle(resources.getString(R.string.read_ar
ticle))

        readArticleViewModel = ViewModelProvider(this,
ViewModelProvider.NewInstanceFactory())
            .get(ReadArticleViewModel::class.java)
        readArticleViewModel.article.observe(this, {
            if (it["status"] == true) {
                title = it["title"].toString()
                content = it["content"].toString()
                binding.ContentReadArticle.tvArticleTitle.text
= it["title"].toString()
                if (it["hasImg"] == true) {
                    Glide.with(this)
                        .load(
resources.getString(R.string.server) +
                            "assets/articles/" + id +
                            "." +
Objects.requireNonNull(it["extension"]).toString()

```

```

        )

        .into(binding.ContentReadArticle.imgArticle)
        } else {

binding.ContentReadArticle.imgArticle.visibility = View.GONE
        }
        val likes = it["like"].toString() + "
${resources.getString(R.string.likes)}"
        binding.ContentReadArticle.tvLikeCount.text =
likes
        binding.ContentReadArticle.tvPostDate.text =
it["post_date"].toString()
        binding.ContentReadArticle.tvPostContent.text =
it["content"].toString()
        val ustadzName =
resources.getString(R.string.by) + " " +
it["ustadz_name"].toString()
        binding.ContentReadArticle.tvNamaUstad.text =
ustadzName

        binding.progressMessage.visibility = View.GONE
        binding.errorMessage.visibility = View.GONE

        article = Article()
        article.id = it["id"].toString()
        article.title = it["title"].toString()
        article.content = it["content"].toString()
        article.postDate= it["post_date"].toString()
        article.ustadzName =
it["ustadz_name"].toString()
        article.hasImg = it["hasImg"].toString()
        article.imgUrl =
resources.getString(R.string.server) +
"assets/articles/${id}.${it["extension"]}"
        article.likes = it["like"].toString().toInt()
    } else {
        binding.progressMessage.visibility = View.GONE
        binding.errorMessage.visibility = View.VISIBLE
        val errorMessage = ""
        Error!
        [${it["code"]}] : ${it["message"]}
        """.trimIndent()
        binding.tvErrorMessage.text = errorMessage
    }
})

readArticleViewModel.setArticleAsync(this, id)
dbArticleHelper= DbArticleHelper.getInstance(this)
dbArticleHelper.open()

GlobalScope.launch(Dispatchers.Main) {
    val deferredArticle = async(Dispatchers.IO) {

```

```

        val cursor = dbArticleHelper.queryById(id)
        MappingHelper.mapCursorToArrayList(cursor)
    }
    val isDownload = deferredArticle.await()
    if(isDownload.size==0){
        downloaded= false

binding.fabDownload.setImageResource(R.drawable.ic_baseline_assignment_returned_24)
    }else{
        downloaded= true

binding.fabDownload.setImageResource(R.drawable.ic_baseline_assignment_turned_in_24)
    }
    }
    binding.fabDownload.setOnClickListener {
        if(downloaded){
            val alert = AlertDialog.Builder(this)

alert.setTitle(resources.getString(R.string.sure))

alert.setMessage(resources.getString(R.string.remove_article_confirm))

alert.setPositiveButton(resources.getString(R.string.yes)) { _, _ ->

                dbArticleHelper.deleteById(id)
                downloaded = false

binding.fabDownload.setImageResource(R.drawable.ic_baseline_assignment_returned_24)
                Toast.makeText(this, "Artikel telah dihapus!", Toast.LENGTH_LONG).show()
            }

alert.setNegativeButton(resources.getString(R.string.no)) { _, _ ->

                /* no-op */
            }
            alert.create().show()
        }else{
            val values= ContentValues()
            values.put(DatabaseContract.ArticleColumns.ID,
article.id)

values.put(DatabaseContract.ArticleColumns.TITLE, article.title)

values.put(DatabaseContract.ArticleColumns.POST_DATE,
article.postDate)

values.put(DatabaseContract.ArticleColumns.CONTENT,
article.content)

```

```

values.put(DatabaseContract.ArticleColumns.HAS_IMG,
article.hasImg)

values.put(DatabaseContract.ArticleColumns.USTADZ_NAME,
article.ustadzName)

values.put(DatabaseContract.ArticleColumns.IMGURL,
article.imgUrl)
        dbArticleHelper.insert(values)
        downloaded= true

binding.fabDownload.setImageResource(R.drawable.ic_baseline_ass
ignment_turned_in_24)
        Toast.makeText(this, "Artikel Berhasil
Diunduh", Toast.LENGTH_LONG).show()
    }
}
// pullToRefresh
binding.pullToRefresh.setOnRefreshListener {
binding.svArticle.visibility = View.GONE
binding.pullToRefresh.isRefreshing = false
binding.errorMessage.visibility = View.GONE
binding.progressMessage.visibility = View.VISIBLE
GlobalScope.launch(Dispatchers.Main) {
    val deferredArticles = async(Dispatchers.IO) {
        id.let {
readArticleViewModel.setArticle(applicationContext, it) }
        }
        val it = deferredArticles.await()
        if (it["status"] == true) {
            title = it["title"].toString()
            content = it["content"].toString()

binding.ContentReadArticle.tvArticleTitle.text =
it["title"].toString()
            if (it["hasImg"] == true) {
                Glide.with(applicationContext)

.load(resources.getString(R.string.server) +
"assets/articles/${id}.${it["extension"]}")

.into(binding.ContentReadArticle.imgArticle)
            } else {

binding.ContentReadArticle.imgArticle.visibility = View.GONE
            }
            val likes = it["like"].toString() + "
${resources.getString(R.string.likes)}"
            binding.ContentReadArticle.tvLikeCount.text
= likes
            binding.ContentReadArticle.tvPostDate.text
= it["post_date"].toString()

```

```

binding.ContentReadArticle.tvPostContent.text =
it["content"].toString()
binding.progressMessage.visibility =
View.GONE
binding.svArticle.visibility = View.VISIBLE
} else {
binding.progressMessage.visibility =
View.GONE
binding.errorMessage.visibility =
View.VISIBLE
val errorMessage = ""
Error!
[ ${it["code"]} ]: ${it["message"]}
"".trimIndent()
binding.tvErrorMessage.text = errorMessage
}
}
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
menuInflater.inflate(R.menu.menu_article_kajian, menu)
return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean
{
when(item.itemId) {
android.R.id.home -> super.onBackPressed()
R.id.share -> {
val mimeType = "text/plain"
ShareCompat.IntentBuilder.from(this).apply {
setType(mimeType)
setChooserTitle("Bagikan artikel ini
sekarang!")
setText("
*${binding.ContentReadArticle.tvArticleTitle.text}*
_${binding.ContentReadArticle.tvNamaUstad.text} pada
${binding.ContentReadArticle.tvPostDate.text}_
${binding.ContentReadArticle.tvPostContent.text}
Informasi ini disebarkan melalui aplikasi
${getString(R.string.app_name)}.
"".trimIndent()
startChooser()
}
}
}
return super.onOptionsItemSelected(item)
}
}

```

```
}
```

- ReadArticleViewModel.java

```
package com.kajianid.android.viewmodels;

import android.content.Context;
import android.os.Looper;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;
import com.kajianid.android.R;
import com.kajianid.android.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import cz.msebera.android.httpclient.Header;

public class ReadArticleViewModel extends ViewModel {

    private MutableLiveData<Map<String, Object>> mData = new
    MutableLiveData<>();

    public Map<String, Object> setArticle(Context context,
    String id) {
        try {
            Looper.prepare();
        } catch (Exception ignored) {

        }

        Map<String, Object> resultMap = new HashMap<>();
        String url = context.getString(R.string.server) +
        "api/articles";
        SyncHttpClient client = new SyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("read", "1");
        params.put("ustadz_mode", "0");
        params.put("article", id);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[]
            headers, byte[] responseBody) {
```

```

        try {
            String result = new String(responseBody);
            JSONObject resultObject = new
JSONObject(result);
            JSONArray list =
resultObject.getJSONArray("data");
            String like =
resultObject.getString("likes");

            JSONObject articleJSONObject =
list.getJSONObject(0);
            String title =
articleJSONObject.getString("title");
            boolean hasImg =
StringHelper.convertToBoolean(articleJSONObject.getString("has_
img"));
            String extension =
articleJSONObject.getString("extension");
            String content =
articleJSONObject.getString("content");
            String ustadzName =
articleJSONObject.getString("ustadz_name");

            String date =
articleJSONObject.getString("post_date");
            Date datePost = new SimpleDateFormat("yyyy-
MM-dd HH:mm:ss", Locale.getDefault()).parse(date);
            assert datePost != null;
            String datePostFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(datePost);

            resultMap.put("status", true);
            resultMap.put("id", id);
            resultMap.put("title", title);
            resultMap.put("hasImg", hasImg);
            resultMap.put("extension", extension);
            resultMap.put("content", content);
            resultMap.put("post_date",
datePostFormatted);
            resultMap.put("ustadz_name", ustadzName);
            resultMap.put("like", like);
            mData.postValue(resultMap);
        } catch (JSONException | ParseException e) {
            resultMap.put("status", false);
            resultMap.put("code", 0);
            resultMap.put("message",
e.getLocalizedMessage() + " id: " + id);
            mData.postValue(resultMap);
            e.printStackTrace();
        }
    }
}

```



```

        @Override
        public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
            resultMap.put("status", false);
            resultMap.put("code", statusCode);
            resultMap.put("message",
error.getLocalizedMessage() + " id: " + id);
            mData.postValue(resultMap);
        }
    });

    return resultMap;
}

public Map<String, Object> setArticleAsync(Context context,
String id){
    Map<String, Object> resultMap = new HashMap<>();
    String url = context.getString(R.string.server)
+"api/articles";
    AsyncHttpClient client = new AsyncHttpClient();

    RequestParams params = new RequestParams();
    params.put("read", "1");
    params.put("ustadz mode", "0");
    params.put("article", id);

    client.get(url, params, new AsyncHttpResponseHandler() {
        @Override
        public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
            try {
                String result = new String(responseBody);
                JSONObject resultObject = new
JSONObject(result);
                JSONArray list =
resultObject.getJSONArray("data");
                String like =
resultObject.getString("likes");

                JSONObject articleJSONObject =
list.getJSONObject(0);
                String title =
articleJSONObject.getString("title");
                boolean hasImg =
StringHelper.convertToBoolean(articleJSONObject.getString("has_
img"));

                String extension =
articleJSONObject.getString("extension");
                String content =
articleJSONObject.getString("content");
                String ustadzName =
articleJSONObject.getString("ustadz_name");

```

```

        String date =
articleJSONObject.getString("post_date");
        Date datePost = new SimpleDateFormat("yyyy-
MM-dd HH:mm:ss", Locale.getDefault()).parse(date);
        assert datePost != null;
        String datePostFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(datePost);

        resultMap.put("status", true);
        resultMap.put("id", id);
        resultMap.put("title", title);
        resultMap.put("hasImg", hasImg);
        resultMap.put("extension", extension);
        resultMap.put("content", content);
        resultMap.put("post_date",
datePostFormatted);

        resultMap.put("ustadz_name", ustadzName);
        resultMap.put("like", like);
        mData.postValue(resultMap);

        //apa fungsine?? apakah di baris 148-153
sama sama resulmap dg method on failur dibawahnya
        //jika dilihat dan dibandingkan dg
shomehome catch ini tdk ada

    } catch (JSONException | ParseException e) {
        resultMap.put("status", false);
        resultMap.put("code", 0);
        resultMap.put("message",
e.getLocalizedMessage() + " id: " + id);
        mData.postValue(resultMap);
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
    resultMap.put("status", false);
    resultMap.put("code", statusCode);
    resultMap.put("message",
error.getLocalizedMessage() + " id: " + id);
    mData.postValue(resultMap);
}

});

return resultMap;
}

public LiveData<Map<String, Object>> getArticle() {

```

```

        return mData;
    }
}

```

- DownloadedArticleFragment.kt

```

package com.kajianid.android.fragment

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.android.adapter.ArticleSQLAdapter
import com.kajianid.android.databases.article.DbArticleHelper
import com.kajianid.android.databases.article.MappingHelper
import
com.kajianid.android.databinding.FragmentDownloadedArticleBindi
ng
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch

class DownloadedArticleFragment : Fragment() {
    private lateinit var articleSQLAdapter: ArticleSQLAdapter
    private lateinit var dbArticleHelper: DbArticleHelper
    private lateinit var binding:
FragmentDownloadedArticleBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        articleSQLAdapter = ArticleSQLAdapter(context)
    }

    override fun onCreateView(inflater: LayoutInflater,
container: ViewGroup?, savedInstanceState: Bundle?): View? {
        binding =
FragmentDownloadedArticleBinding.inflate(inflater)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState:
Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        binding.rvArticles.layoutManager =
LinearLayoutManager(context)
        binding.rvArticles.adapter = articleSQLAdapter
        binding.rvArticles.setHasFixedSize(true) //ukuran tidak
aka berubah jk diputar

        //manggil database (buka)
        dbArticleHelper =
DbArticleHelper.getInstance(requireContext().applicationContext

```

```

)

        dbArticleHelper.open ()

        loadArticle ()

    }

    private fun loadArticle () {
        GlobalScope.launch (Dispatchers.Main) {
            val deferredArticle = async (Dispatchers.IO) {
                val cursor = dbArticleHelper.queryAll ()
                MappingHelper.mapCursorToArrayList (cursor)
            }
            val article = deferredArticle.await ()
            if (article.size > 0)
                articleSQLAdapter.setArticle (article)
            else {
                articleSQLAdapter.setArticle (ArrayList ())
            }
        }
    }

    override fun onResume () {
        super.onResume ()
        loadArticle ()
    }
}

```

#### e. Menu Kajian

##### a) Tampilan

- activity\_kajian.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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=".activities.KajianActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize" />
    </AppBarLayout>
</RelativeLayout>

```

```

</com.google.android.material.appbar.AppBarLayout>

<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:visibility="visible"/>

<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
    android:id="@+id/pullToRefresh"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/appBar">

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

app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:listitem="@layout/list_kajian"/>

</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

</RelativeLayout>

```

- list\_data\_kajian.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/layout_data_kajian"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:focusable="true"
    android:paddingLeft="8dp"
    android:paddingRight="8dp">

    <androidx.cardview.widget.CardView
        android:id="@+id/cv_item_note"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        app:cardElevation="4dp"
        app:cardMaxElevation="4dp"
        app:cardUseCompatPadding="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:padding="16dp"
            android:layout_height="wrap_content">

```

```

<ImageView
    android:id="@+id/Image_Judul"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:contentDescription="foto"
    android:src="@drawable/foto" />

<TextView
    android:id="@+id/TextView_Judul"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:ellipsize="end"
    android:maxLines="1"
    android:text="@string/example_kajian_title"
    android:textSize="18sp"
    android:textStyle="bold"
    android:textColor="?android:textColorPrimary"/>

<TextView
    android:id="@+id/TextView_Ustadz"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:maxLines="1"
    android:ellipsize="end"
    android:layout_below="@+id/TextView_Judul"
    android:layout_marginStart="16dp"
    android:layout_marginTop="4dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:text="@string/example_ustads" />

<TextView
    android:id="@+id/TextView_Mosque"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/TextView_Ustadz"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:ellipsize="end"
    android:maxLines="1"
    android:text="Masjid Fatimatuzzahra" />

<TextView
    android:id="@+id/TextView_Keterangan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/TextView_Mosque"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:text="@string/example_keterangan" />

```

```

        <TextView
            android:id="@+id/TextView_Tanggal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/TextView_Mosque"
            android:layout_alignParentEnd="true"
            android:text="@string/example_tanggal" />
    </RelativeLayout>

    </androidx.cardview.widget.CardView>

</RelativeLayout>

```

- list\_kajian.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/layout_data_kajian"
    android:layout_height="wrap_content"
    android:background="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:focusable="true"
    android:paddingLeft="8dp"
    android:paddingRight="8dp">

    <androidx.cardview.widget.CardView
        android:id="@+id/cv_item_note"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        app:cardElevation="4dp"
        app:cardMaxElevation="4dp"
        app:cardUseCompatPadding="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:padding="16dp"
            android:layout_height="wrap_content">

            <ImageView
                android:id="@+id/Image_Judul"
                android:layout_width="70dp"
                android:layout_height="70dp"
                android:contentDescription="foto"
                android:src="@drawable/ic_menu_foto" />

            <TextView
                android:id="@+id/TextViewJudul"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="16dp"

```

```

        android:layout_toEndOf="@+id/Image_Judul"
        android:ellipsize="end"
        android:maxLines="1"
        android:text="@string/example_kajian_title"
        android:textSize="18sp"
        android:textStyle="bold"
        android:textColor="?android:textColorPrimary"/>

<TextView
    android:id="@+id/TextViewUstadz"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:maxLines="1"
    android:ellipsize="end"
    android:layout_below="@+id/TextViewJudul"
    android:layout_marginStart="16dp"
    android:layout_marginTop="4dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:text="@string/example_ustadz" />

<TextView
    android:id="@+id/TextView_Mosque"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/TextViewUstadz"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:ellipsize="end"
    android:maxLines="1"
    android:text="@string/example_mosque" />

<TextView
    android:id="@+id/TextView_Keterangan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/TextView_Mosque"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/Image_Judul"
    android:text="@string/example_keterangan" />

<TextView
    android:id="@+id/TextView_Tanggal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/TextView_Mosque"
    android:layout_alignParentEnd="true"
    android:text="@string/example_tanggal" />
</RelativeLayout>
</androidx.cardview.widget.CardView>
</RelativeLayout>

```

- activity\_show\_kajian.xml



```

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

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize" />

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/pullToRefresh"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar">

        <ScrollView
            android:id="@+id/svKajian"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <include
                android:id="@+id/content_show_kajian"
                layout="@layout/content_show_kajian" />

        </ScrollView>

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

    <RelativeLayout
        android:id="@+id/progressMessage"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar"
        android:layout_centerInParent="true"
        android:background="@android:color/white"
        android:padding="30dp">

        <ProgressBar
            android:id="@+id/progressBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true" />

```

```

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/progressBar"
            android:layout_centerHorizontal="true"
            android:paddingTop="16dp"
            android:text="@string/load_data"
            android:textAlignment="center" />

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/errorMessage"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar"
        android:layout_centerInParent="true"
        android:background="@android:color/white"
        android:padding="30dp"
        android:visibility="gone">

        <ImageView
            android:id="@+id/imgWarning"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_centerHorizontal="true"
            android:contentDescription="@string/warning"
            android:src="@drawable/ic_baseline_warning_24" />

        <TextView
            android:id="@+id/tvErrorMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/imgWarning"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:text="@string/example_error_message"
            android:textAlignment="center" />

        <Button
            android:id="@+id/btnRefresh"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/tvErrorMessage"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:text="@string/refresh" />

    </RelativeLayout>

    <com.google.android.material.floatingactionbutton.FloatingAction
    nButton
        android:id="@+id/fabreminder"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="24dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:tint="?android:textColorPrimary"
        android:foregroundGravity="bottom"
        app:fabSize="normal"

        app:srcCompat="@drawable/ic_baseline_notifications_none_24"/>

    </RelativeLayout>

```

- content\_show\_kajian.xml

```

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

    <ImageView
        android:id="@+id/imgThumbnail"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:src="@drawable/icon"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:visibility="visible" />

    <ImageButton
        android:id="@+id/btnPlay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:src="@drawable/ic_baseline_play_circle_outline_100"
        android:background="@android:color/transparent"

        app:layout_constraintBottom_toBottomOf="@+id/imgThumbnail"
        app:layout_constraintEnd_toEndOf="@+id/imgThumbnail"

        app:layout_constraintStart_toStartOf="@+id/imgThumbnail"
        app:layout_constraintTop_toTopOf="@+id/imgThumbnail"
        android:contentDescription="@string/play_video" />

    <TextView
        android:id="@+id/tvKajianTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="16dp"
        android:text="@string/example_kajian_title"

```

```

        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imgThumbnail"
    />

    <TextView
        android:id="@+id/tvMosqueAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/example_mosque"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/tvKajianTitle" />

    <TextView
        android:id="@+id/tvUstadzName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/by"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/tvMosqueAddress" />

    <TextView
        android:id="@+id/tvCategory"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/on_stage"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvUstadzName"
    />

    <TextView
        android:id="@+id/tvTimestampAnnounce"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/timestamp_announce"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvCategory"
    />

    <TextView
        android:id="@+id/tvTimestampDue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/timestamp_due"
        app:layout_constraintStart_toStartOf="parent"

```

```

app:layout_constraintTop_toBottomOf="@+id/tvTimestampAnnounce"
/>

    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/description"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/tvTimestampDue" />

    <TextView
        android:id="@+id/tvDescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="84dp"
        android:text="@string/example_text"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView7" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

- menu\_article\_kajian.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/share"
        android:icon="@drawable/ic_baseline_share_24"
        android:title="@string/share"
        app:showAsAction="collapseActionView|ifRoom" />

</menu>

```

- fragment\_saved\_kajian.xml

```

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

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

```

```

app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:listitem="@layout/list_kajian" />

</RelativeLayout>

```

## b) Program

### - Kajian.java

```

package com.kajianid.android.data;

import android.os.Parcel;
import android.os.Parcelable;

public class Kajian implements Parcelable {
    protected Kajian(Parcel in) {
        id = in.readString();
        title = in.readString();
        mosque = in.readString();
        place = in.readString();
        ustadzName = in.readString();
        description = in.readString();
        address = in.readString();
        youtubelink = in.readString();
        date = in.readString();
        dateAnnounce = in.readString();
        imgResource = in.readString();
    }

    public Kajian() {

    }

    public static final Creator<Kajian> CREATOR = new
    Creator<Kajian>() {
        @Override
        public Kajian createFromParcel(Parcel in) {
            return new Kajian(in);
        }

        @Override
        public Kajian[] newArray(int size) {
            return new Kajian[size];
        }
    };

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
}

```

```

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getMosque() {
    return mosque;
}

public void setMosque(String mosque) {
    this.mosque = mosque;
}

public String getPlace() {
    return place;
}

public void setPlace(String place) {
    this.place = place;
}

public String getUstadzName() {
    return ustadzName;
}

public void setUstadzName(String ustadzName) {
    this.ustadzName = ustadzName;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getYoutubelink() {
    return youtubelink;
}

public void setYoutubelink(String youtubelink) {

```

```

        this.youtubelink = youtubelink;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public String getDateAnnounce() {
        return dateAnnounce;
    }

    public void setDateAnnounce(String dateAnnounce) {
        this.dateAnnounce = dateAnnounce;
    }

    public String getImgResource() {
        return imgResource;
    }

    public void setImgResource(String imgResource) {
        this.imgResource = imgResource;
    }

    private String id;
    private String title;
    private String mosque;
    private String place;
    private String ustadzName;
    private String description;
    private String address;
    private String youtubelink;
    private String date;
    private String dateAnnounce;
    private String imgResource;

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(id);
        dest.writeString(title);
        dest.writeString(mosque);
        dest.writeString(place);
        dest.writeString(ustadzName);
        dest.writeString(description);
        dest.writeString(address);
    }

```



```

        dest.writeString(youtubelink);
        dest.writeString(date);
        dest.writeString(dateAnnounce);
        dest.writeString(imgResource);
    }
}

```

#### - KajianActivity.java

```

package com.kajianid.android.activities

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MenuItem
import android.view.View
import android.widget.SearchView
import android.widget.Toast
import androidx.lifecycle.Observer
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.android.R
import com.kajianid.android.adapter.ListKajianAdapter
import com.kajianid.android.databinding.ActivityKajianBinding
import com.kajianid.android.viewmodels.KajianViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import org.json.JSONObject

class KajianActivity : AppCompatActivity() {
    var id = ""
    private lateinit var kajianViewModel: KajianViewModel
    private var listKajianAdapter: ListKajianAdapter =
ListKajianAdapter(this)
    private var searchView: SearchView? = null
    private lateinit var binding: ActivityKajianBinding
    override fun onOptionsItemSelected(item: MenuItem): Boolean
    {
        if (item.itemId == android.R.id.home) {
            super.onBackPressed()
        }
        return super.onOptionsItemSelected(item)
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityKajianBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.progressBar.visibility = View.VISIBLE

        // fungsi Toolbar dan ActionBar(back pd kajian)
        setSupportActionBar(binding.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
    }
}

```

```

supportActionBar?.setTitle(resources.getString(R.string.show_kajian))

        binding.recDataKajian.layoutManager =
LinearLayoutManager(this)
        binding.recDataKajian.adapter = listKajianAdapter

        kajianViewModel = ViewModelProvider(this,
ViewModelProvider.NewInstanceFactory())
            .get(KajianViewModel::class.java)
        kajianViewModel.kajian.observe(this, Observer {
            if (it != null) {
                binding.progressBar.visibility = View.GONE
                listKajianAdapter.setKajian(it)
            }
        })
        kajianViewModel.setKajianAsync(this, listKajianAdapter,
        "")

        binding.pullToRefresh.setOnRefreshListener {
            GlobalScope.launch (Dispatchers.Main) {
                val deferredKajian = async(Dispatchers.IO) {
kajianViewModel.setKajian(this@KajianActivity, "")
                }
                val status = deferredKajian.await()
                if (status != null) {
                    val parse = JSONObject(status)
                    Toast.makeText(this@KajianActivity,
parse.getString("message"), Toast.LENGTH_SHORT).show()
                } else {
                    listKajianAdapter.notifyDataSetChanged()
                }
                binding.pullToRefresh.isRefreshing = false
            }
        }
    }
}

```

#### - KajianSQLAdapter.java

```

package com.kajianid.android.adapter;

import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.kajianid.android.activities.ShowKajianSQLActivity;
import com.kajianid.android.data.Kajian;
import com.kajianid.android.databinding.ListKajianBinding;
import java.util.ArrayList;

```

```

public class KajianSQLAdapter extends
RecyclerView.Adapter<KajianSQLAdapter.KajianSQLViewHolder> {
    ArrayList<Kajian> mKajian = new ArrayList<>();

    public KajianSQLAdapter() {
    }

    public void setKajian(ArrayList<Kajian> mKajian) {
        if (mKajian != null) this.mKajian.clear();
        assert mKajian != null;
        this.mKajian.addAll(mKajian);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public KajianSQLViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        ListKajianBinding binding =
ListKajianBinding.inflate(LayoutInflater.from(parent.getContext
()));
        return new
KajianSQLAdapter.KajianSQLViewHolder(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull
KajianSQLAdapter.KajianSQLViewHolder holder, int position) {
        holder.bind(mKajian.get(position));
    }

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

    public class KajianSQLViewHolder extends
RecyclerView.ViewHolder {
        private final ListKajianBinding binding;

        public KajianSQLViewHolder(ListKajianBinding itemView)
{
            super(itemView.getRoot());
            this.binding = itemView;
        }

        public void bind(Kajian kajian) {
            binding.TextViewJudul.setText(kajian.getTitle());
binding.TextViewUstadz.setText(kajian.getUstadzName());
            binding.TextViewMosque.setText(kajian.getMosque());

```

```

binding.TextViewKeterangan.setText(kajian.getDescription());
binding.TextViewTanggal.setText(kajian.getDate());

if (kajian.getImgResource().equals("false")) {
    binding.ImageJudul.setVisibility(View.GONE);
} else {
    Glide.with(binding.getRoot().getContext())
        .asBitmap()
        .load(kajian.getImgResource())
        .into(binding.ImageJudul);
}

binding.getRoot().setOnClickListener(view -> {
    Intent kajianIntent = new
Intent(binding.getRoot().getContext(),
ShowKajianSQLActivity.class);

kajianIntent.putExtra(ShowKajianSQLActivity.EXTRA_PARCEL_KAJIAN
, kajian);

binding.getRoot().getContext().startActivity(kajianIntent);
});
}
}
}

```

- ListKajianAdapter.java

```

package com.kajianid.android.adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.kajianid.android.activities.ShowKajianActivity;
import com.kajianid.android.data.Kajian;
import com.kajianid.android.databinding.ListKajianBinding;
import java.util.ArrayList;
import java.util.Objects;

public class ListKajianAdapter extends
RecyclerView.Adapter<ListKajianAdapter.MyHolder>{
    public ListKajianAdapter(Context context) {
        this.context = context;
    }
    ArrayList<Kajian> mKajian = new ArrayList<>();
    private Context context;

    public void setKajian(ArrayList<Kajian> mKajian) {
        if (mKajian != null) this.mKajian.clear();
        assert mKajian != null;
    }
}

```

```

        this.mKajian.addAll(mKajian);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        ListKajianBinding binding =
ListKajianBinding.inflate(LayoutInflater.from(parent.getContext
()));
        return new MyHolder(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull final MyHolder
holder, final int position) {
        holder.bind(mKajian.get(position));
    }

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

    static class MyHolder extends RecyclerView.ViewHolder{

        private ListKajianBinding binding;

        public MyHolder(@NonNull ListKajianBinding itemView) {
            super(itemView.getRoot());
            this.binding = itemView;
        }

        public void bind(Kajian kajian) {
            Glide.with(binding.getRoot().getContext())
                .asBitmap()
                .load(kajian.getImgResource())
                .into(binding.ImageJudul);

            binding.TextViewJudul.setText(kajian.getTitle());
            binding.TextViewUstadz.setText(kajian.getUstadzName());
            binding.TextViewMosque.setText((Objects.equals(kajian.getAddres
s(), "null")) ? kajian.getMosque() : kajian.getAddress());
            binding.TextViewKeterangan.setText(kajian.getPlace());
            binding.TextViewTanggal.setText(kajian.getDate());

            binding.getRoot().setOnClickListener(new
View.OnClickListener() {
                @Override

```

```

        public void onClick(View view) {
            Intent KajianIntent = new
Intent(binding.getRoot().getContext(),
ShowKajianActivity.class);

KajianIntent.putExtra(ShowKajianActivity.EXTRA_KAJIAN_ID,
kajian.getId());

binding.getRoot().getContext().startActivity(KajianIntent);
        }
    });
}
}
}

```

- DbKajianHelper.kt

```

package com.kajianid.android.databases.kajian

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteException
import com.kajianid.android.databases.DatabaseHelper
import
com.kajianid.android.databases.DatabaseContract.KajianColumns.C
ompanion.ID
import
com.kajianid.android.databases.DatabaseContract.KajianColumns.C
ompanion.TABLE_NAME_KAJIAN

class DbKajianHelper (context: Context) {
    companion object {
        private const val DATABASE_TABLE = TABLE_NAME_KAJIAN
        private lateinit var databaseHelper: DatabaseHelper
        private var INSTANCE: DbKajianHelper? = null
        private lateinit var database: SQLiteDatabase

        fun getInstance(context: Context): DbKajianHelper =
INSTANCE ?: synchronized(this) {
            INSTANCE ?: DbKajianHelper(context)
        }
    }

    init {
        databaseHelper = DatabaseHelper(context)
    }

    @Throws(SQLiteException::class)
    fun open() {
        database = databaseHelper.writableDatabase
    }

    fun close() {

```

```

        databaseHelper.close()
        if (database.isOpen) database.close()
    }

    fun queryAll(): Cursor {
        return database.query(TABLE_NAME_KAJIAN,
            null,
            null,
            null,
            null,
            null,
            null)
    }

    fun queryById(id: String): Cursor {
        return database.query(
            TABLE_NAME_KAJIAN,
            null,
            "$ID = ?",
            arrayOf(id),
            null,
            null,
            null,
            null)
    }

    fun insert(values: ContentValues?): Long {
        return database.insert(DATABASE_TABLE, null, values)
    }

    fun update(id: String, values: ContentValues?): Int {
        return database.update(DATABASE_TABLE, values, "$ID = ?", arrayOf(id))
    }

    fun deleteById(id: String): Int {
        return database.delete(DATABASE_TABLE, "$ID = '$id'", null)
    }
}

```

#### - MappingHelper.kt

```

package com.kajianid.android.databases.kajian

import android.database.Cursor
import com.kajianid.android.data.Kajian
import com.kajianid.android.databases.DatabaseContract

object MappingHelper {
    fun mapCursorToArrayList(kajianCursor: Cursor?):
    ArrayList<Kajian> {
        var kajian2 = ArrayList<Kajian>()
        kajianCursor?.apply {
            while (moveToNext()) {

```

```

        val id =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
ID))
        val kajian_title =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
KAJIAN_TITLE))
        val ustadz_name =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
USTADZ_NAME))
        val mosque_name =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
MOSQUE_NAME))
        val address =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
ADDRESS))
        val place =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
PLACE))
        val youtube_link =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
YOUTUBE_LINK))
        val description =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
DESCRIPTION))
        val img_resource =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
IMG_RESOURCE))
        val date_announce =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
DATE_ANNOUNCE))
        val date_due =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
DATE_DUE))

        val kajian = Kajian()

        kajian.id = id
        kajian.title = kajian_title
        kajian.ustadzName = ustadz_name
        kajian.mosque = mosque_name
        kajian.address = address
        kajian.place = place
        kajian.youtubelink = youtube_link
        kajian.description = description
        kajian.imgResource = img_resource
        kajian.dateAnnounce = date_announce
        kajian.date = date_due
        kajian2.add(kajian)
    }
}
return kajian2
}
fun mapCursorToObject(kajianCursor: Cursor?): Kajian {
    var kajian = Kajian()

```



```

        kajianCursor?.apply {
            moveToFirst()
            val id =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
ID))
            val kajian_title =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
KAJIAN_TITLE))
            val ustadz_name =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
USTADZ_NAME))
            val mosque_name =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
MOSQUE_NAME))
            val address =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
ADDRESS))
            val place =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
PLACE))
            val youtube_link =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
YOUTUBE_LINK))
            val description =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
DESCRIPTION))
            val img_resource =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
IMG_RESOURCE))
            val date_announce =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
DATE_ANNOUNCE))
            val date_due =
getString(getColumnIndexOrThrow(DatabaseContract.KajianColumns.
DATE_DUE))

            kajian.id = id
            kajian.title = kajian_title
            kajian.ustadzName = ustadz_name
            kajian.mosque = mosque_name
            kajian.address = address
            kajian.place = place
            kajian.youtubelink = youtube_link
            kajian.description = description
            kajian.imgResource = img_resource
            kajian.dateAnnounce = date_announce
            kajian.date = date_due
        }
        return kajian
    }
}

```

- DatabaseContract.kt

```

package com.kajianid.android.databases

```

```

import android.provider.BaseColumns

object DatabaseContract {
    class KajianColumns : BaseColumns {
        companion object {
            const val TABLE_NAME_KAJIAN = "kajian"
            val ID = "id"
            val KAJIAN_TITLE = "kajian_title"
            val USTADZ_NAME = "ustadz_name"
            val MOSQUE_NAME = "mosque_name"
            val ADDRESS = "address"
            val PLACE = "place"
            val YOUTUBE_LINK = "youtube_link"
            val DESCRIPTION = "description"
            val IMG_RESOURCE = "img_resource"
            val DATE_ANNOUNCE = "date_announce"
            val DATE_DUE = "date_due"
        }
    }

    class ArticleColumns : BaseColumns {
        companion object {
            const val TABLE_NAME_ARTICLES = "articles"
            val ID = "id"
            val TITLE = "title"
            val POST_DATE = "post_data"
            val CONTENT = "content"
            val HAS_IMG = "has_img"
            val USTADZ_NAME = "ustadz_name"
            val IMGURL = "img_url"
        }
    }
}

```

- DatabaseHelper.kt

```

package com.kajianid.android.databases

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import com.kajianid.android.databases.DatabaseContract.ArticleColumns.Companion.TABLE_NAME_ARTICLES
import com.kajianid.android.databases.DatabaseContract.KajianColumns.Companion.TABLE_NAME_KAJIAN

class DatabaseHelper (context: Context):
    SQLiteOpenHelper(context, DATABASE_NAME, null,
        DATABASE_VERSION) {
    companion object {
        private const val DATABASE_NAME = "kajianid"
        private const val DATABASE_VERSION = 1
    }
}

```

```

        private val SQL_CREATE_NAME_KAJIAN = """
            CREATE TABLE $TABLE_NAME_KAJIAN(
                ${DatabaseContract.KajianColumns.ID} varchar(5) not
null primary key,
                ${DatabaseContract.KajianColumns.KAJIAN_TITLE}
Text,
                ${DatabaseContract.KajianColumns.USTADZ_NAME}
varchar(15),
                ${DatabaseContract.KajianColumns.MOSQUE_NAME}
varchar(30),
                ${DatabaseContract.KajianColumns.ADDRESS} Text,
                ${DatabaseContract.KajianColumns.PLACE}
varchar(30),
                ${DatabaseContract.KajianColumns.YOUTUBE_LINK}
Text,
                ${DatabaseContract.KajianColumns.DESCRPTION} Text,
                ${DatabaseContract.KajianColumns.IMG_RESOURCE}
Text,
                ${DatabaseContract.KajianColumns.DATE_ANNOUNCE}
Text,
                ${DatabaseContract.KajianColumns.DATE_DUE} Text)
        """.trimIndent()

        private val SQL_CREATE_TABLE_ARTIKEL = """
            CREATE TABLE $TABLE_NAME_ARTICLES(
                ${DatabaseContract.ArticleColumns.ID} varchar(5) not
null primary key,
                ${DatabaseContract.ArticleColumns.TITLE} text,
                ${DatabaseContract.ArticleColumns.POST_DATE} Text,
                ${DatabaseContract.ArticleColumns.CONTENT} Text,
                ${DatabaseContract.ArticleColumns.HAS_IMG} Int,
                ${DatabaseContract.ArticleColumns.USTADZ_NAME}
varchar(15),
                ${DatabaseContract.ArticleColumns.IMGURL} Text)
        """.trimIndent()
    }

    override fun onCreate(db: SQLiteDatabase?) {
        db?.execSQL(SQL_CREATE_NAME_KAJIAN)
        db?.execSQL(SQL_CREATE_TABLE_ARTIKEL)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion:
Int, newVersion: Int) {
        onCreate(db)
    }
}

```

- KajianViewModel.java

```

package com.kajianid.android.viewmodels;

import android.content.Context;
import android.os.Looper;

```

```

import android.util.Log;
import android.widget.Toast;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;
import com.kajianid.android.adapter.ListKajianAdapter;
import com.kajianid.android.R;
import com.kajianid.android.data.Kajian;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Locale;
import cz.msebera.android.httpclient.Header;

public class KajianViewModel extends ViewModel {
    private MutableLiveData<ArrayList<Kajian>> listKajian = new
    MutableLiveData<>();

    public String setKajian(Context context, String
    searchQuery) {

        try {
            Looper.prepare();
        } catch (Exception e) {
        }
        final String[] ret = {null};
        String url =
context.getResources().getString(R.string.server) +
"api/kajian";
        ArrayList<Kajian> listItems = new ArrayList<>();
        SyncHttpClient client = new SyncHttpClient();

        // Request Parameters
        RequestParams params = new RequestParams();
        params.put("read", "0");
        params.put("ustadz_mode", "0");
        params.put("query", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler()
{
            @Override
            public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
                try {
                    String result = new String(responseBody);

```

```

        JSONObject resultObject = new
JSONObject(result);
        JSONArray list =
resultObject.getJSONArray("data");

        Log.d("Isi", result);

        for (int i = 0; i < list.length(); i++) {
            JSONObject kajianJSONObject =
list.getJSONObject(i);
            Kajian kajian = new Kajian();

kajian.setId(kajianJSONObject.getString("id"));

kajian.setTitle(kajianJSONObject.getString("kajian_title"));

kajian.setUstadzName(kajianJSONObject.getString("ustadz_name"))
;

kajian.setImgResource(kajianJSONObject.getString("img_resource"
));

kajian.setPlace(kajianJSONObject.getString("place"));

kajian.setAddress(kajianJSONObject.getString("address"));

kajian.setMosque(kajianJSONObject.getString("mosque_name"));

            String date =
kajianJSONObject.getString("post_date");
            Date dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault()).parse(date);
            String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue);

            kajian.setDate(dateDueFormatted);
            listItems.add(kajian);
        }
    } catch (ParseException | JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
    error.printStackTrace();
    String expected = new String(responseBody);
    ret[0] = expected;
}

```

```

    }
    });
    return ret[0];
}

    public boolean setKajianAsync(Context context,
ListKajianAdapter adapter, String searchQuery) {
    final boolean[] ret = {false};
    String url =
context.getResources().getString(R.string.server) +
"api/kajian";
    ArrayList<Kajian> listItems = new ArrayList<>();
    AsyncHttpClient client = new AsyncHttpClient();

    RequestParams params = new RequestParams();
    params.put("read", "0");
    params.put("ustadz_mode", "0");
    params.put("query", searchQuery);

    client.get(url, params, new AsyncHttpResponseHandler()
{
    @Override
    public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
        try {
            String result = new String(responseBody);
            JSONObject responseObject = new
JSONObject(result);
            JSONArray list =
responseObject.getJSONArray("data");

            Log.d("Isi", result);

            for (int i = 0; i < list.length(); i++) {
                JSONObject kajianJSONObject =
list.getJSONObject(i);
                Kajian kajian = new Kajian();

                kajian.setId(kajianJSONObject.getString("id"));

                kajian.setTitle(kajianJSONObject.getString("kajian_title"));

                kajian.setUstadzName(kajianJSONObject.getString("ustadz_name"))
;

                kajian.setImgResource(kajianJSONObject.getString("img_resource"
));

                kajian.setPlace(kajianJSONObject.getString("place"));

                kajian.setAddress(kajianJSONObject.getString("address"));

```

```

kajian.setMosque(kajianJSONObject.getString("mosque_name"));

        String date =
kajianJSONObject.getString("date_due");
        Date dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault()).parse(date);
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue);

        kajian.setDate(dateDueFormatted);
        listItems.add(kajian);
        adapter.notifyDataSetChanged();
    }
    listKajian.postValue(listItems);
    ret[0] = true;
} catch (JSONException | ParseException e) {
    e.printStackTrace();
}

}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {

Toast.makeText(context, "Failure!\n"+error.getMessage()+"\n"+res
ponseBody+"\n"+url, Toast.LENGTH_SHORT).show();
    error.printStackTrace();
    ret[0] = false;
}

});
return ret[0];
}

public LiveData<ArrayList<Kajian>> getKajian() {
    return listKajian;
}

}

```

#### - ShowKajianActivity.kt

```

package com.kajianid.android.activities

import android.content.ContentValues
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity

```

```

import androidx.appcompat.widget.Toolbar
import androidx.core.app.ShareCompat
import androidx.lifecycle.ViewModelProvider
import com.bumptech.glide.Glide
import com.kajianid.android.R
import com.kajianid.android.data.Kajian
import com.kajianid.android.databases.DatabaseContract
import com.kajianid.android.databases.kajian.DbKajianHelper
import com.kajianid.android.databases.kajian.MappingHelper
import
com.kajianid.android.databinding.ActivityShowKajianBinding
import com.kajianid.android.receiver.KajianAlarmReceiver
import com.kajianid.android.viewmodels.ShowKajianViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import java.util.*

class ShowKajianActivity : AppCompatActivity() {
    private lateinit var showKajianViewModel:
ShowKajianViewModel
    private var id : String = ""
    private var reminder = false
    private var dateDueUnformatted: String = ""
    private lateinit var dbKajianHelper: DbKajianHelper
    private lateinit var kajian: Kajian
    private lateinit var binding: ActivityShowKajianBinding

    companion object {
        const val EXTRA_KAJIAN_ID = "extra_kajian_id"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityShowKajianBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val alarmReceiver = KajianAlarmReceiver()

        binding.progressMessage.visibility = View.VISIBLE
        binding.errorMessage.visibility = View.GONE

        val bundle = intent.extras
        id = bundle?.get(EXTRA_KAJIAN_ID).toString()

        setSupportActionBar(binding.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        supportActionBar?.title = "Lihat Kajian"

        // MVVM Show Kajian
        showKajianViewModel = ViewModelProvider(this,

```



```

ViewModelProvider.NewInstanceFactory())
        .get(ShowKajianViewModel::class.java)
        showKajianViewModel.kajian.observe(this, {

            //jika ada data
            if (it["status"] == true) {
                binding.progressMessage.visibility = View.GONE
                binding.errorMessage.visibility = View.GONE
                val title = it["title"].toString()
                val ustadzName =
                    resources.getString(R.string.by) + " " +
                    it["ustadz_name"].toString()
                val mosqueName = it["mosque_name"].toString()
                val address = it["address"].toString()
                val description = it["description"].toString()
                val dateAnnounce =
                    resources.getString(R.string.timestamp_announce) + " " +
                    it["date_announce"].toString()
                val dateDue =
                    resources.getString(R.string.timestamp_due) + " " +
                    it["date_due"]
                dateDueUnformatted =
                    it["date_due_unformatted"].toString()

                binding.contentShowKajian.tvKajianTitle.text =
                    title
                binding.contentShowKajian.tvUstadzName.text =
                    ustadzName
                binding.contentShowKajian.tvDescription.text =
                    description

                binding.contentShowKajian.tvTimestampAnnounce.text =
                    dateAnnounce
                binding.contentShowKajian.tvTimestampDue.text =
                    dateDue

                if (it["category"] == "Di Tempat") {
                    binding.contentShowKajian.btnPlay.visibility = View.GONE
                    binding.contentShowKajian.tvCategory.text =
                        it["category"].toString()

                    binding.contentShowKajian.tvMosqueAddress.text = address
                } else {

                    binding.contentShowKajian.btnPlay.visibility = View.VISIBLE
                    val category =
                        it["category"].toString().toUpperCase(Locale.ROOT) + " -
                        Courtesy of YouTube"
                    binding.contentShowKajian.tvCategory.text =
                        category

                    binding.contentShowKajian.tvMosqueAddress.text = mosqueName

```

```

        val uri = it["youtube_link"].toString()

binding.contentShowKajian.btnPlay.setOnClickListener {
    val i = Intent(Intent.ACTION_VIEW,
Uri.parse(uri))

        startActivity(i)
    }
}

        if (it["img_resource"] != null) {
            Glide.with(this)

.load(it["img_resource"].toString())

.into(binding.contentShowKajian.imgThumbnail)
        } else {

binding.contentShowKajian.imgThumbnail.visibility = View.GONE

binding.contentShowKajian.btnPlay.visibility = View.GONE
        }
        kajian = Kajian()
        kajian.id = it["id"].toString()
        kajian.title = it["title"].toString()
        kajian.mosque = it["mosque_name"].toString()
        kajian.place = it["category"].toString()
        kajian.ustadzName =
it["ustadz_name"].toString()
        kajian.description =
it["description"].toString()
        kajian.address = it["address"].toString()
        kajian.youtubelink =
it["youtube_link"].toString()
        kajian.date = it["date_due"].toString()
        kajian.dateAnnounce =
it["date_announce"].toString()
        kajian.imgResource =
it["img_resource"].toString()
        } else {
            binding.progressBar.visibility = View.GONE
            binding.errorMessage.visibility = View.VISIBLE
            val errorMessage = ""
            Error!
            [${it["code"]}] : ${it["message"]}
            """.trimIndent()
            binding.tvErrorMessage.text = errorMessage
        }
    })
    showKajianViewModel.setKajianAsync(this, id)
    dbKajianHelper = DbKajianHelper.getInstance(this)
    dbKajianHelper.open()
    GlobalScope.launch(Dispatchers.Main) {
        val deferredKajian = async(Dispatchers.IO) {

```

```

        val cursor = dbKajianHelper.queryById(id)
        MappingHelper.mapCursorToArrayList(cursor)
    }
    val isReminder = deferredKajian.await()
    if (isReminder.size == 0){
        reminder = false

binding.fabreminder.setImageResource(R.drawable.ic_baseline_notifications_none_24)
    }else{
        reminder = true

binding.fabreminder.setImageResource(R.drawable.ic_baseline_notifications_active_24)
    }
    binding.fabreminder.setOnClickListener{
        if (reminder){
            val alert = AlertDialog.Builder(this)

alert.setTitle(resources.getString(R.string.sure))

alert.setMessage(resources.getString(R.string.remove_reminder_confirm))

alert.setPositiveButton(resources.getString(R.string.yes)) { _, _ ->
            dbKajianHelper.deleteById(id)
            reminder = false

binding.fabreminder.setImageResource(R.drawable.ic_baseline_notifications_none_24)
            alarmReceiver.cancelReminder(this,
kajian.id.toInt())
        }

alert.setNegativeButton(resources.getString(R.string.no)) { _, _ ->
            /* no-op */
        }
        alert.create().show()
    }else{
        val values = ContentValues()
        values.put(DatabaseContract.KajianColumns.ID,
kajian.id)

values.put(DatabaseContract.KajianColumns.KAJIAN_TITLE,
kajian.title)

values.put(DatabaseContract.KajianColumns.USTADZ_NAME,
kajian.ustadzName)

values.put(DatabaseContract.KajianColumns.MOSQUE_NAME,

```

```

kajian.mosque)

values.put(DatabaseContract.KajianColumns.ADDRESS,
kajian.address)

values.put(DatabaseContract.KajianColumns.PLACE, kajian.place)

values.put(DatabaseContract.KajianColumns.YOUTUBE_LINK,
kajian.youtubelink)

values.put(DatabaseContract.KajianColumns.DESCRPTION,
kajian.description)

values.put(DatabaseContract.KajianColumns.IMG_RESOURCE,
kajian.imgResource)

values.put(DatabaseContract.KajianColumns.DATE_ANNOUNCE,
kajian.dateAnnounce)

values.put(DatabaseContract.KajianColumns.DATE_DUE,
kajian.date)

        dbKajianHelper.insert(values)
        reminded = true

binding.fabreminder.setImageResource(R.drawable.ic_baseline_not
ifications_active_24)
        alarmReceiver.setReminder(this,
kajian.id.toInt(), kajian.title, dateDueUnformatted,
kajian.ustadzName)
    }
}
// Pull-to-Refresh
binding.pullToRefresh.setOnRefreshListener {
    binding.pullToRefresh.isRefreshing = false
    binding.svKajian.visibility = View.GONE
    binding.progressMessage.visibility = View.VISIBLE
    binding.errorMessage.visibility = View.GONE
    GlobalScope.launch(Dispatchers.Main) {
        val deferredKajian = async(Dispatchers.IO) {
showKajianViewModel.setKajian(applicationContext, id)
        }
        val it = deferredKajian.await()
        if (it["status"] == true) {
            binding.progressMessage.visibility =
View.GONE

            binding.errorMessage.visibility = View.GONE
            val title = it["title"].toString()
            val ustadzName =
resources.getString(R.string.by) + " " +
it["ustadz_name"].toString()
            val mosqueName =
it["mosque_name"].toString()

```

```

        val address = it["address"].toString()
        val description =
it["description"].toString()
        val dateAnnounce =
resources.getString(R.string.timestamp_announce) + " " +
it["date_announce"].toString()
        val dateDue =
resources.getString(R.string.timestamp_due) + " " +
it["date_due"]

        dateDueUnformatted =
it["date_due_unformatted"].toString()

        // write to model class
        kajian.id = it["id"].toString()
        kajian.title = it["title"].toString()
        kajian.mosque =
it["mosque_name"].toString()
        kajian.place = it["category"].toString()
        kajian.youtubelink =
it["youtube_link"].toString()
        kajian.description =
it["description"].toString()
        kajian.address = it["address"].toString()
        kajian.imgResource =
it["img_resource"].toString()
        kajian.date =
it["date_due_unformatted"].toString()

binding.contentShowKajian.tvKajianTitle.text = title
        binding.contentShowKajian.tvUstadzName.text
= ustadzName

binding.contentShowKajian.tvDescription.text = description

binding.contentShowKajian.tvTimestampAnnounce.text =
dateAnnounce

binding.contentShowKajian.tvTimestampDue.text = dateDue

        if (it["category"] == "Di Tempat") {

binding.contentShowKajian.btnPlay.visibility = View.GONE

binding.contentShowKajian.tvCategory.text =
it["category"].toString()

binding.contentShowKajian.tvMosqueAddress.text = address
        } else {

binding.contentShowKajian.btnPlay.visibility = View.VISIBLE
        val category =
it["category"].toString().toUpperCase(Locale.ROOT) + " -

```

```

    Courtesy of YouTube"

    binding.contentShowKajian.tvCategory.text = category

    binding.contentShowKajian.tvMosqueAddress.text = mosqueName
        val uri = it["youtube_link"].toString()

    binding.contentShowKajian.btnPlay.setOnClickListener {
        val i = Intent(Intent.ACTION_VIEW,
        Uri.parse(uri))

        startActivity(i)
    }

    if (it["img_resource"] != null) {
        Glide.with(applicationContext)

        .load(it["img_resource"].toString())

        .into(binding.contentShowKajian.imgThumbnail)
    } else {

    binding.contentShowKajian.imgThumbnail.visibility = View.GONE

    binding.contentShowKajian.btnPlay.visibility = View.GONE
    }
    binding.svKajian.visibility = View.VISIBLE
    } else {
        binding.progressMessage.visibility =
    View.GONE

        binding.errorMessage.visibility =
    View.VISIBLE

        val errorMessage = ""
            Error!
            [${it["code"]}] : ${it["message"]}
            """.trimIndent()
        binding.tvErrorMessage.text = errorMessage
    }
    }
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_article_kajian, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean
{
    when(item.itemId){
        android.R.id.home -> super.onBackPressed()
        R.id.share -> {
            val mimeType = "text/plain"

```

```

        ShareCompat.IntentBuilder.from(this).apply {
            setType(mimeType)
            setChooserTitle("Bagikan kajian ini
sekarang!")
            setText("""
*Kajian ${binding.contentShowKajian.tvKajianTitle.text}*
${binding.contentShowKajian.tvUstadzName.text}
Waktu Pelaksanaan:
${binding.contentShowKajian.tvTimestampDue.text}

Kajian ini bersifat ${kajian.place}

Deskripsi:
${kajian.description}

Informasi ini disebarakan melalui aplikasi
${getString(R.string.app_name)}.
        """.trimIndent())
            startChooser()
        }
    }
}
return super.onOptionsItemSelected(item)
}
}

```

- ShowKajianSQLActivity.kt

```

package com.kajianid.android.activities

import android.content.ContentValues
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.Toolbar
import androidx.core.app.ShareCompat
import com.bumptech.glide.Glide
import com.kajianid.android.R
import com.kajianid.android.data.Kajian
import com.kajianid.android.databases.DatabaseContract
import com.kajianid.android.databases.kajian.DbKajianHelper
import com.kajianid.android.databases.kajian.MappingHelper
import com.kajianid.android.databinding.ActivityShowKajianBinding
import com.kajianid.android.receiver.KajianAlarmReceiver
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch

```

```

import java.util.*

class ShowKajianSQLActivity: AppCompatActivity() {
    private lateinit var kajian: Kajian
    private var reminder = false
    private lateinit var dbKajianHelper: DbKajianHelper
    private lateinit var binding: ActivityShowKajianBinding

    companion object {
        const val EXTRA_PARCEL_KAJIAN = "extra_parcel_kajian"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
            ActivityShowKajianBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val alarmReceiver = KajianAlarmReceiver()

        binding.progressMessage.visibility = View.GONE
        binding.errorMessage.visibility = View.GONE

        val toolbar = findViewById<Toolbar>(R.id.toolbar)
        setSupportActionBar(toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        supportActionBar?.title = "Lihat Kajian"

        //mengirim data ke ShowKajianActivity dari
        KajianSQLAdapter
        val bundle = intent.extras!!
        kajian = bundle.getParcelable(EXTRA_PARCEL_KAJIAN)!!
        binding.contentShowKajian.tvKajianTitle.text =
            kajian.title
        binding.contentShowKajian.tvMosqueAddress.text =
            kajian.mosque
        binding.contentShowKajian.tvUstadzName.text =
            kajian.ustadzName
        val dateAnnounce =
            resources.getString(R.string.timestamp_announce) + " " +
            kajian.dateAnnounce
        val dateDue =
            resources.getString(R.string.timestamp_due) + " " + kajian.date
        binding.contentShowKajian.tvTimestampAnnounce.text =
            dateAnnounce
        binding.contentShowKajian.tvTimestampDue.text = dateDue
        binding.contentShowKajian.tvDescription.text =
            kajian.description
        Glide.with(this)
            .load(kajian.imgResource)
            .into(binding.contentShowKajian.imgThumbnail)

        //jika lokasi kajian di tempat

```



```

        if (kajian.place == "Di Tempat") {
            binding.contentShowKajian.btnPlay.visibility =
View.GONE
            binding.contentShowKajian.tvCategory.text =
kajian.place.toString()
            binding.contentShowKajian.tvMosqueAddress.text =
kajian.address
        } //jika lokasi kajian lewat online
        else {
            binding.contentShowKajian.btnPlay.visibility =
View.VISIBLE
            val category =
kajian.place.toString().toUpperCase(Locale.ROOT) + " - Courtesy
of YouTube"
            binding.contentShowKajian.tvCategory.text =
category
            binding.contentShowKajian.tvMosqueAddress.text =
kajian.mosque
            val uri = kajian.youtubelink.toString()

binding.contentShowKajian.btnPlay.setOnClickListener {
            val i = Intent(Intent.ACTION_VIEW,
Uri.parse(uri))
            startActivity(i)
        }
    }

    //mengaktifkan DbKajianHelper
    dbKajianHelper = DbKajianHelper.getInstance(this)
    dbKajianHelper.open()

    //GlobalScope digunakan untuk mengatur lifecycle
    coroutine mengikuti lifecycle aplikasi
    //launch memanggil proses Dispatchers.Main untuk
    menjalankan proses yang berhubungan dengan IO
    //CoroutineScope sebagai pengganti proses asynchronous
    (eksekusi kode tidak berurutan)
    GlobalScope.launch(Dispatchers.Main) {

        //Dispatchers.IO untuk menjalankan fungsi yang
        berhubungan dengan database
        val deferredKajian = async(Dispatchers.IO) {

            //mencari data kajian berdasarkan id di
            DbKajianHelper di dalam function queryById
            val cursor = kajian.id?.let {
            dbKajianHelper.queryById(it) }

            //jika data sudah ditemukan, maka akan masuk
            ke file MappingHelper untuk konversi data ke ArrayList sehingga
            data menjadi satu, tidak terpisah-pisah
            MappingHelper.mapCursorToArrayList(cursor)
        }
    }

```

```

        //menunggu apakah proses di variabel deferredKajian
sudah selesai atau belum
        val isReminder = deferredKajian.await()

        //jika sudah selesai data ukuran data 0 atau tidak
ada
        if (isReminder.size == 0){
            remindered = false

binding.fabreminder.setImageResource(R.drawable.ic_baseline_not
ifications_none_24)
        }else{
            remindered = true

binding.fabreminder.setImageResource(R.drawable.ic_baseline_not
ifications_active_24)
        }
    }

    binding.fabreminder.setOnClickListener{
        if (remindered){
            val alert = AlertDialog.Builder(this)

alert.setTitle(resources.getString(R.string.sure))

alert.setMessage(resources.getString(R.string.remove_reminder_c
onfirm))

alert.setPositiveButton(resources.getString(R.string.yes)) { _,
_ ->
            kajian.id?.let { it1 ->
dbKajianHelper.deleteById(it1) }
            remindered = false

binding.fabreminder.setImageResource(R.drawable.ic_baseline_not
ifications_none_24)
            alarmReceiver.cancelReminder(this,
kajian.id.toInt())
            finish()
        }

alert.setNegativeButton(resources.getString(R.string.no)) { _,
_ ->
            /* no-op */
        }
        alert.create().show()
    }else{
        val values = ContentValues()
        values.put(DatabaseContract.KajianColumns.ID,
kajian.id)
        values.put(DatabaseContract.KajianColumns.KAJIAN_TITLE,

```

```

kajian.title)

values.put(DatabaseContract.KajianColumns.USTADZ_NAME,
kajian.ustadzName)

values.put(DatabaseContract.KajianColumns.MOSQUE_NAME,
kajian.mosque)

values.put(DatabaseContract.KajianColumns.ADDRESS,
kajian.address)

values.put(DatabaseContract.KajianColumns.PLACE, kajian.place)

values.put(DatabaseContract.KajianColumns.YOUTUBE_LINK,
kajian.youtubelink)

values.put(DatabaseContract.KajianColumns.DESCRPTION,
kajian.description)

values.put(DatabaseContract.KajianColumns.IMG_RESOURCE,
kajian.imgResource)

values.put(DatabaseContract.KajianColumns.DATE_ANNOUNCE,
kajian.dateAnnounce)

values.put(DatabaseContract.KajianColumns.DATE_DUE,
kajian.date)

        dbKajianHelper.insert(values)
        reminded = true

binding.fabreminder.setImageResource(R.drawable.ic_baseline_not
ifications_active_24)
        Toast.makeText(this, "Kajian disimpan.",
Toast.LENGTH_LONG).show()
    }

    binding.pullToRefresh.setOnRefreshListener {
        binding.pullToRefresh.isRefreshing = false
    }
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_article_kajian, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean
{
    when(item.itemId) {
        android.R.id.home -> super.onBackPressed()
        R.id.share -> {
            val mimeType = "text/plain"

```

```

        ShareCompat.IntentBuilder.from(this).apply {
            setType(mimeType)
            setChooserTitle("Bagikan kajian ini
sekarang!")
            setText("""
Kajian: ${binding.contentShowKajian.tvKajianTitle.text}
Oleh: ${binding.contentShowKajian.tvUstadzName.text}
Waktu Pelaksanaan:
${binding.contentShowKajian.tvTimestampDue.text}

Kajian ini bersifat ${kajian.place}

Deskripsi:
${kajian.description}

Informasi ini disebarakan melalui aplikasi
${getString(R.string.app_name)}.
        """.trimIndent())
            startChooser()
        }
    }
}
return super.onOptionsItemSelected(item)
}
}

```

- ShowKajianViewModel.java

```

package com.kajianid.android.viewmodels;

import android.content.Context;
import android.os.Looper;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;
import com.kajianid.android.R;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import cz.msebera.android.httpclient.Header;

public class ShowKajianViewModel extends ViewModel {
    private MutableLiveData<Map<String, Object>> mData = new
MutableLiveData<>();
}

```

```

        public LiveData<Map<String, Object>> getKajian() {
            return mData;
        }

        public Map<String, Object> setKajian(Context context,
String id) {
            try {
                Looper.prepare();
            } catch (Exception ignored) {

            }

            Map<String, Object> resultMap = new HashMap<>();
            String url = context.getString(R.string.server) +
"api/kajian";
            SyncHttpClient client = new SyncHttpClient();

            RequestParams params = new RequestParams();
            params.put("read", "1");
            params.put("ustadz_mode", "0");
            params.put("kajian", id);

            client.get(url, params, new AsyncHttpResponseHandler()
{
                @Override
                public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
                    try {
                        String result = new String(responseBody);
                        JSONObject responseObject = new
JSONObject(result);
                        JSONArray list =
responseObject.getJSONArray("data");

                        JSONObject kajianJSONObject =
list.getJSONObject(0);
                        String title =
kajianJSONObject.getString("kajian_title");
                        String ustadzName =
kajianJSONObject.getString("ustadz_name");
                        String mosqueId =
kajianJSONObject.getString("mosque_id");
                        String mosqueName =
kajianJSONObject.getString("mosque_name");
                        String address =
kajianJSONObject.getString("address");
                        String category =
kajianJSONObject.getString("place");
                        String youtubeLink =
kajianJSONObject.getString("youtube_link");
                        String description =
kajianJSONObject.getString("description");
                        String imgResource =
kajianJSONObject.getString("img_resource");

```

```

        String dateAnnounce =
kajianJSONObject.getString("date_announce");
        String dateDue =
kajianJSONObject.getString("date_due");

        DateFormat dateAnnounced = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        String dateAnnounceFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateAnnounced.parse(dateAnnounce));

        DateFormat dateDueD = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDueD.parse(dateDue));

        resultMap.put("status", true);
        resultMap.put("id", id);
        resultMap.put("title", title);
        resultMap.put("ustadz_name", ustadzName);
        resultMap.put("mosque_id", mosqueId);
        resultMap.put("mosque_name", mosqueName);
        resultMap.put("address", address);
        resultMap.put("category", category);
        resultMap.put("youtube_link", youtubeLink);
        resultMap.put("description", description);
        resultMap.put("img_resource", imgResource);
        resultMap.put("date_announce",
dateAnnounceFormatted);
        resultMap.put("date_due",
dateDueFormatted);
        resultMap.put("date_due_unformatted",
dateDue);

        mData.postValue(resultMap);
    } catch (JSONException | ParseException e) {
        resultMap.put("status", false);
        resultMap.put("code", 0);
        resultMap.put("message",
e.getLocalizedMessage() + " id: " + id);
        mData.postValue(resultMap);
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
    resultMap.put("status", false);
    resultMap.put("code", statusCode);
    resultMap.put("message",
error.getLocalizedMessage() + " id: " + id);
    mData.postValue(resultMap);
}

```

```

    }
    });

    return resultMap;
}

public Map<String, Object> setKajianAsync(Context context,
String id) {
    Map<String, Object> resultMap = new HashMap<>();
    String url = context.getString(R.string.server) +
"api/kajian";
    AsyncHttpClient client = new AsyncHttpClient();

    RequestParams params = new RequestParams();
    params.put("read", "1");
    params.put("ustadz_mode", "0");
    params.put("kajian", id);

    client.get(url, params, new AsyncHttpResponseHandler()
{
    @Override
    public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
        try {
            String result = new String(responseBody);
            JSONObject responseObject = new
JSONObject(result);
            JSONArray list =
responseObject.getJSONArray("data");

            JSONObject kajianJSONObject =
list.getJSONObject(0);
            String title =
kajianJSONObject.getString("kajian_title");
            String ustadzName =
kajianJSONObject.getString("ustadz_name");
            String mosqueId =
kajianJSONObject.getString("mosque_id");
            String mosqueName =
kajianJSONObject.getString("mosque_name");
            String address =
kajianJSONObject.getString("address");
            String category =
kajianJSONObject.getString("place");
            String youtubeLink =
kajianJSONObject.getString("youtube_link");
            String description =
kajianJSONObject.getString("description");
            String imgResource =
kajianJSONObject.getString("img_resource");
            String dateAnnounce =
kajianJSONObject.getString("date_announce");
            String dateDue =

```

```

kajianJSONObject.getString("date_due");

        DateFormat dateAnnounced = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        assert dateAnnounced != null;
        String dateAnnounceFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateAnnounced.parse(dateAnnounce));

        DateFormat dateDueD = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        assert dateDueD != null;
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDueD.parse(dateDue));

        resultMap.put("status", true);
        resultMap.put("id", id);
        resultMap.put("title", title);
        resultMap.put("ustadz_name", ustadzName);
        resultMap.put("mosque_id", mosqueId);
        resultMap.put("mosque_name", mosqueName);
        resultMap.put("address", address);
        resultMap.put("category", category);
        resultMap.put("youtube_link", youtubeLink);
        resultMap.put("description", description);
        resultMap.put("img_resource", imgResource);
        resultMap.put("date_announce",
dateAnnounceFormatted);
        resultMap.put("date_due",
dateDueFormatted);
        resultMap.put("date_due_unformatted",
dateDue);

        mData.postValue(resultMap);
    } catch (JSONException | ParseException e) {
        resultMap.put("status", false);
        resultMap.put("code", 0);
        resultMap.put("message",
e.getLocalizedMessage() + " id: " + id);
        mData.postValue(resultMap);
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
    resultMap.put("status", false);
    resultMap.put("code", statusCode);
    resultMap.put("message",
error.getLocalizedMessage() + " id: " + id);
    mData.postValue(resultMap);
}
}

```



```

        });

        return resultMap;
    }
}

```

#### - SavedKajianFragment.kt

```

package com.kajianid.android.fragment

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.android.adapter.KajianSQLAdapter
import com.kajianid.android.databases.kajian.DbKajianHelper
import com.kajianid.android.databases.kajian.MappingHelper
import
com.kajianid.android.databinding.FragmentSavedKajianBinding
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch

class SavedKajianFragment : Fragment() {
    private lateinit var kajianSQLAdapter: KajianSQLAdapter
    private lateinit var dbKajianHelper: DbKajianHelper
    private lateinit var binding: FragmentSavedKajianBinding

    override fun onCreateView(view: View, savedInstanceState:
Bundle?) {
        super.onCreateView(view, savedInstanceState)

        binding.rvKajian.layoutManager =
LinearLayoutManager(context)
        binding.rvKajian.adapter = kajianSQLAdapter
        binding.rvKajian.setHasFixedSize(true) //tidak efek
ketika diputar

        dbKajianHelper =
DbKajianHelper.getInstance(requireContext().applicationContext)
        dbKajianHelper.open()

        loadKajian()
    }

    private fun loadKajian() {
        GlobalScope.launch ( Dispatchers.Main ) {
            val deferredKajian = async (Dispatchers.IO){
                val cursor = dbKajianHelper.queryAll()
            }
        }
    }
}

```

```

        MappingHelper.mapCursorToArrayList(cursor)
    }
    val kajian = deferredKajian.await()
    if (kajian.size
>0) kajianSQLAdapter.setKajian(kajian)
    else{
        kajianSQLAdapter.setKajian(ArrayList())
    }
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    kajianSQLAdapter = KajianSQLAdapter()
}

override fun onCreateView(inflater: LayoutInflater,
container: ViewGroup?,
                                savedInstanceState: Bundle?):
View {
    // Inflate the layout for this fragment
    binding = FragmentSavedKajianBinding.inflate(inflater,
container, false)
    return binding.root
}

override fun onResume() {
    super.onResume()
    loadKajian()
}
}

```

- **KajianAlarmReceiver.kt**

```

package com.kajianid.android.receiver

import android.app.AlarmManager
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.media.RingtoneManager
import android.os.Build
import android.util.Log
import android.widget.Toast
import androidx.core.app.NotificationCompat
import androidx.core.content.ContextCompat
import com.kajianid.android.R
import java.text.ParseException
import java.text.SimpleDateFormat
import java.util.*

class KajianAlarmReceiver : BroadcastReceiver() {

```

```

companion object {
    const val EXTRA_KAJIAN_TITLE = "extra_kajian_title"
    const val EXTRA_USTADZ_NAME = "extra_ustadz_name"

    private const val CHANNEL_ID = "Channel_1"
    private const val CHANNEL_NAME = "AlarmManager channel"

    private const val DATE_FORMAT = "yyyy-MM-dd HH:mm:ss"
    private const val NOTIFICATION_ID = 100
}

override fun onReceive(context: Context, intent: Intent) {
    // method yang akan dipanggil saat jam sudah
    menunjukkan
    // sesuai pada tanggal dan waktu yang ditentukan
    val title =
intent.getStringExtra(EXTRA_KAJIAN_TITLE).toString()
    val ustadzName =
intent.getStringExtra(EXTRA_USTADZ_NAME).toString()

    showReminderNotification(context, title, ustadzName)
}

fun setReminder(context: Context, id: Int, title: String,
dateUnformatted: String, ustadzName: String) {
    if (isDateInvalid(dateUnformatted)) return

    // cek tanggal apakah sudah outdated atau belum
    val dateCheck = SimpleDateFormat(DATE_FORMAT,
Locale.getDefault()).parse(dateUnformatted)!!
    if (System.currentTimeMillis() > dateCheck.time) {
        Toast.makeText(context, "Jadwal Kajian ini sudah
usang. Kami hanya akan menyimpan kajian ini ke dalam \"Kajian
Tersimpan\".", Toast.LENGTH_SHORT).show()
        return
    }

    val alarmManager =
context.getSystemService(Context.ALARM_SERVICE) as AlarmManager
    val intent = Intent(context,
KajianAlarmReceiver::class.java)

    intent.putExtra(EXTRA_KAJIAN_TITLE, title)
    intent.putExtra(EXTRA_USTADZ_NAME, ustadzName)

    Log.e("setReminder", "$dateUnformatted: $title by
$ustadzName")

    val dateTimeArray = dateUnformatted.split("
").toTypedArray()

    val dateArray = dateTimeArray[0].split("-

```

```

    ).toTypedArray()
    val timeArray =
dateTimeArray[1].split(":").toTypedArray()

    val calendar = Calendar.getInstance()
    calendar.set(Calendar.YEAR,
Integer.parseInt(dateArray[0]))
    calendar.set(Calendar.MONTH,
Integer.parseInt(dateArray[1]) - 1)
    calendar.set(Calendar.DAY_OF_MONTH,
Integer.parseInt(dateArray[2]))

    calendar.set(Calendar.HOUR_OF_DAY,
Integer.parseInt(timeArray[0]))
    calendar.set(Calendar.MINUTE,
Integer.parseInt(timeArray[1]) - 31)
    calendar.set(Calendar.SECOND, 0)

    val dateString = SimpleDateFormat("dd MMM yyyy
HH:mm:ss", Locale.getDefault()).format(calendar.time)

    val pendingIntent = PendingIntent.getBroadcast(context,
id, intent, 0)
    alarmManager.set(AlarmManager.RTC_WAKEUP,
calendar.timeInMillis, pendingIntent)

    Toast.makeText(context, "Kajian $title akan kami
ingatkan pada $dateString!", Toast.LENGTH_SHORT).show()
}

fun cancelReminder(context: Context, id: Int) {
    val alarmManager =
context.getSystemService(Context.ALARM_SERVICE) as AlarmManager
    val intent = Intent(context,
KajianAlarmReceiver::class.java)

    val pendingIntent = PendingIntent.getBroadcast(context,
id, intent, 0)
    pendingIntent.cancel()

    alarmManager.cancel(pendingIntent)

    Toast.makeText(context, "Peningkat pada kajian ini
berhasil dibatalkan!", Toast.LENGTH_SHORT).show()
}

private fun showReminderNotification(context: Context,
title: String, ustadzName: String) {
    val contentTitle =
context.getString(R.string.notification_half_title, title)
    val contentMessage =
context.getString(R.string.notification_half_message, title,
ustadzName)

```

```

        val notificationManagerCompat =
context.getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
        val alarmSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION
)
        val builder = NotificationCompat.Builder(context,
CHANNEL_ID)
                .setSmallIcon(R.drawable.icon)
                .setContentTitle(contentTitle)
                .setContentText(contentMessage)
                .setColor(ContextCompat.getColor(context,
android.R.color.transparent))
                .setVibrate(longArrayOf(1000, 1000, 1000, 1000,
1000))
                .setSound(alarmSound)

        /*
        Untuk android Oreo ke atas perlu menambahkan
notification channel
        */
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            /* Create or update. */
            val channel = NotificationChannel(CHANNEL_ID,
CHANNEL_NAME,
NotificationManager.IMPORTANCE_DEFAULT)

            channel.enableVibration(true)
            channel.vibrationPattern = longArrayOf(1000, 1000,
1000, 1000, 1000)

            builder.setChannelId(CHANNEL_ID)

notificationManagerCompat.createNotificationChannel(channel)
        }

        val notification = builder.build()
        notificationManagerCompat.notify(NOTIFICATION_ID,
notification)
    }

    private fun isDateInvalid(date: String): Boolean {
        return try {
            val df = SimpleDateFormat(
DATE_FORMAT,
Locale.getDefault())
            df.isLenient = false
            df.parse(date)
            false
        } catch (e: ParseException) {
            true
        }
    }

```

```
}  
}
```

f. Menu About

a) Tampilan

- activity\_about.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=".activities.AboutActivity">  
  
  <com.google.android.material.appbar.AppBarLayout  
    android:id="@+id/appBar"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:elevation="0dp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent">  
  
    <androidx.appcompat.widget.Toolbar  
      android:id="@+id/toolbar"  
      android:layout_width="match_parent"  
      android:layout_height="wrap_content" />  
  
    </com.google.android.material.appbar.AppBarLayout>  
  
    <RelativeLayout  
      android:id="@+id/rlAppName"  
      android:layout_width="match_parent"  
      android:layout_height="wrap_content"  
      android:padding="8dp">  
  
        <ImageView  
          android:id="@+id/appIcon"  
          android:layout_width="60dp"  
          android:layout_height="60dp"  
          android:src="@drawable/icon" />  
  
        <TextView  
          android:id="@+id/appName"  
          android:layout_width="wrap_content"  
          android:layout_height="wrap_content"  
          android:layout_marginStart="8dp"  
          android:layout_toEndOf="@+id/appIcon"  
          android:text="@string/app_name"  
          android:textColor="?android:textColorPrimary"
```

```

        android:textSize="18sp"
        android:textStyle="bold" />

        <TextView
            android:id="@+id/appVer"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/appName"
            android:layout_marginStart="8dp"
            android:layout_toEndOf="@+id/appIcon"
            android:text="@string/app_version"
            android:textSize="14sp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/appVer"
            android:layout_marginStart="8dp"
            android:layout_toEndOf="@+id/appIcon"
            android:text="@string/copyright"
            android:textSize="14sp" />

    </RelativeLayout>

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/rlAppName"
        app:tabIndicatorAnimationDuration="50"
        app:tabIndicatorColor="#001FBD"
        app:tabSelectedTextColor="#001FBD" />
    <!-- app:tabIndicatorAnimationDuration="50"-->
    <!-- app:tabIndicatorColor="#001FBD"-->
    <!-- app:tabSelectedTextColor="#001FBD"-->
    <!-- android:background="?android:attr/colorPrimary"-->

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/vpAbout"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>

```

- menu\_about.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/menuGithub"
        android:title="@string/github"

```

```

        app:showAsAction="always" />
    </menu>

```

- fragment\_developers.xml

```

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:orientation="vertical">

        <de.hdodenhof.circleimageview.CircleImageView
            android:id="@+id/civPhoto"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:layout_gravity="center"
            android:layout_margin="16dp"
            android:elevation="1dp" />

        <TextView
            android:id="@+id/tvName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/example_about_name"
            android:textColor="?android:textColorPrimary"
            android:textSize="20sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/tvNim"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/example_about_nim" />

        <TableLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp">

            <TableRow
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginBottom="8dp">

                <TextView

```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="16dp"
        android:text="@string/label_about_grade" />

        <TextView
            android:id="@+id/tvGrade"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/example_about_grade"
            android:textAlignment="textEnd"

android:textColor="?android:textColorPrimary" />

    </TableRow>

    <TableRow

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="8dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"

android:text="@string/label_about_year_registered" />

        <TextView
            android:id="@+id/tvYearRegistered"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"

android:text="@string/example_about_year_registered"
            android:textAlignment="textEnd"

android:textColor="?android:textColorPrimary" />

    </TableRow>

    <TableRow

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="8dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"
            android:text="@string/label_about_github"

/>

```

```

        <TextView
            android:id="@+id/tvGitHub"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/example_about_github"
            android:textAlignment="textEnd"

android:textColor="?android:textColorPrimary" />

    </TableRow>

    <TableRow

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginBottom="8dp">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginEnd="16dp"
                    android:text="@string/label_about_email" />

                <TextView
                    android:id="@+id/tvEmail"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:autoLink="email"
                    android:text="@string/example_about_email"
                    android:textAlignment="textEnd"

android:textColor="?android:textColorPrimary"

android:textColorLink="?android:textColorPrimary" />

            </TableRow>

            <TableRow

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginEnd="16dp"
                    android:text="@string/label_about_address"

/>

                <TextView
                    android:id="@+id/tvAddress"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"

        android:text="@string/example_about_address"
        android:textAlignment="textEnd"

        android:textColor="?android:textColorPrimary" />

    </TableRow>

</TableLayout>

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnVisit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="36dp"
        android:text="@string/visit_my_github" />

</LinearLayout>

</ScrollView>

```

## b) Program

- AboutActivity.java

```

package com.kajianid.android.activities;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.kajianid.android.R;
import com.kajianid.android.databinding.ActivityAboutBinding;
import com.kajianid.android.pageradapter.AboutPagerAdapter;
import java.util.Objects;

public class AboutActivity extends AppCompatActivity {
    private ActivityAboutBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityAboutBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        AboutPagerAdapter adapter = new AboutPagerAdapter(this,
getSupportFragmentManager());

```

```

        binding.vpAbout.setAdapter(adapter);
        binding.tabs.setupWithViewPager(binding.vpAbout);

        setSupportActionBar(binding.toolbar);

Objects.requireNonNull(getSupportActionBar()).setTitle(getString(R.string.about));
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

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

    @SuppressWarnings("NonConstantResourceId")
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem
item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                super.onBackPressed();
                break;
            case R.id.menuGithub:
                Intent i = new Intent(Intent.ACTION_VIEW);
                i.setData(Uri.parse("https://github.com/IF18A-Kajian-ID"));
                startActivity(i);
                break;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        binding = null;
    }
}

```

#### - DevelopersFragment.java

```

package com.kajianid.android.fragment;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import com.bumptech.glide.Glide;

```

```

import
com.kajianid.android.databinding.FragmentDevelopersBinding;
import com.kajianid.android.utils.DeveloperData;

public class DevelopersFragment extends Fragment {
    private static final String ARG_DEVELOPER_POSITION =
"arg_developer_position";
    private FragmentDevelopersBinding binding;

    public DevelopersFragment() {
        // Required empty public constructor
    }

    public static DevelopersFragment newInstance(int position)
{
        DevelopersFragment fragment = new DevelopersFragment();
        Bundle args = new Bundle();

        args.putInt(ARG_DEVELOPER_POSITION, position);

        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
ViewGroup container,
                                Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        binding = FragmentDevelopersBinding.inflate(inflater);
        return binding.getRoot();
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable
Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        int position = 1;
        if (getArguments() != null) {
            position =
requireArguments().getInt(ARG_DEVELOPER_POSITION, 1);
        }

        Glide.with(this)
            .load(DeveloperData.drawableId[position])
            .override(150, 150)
            .centerCrop()

```

```

        .into(binding.civPhoto);

        binding.tvName.setText(DeveloperData.name[position]);
        binding.tvNim.setText(DeveloperData.nim[position]);
        binding.tvGrade.setText(DeveloperData.kelas[position]);

        binding.tvYearRegistered.setText(DeveloperData.yearRegister[position]);

        binding.tvGitHuB.setText(DeveloperData.githubUsername[position]);

        binding.tvEmail.setText(DeveloperData.email[position]);

        binding.tvAddress.setText(DeveloperData.address[position]);

        String githubUrl = "https://github.com/" +
DeveloperData.githubUsername[position];
        binding.btnVisit.setOnClickListener(itView -> {
            Intent i = new Intent(Intent.ACTION_VIEW);
            i.setData(Uri.parse(githubUrl));
            startActivity(i);
        });
    }
}

```

#### - AboutPagerAdapter.java

```

package com.kajianid.android.pageradapter;

import android.content.Context;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.StringRes;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;
import com.kajianid.android.R;
import com.kajianid.android.fragment.DevelopersFragment;

public class AboutPagerAdapter extends FragmentPagerAdapter {
    private final Context context;

    @StringRes
    private final int[] tabTitles = {
        R.string.leader,
        R.string.member_one,
        R.string.member_two,
        R.string.member_three,
        R.string.member_four
    };

    public AboutPagerAdapter(Context context, FragmentManager fm) {
        super(fm, BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
        this.context = context;
    }
}

```

```

    }

    @NonNull
    @Override
    public Fragment getItem(int position) {
        return DevelopersFragment.newInstance(position);
    }

    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        return
context.getResources().getString(tabTitles[position]);
    }

    @Override
    public int getCount() {
        return tabTitles.length;
    }
}

```

- DeveloperData.java

```

package com.kajianid.android.utils;

import androidx.annotation.DrawableRes;
import com.kajianid.android.R;

public class DeveloperData {
    @DrawableRes
    public static final int[] drawableId = {
        R.drawable.dwichan,
        R.drawable.maratun,
        R.drawable.fendi,
        R.drawable.diyah,
        R.drawable.bayu
    };

    public static final String[] name = {
        "Dwi Candra Permana",
        "Mar Atun Solehah",
        "Fendi Riawan",
        "Diyah Primasari",
        "Muhammad Rizky Bayu Pratama",
    };

    public static final String[] nim = {
        "18.11.0004",
        "18.11.0017",
        "18.11.0268",
        "18.11.0289",
        "18.11.0258",
    };

    public static final String[] kelas = {

```

```

        "IF 2018 A",
        "IF 2018 A",
        "IF 2018 A",
        "IF 2018 A",
        "IF 2018 A",
    };

    public static final String[] yearRegister = {
        "2018",
        "2018",
        "2018",
        "2018",
        "2018",
    };

    public static final String[] githubUsername = {
        "dwichan0905",
        "maratun1725",
        "fendi-riawan",
        "DiyahPrimasari0289",
        "rizkybayu1411",
    };

    public static final String[] email = {
        "dwichan@outlook.com",
        "marhatunsolehal7@gmail.com",
        "fendirriawan0102@gmail.com",
        "diyahprimasari24@gmail.com",
        "rizkybayu1411@gmail.com",
    };

    public static final String[] address = {
        "Jalan Mandor Martinem Roy 1 Nomor 99 RT 005 RW 006",
        "Desa Rejasari Kecamatan Langensari Kota Banjar",
        "Desa losari RT 04/RW 04 Kecamatan Rawalo Kabupaten",
        "Banyumas",
        "Kemutug Lor RT 04 rw 03 Kecamatan Baturraden",
        "Kasih, RT 004 RW 003, Kecamatan Kertanegara,",
        "Kabupaten Purbalingga",
        "Desa Papringan Kec. Banyumas, Kabupaten Banyumas,",
        "Jawa Tengah",
    };
}

```

#### g. arrays.xml

```

<resources>
    <!-- Reply Preference -->
    <string-array name="theme_entries">
        <item>Default (mengikuti sistem)</item>
        <item>Terang</item>
        <item>Gelap</item>
    </string-array>

```



```

<string-array name="theme_values">
    <item>auto</item>
    <item>light</item>
    <item>dark</item>
</string-array>

<string-array name="map_type_entries">
    <item>Satellite</item>
    <item>Roadmap (default)</item>
    <item>Terrain</item>
</string-array>

<string-array name="map_type_values">
    <item>4</item>
    <item>1</item>
    <item>3</item>
</string-array>

<string-array name="ajmlKolom">
    <item>Silahkan Pilih Jumlah Kolom</item>
    <item>1</item>
    <item>2</item>
</string-array>

<string-array name="ajeniszakat">
    <item>Beras</item>
    <item>Uang</item>
</string-array>
</resources>

```

#### h. colors.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>

    <color name="colorPrimary">#6200EE</color>
    <!-- <color name="colorPrimaryDark">#3700B3</color>-->
    <color name="transparent">#80FFFFFF</color>
    <color name="colorAccent">#03DAC5</color>
    <color name="Judul">#000000</color>
    <color name="putih">#F6F5F5</color>
    <color name="hitm">#000000</color>
    <color name="warna_tulisan_button">#FFFCFC</color>
    <color name="warna_background_button">#3588CA</color>
    <color name="hitam">#000000</color>
    <color name="biru">#001297</color>
    <color name="abuabu">#5A5A5A</color>
</resources>

```

i. **dimens.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="nav_header_vertical_spacing">8dp</dimen>
    <dimen name="nav_header_height">176dp</dimen>
    <dimen name="fab_margin">16dp</dimen>
    <dimen name="app_bar_height">180dp</dimen>
    <dimen name="etLebar">396dp</dimen>
    <dimen name="etTinggi">51dp</dimen>
</resources>
```

j. **drawables.xml**

```
<resources xmlns:android="http://schemas.android.com/apk/res/android">
    <item name="ic_menu_camera"
type="drawable">@android:drawable/ic_menu_camera</item>
    <item name="ic_menu_gallery"
type="drawable">@android:drawable/ic_menu_gallery</item>
    <item name="ic_menu_slideshow"
type="drawable">@android:drawable/ic_menu_slideshow</item>
    <item name="ic_menu_manage"
type="drawable">@android:drawable/ic_menu_manage</item>
    <item name="ic_menu_share"
type="drawable">@android:drawable/ic_menu_share</item>
    <item name="ic_menu_send"
type="drawable">@android:drawable/ic_menu_send</item>
</resources>
```

k. **strings.xml**

```
<resources>
    <string name="app_name">Kajian.ID</string>
    <string name="app_version">1.0 (2020)</string>
    <string name="title_activity_main">Beranda</string>
    <string name="navigation_drawer_open">Open navigation
drawer</string>
    <string name="navigation_drawer_close">Close navigation
drawer</string>
    <string name="nav_header_title">John Smith</string>
    <string
name="nav_header_subtitle">john.smith@example.com</string>
    <string name="nav_header_desc">NAVIGASI</string>
    <string name="action_settings">Pengaturan</string>

    <string name="menu_home">Beranda</string>
    <string name="menu_settings">Pengaturan</string>
    <string name="menu_downloaded_articles">Artikel Terunduh</string>
    <string name="menu_saved_kajian">Kajian Tersimpan</string>
    <string name="title_activity_settings">Pengaturan</string>

    <!-- Preference Titles -->
    <string name="display_header">Tampilan</string>
    <string name="sahur_header">Alarm Sahur</string>

    <!-- Messages Preferences -->
```

```

<string name="theme_title">Tema</string>

<!-- Sync Preferences -->
<string name="duration_imsak">Durasi sebelum imsak
(menit)</string>
<string name="duration_subtitle">Jeda waktu untuk alarm sebelum
sahur</string>
<string name="alarm_message">Catatan: Suara alarm akan mengikuti
suara bawaan dari smarphone Anda.</string>
<string name="location_title">Lokasi</string>
<string name="title_activity_maps_result">Titik
Koordinat</string>
<string name="title_activity_maps">Titik Koordinat</string>
<string name="defaultLocation_title">Atur Lokasi Utama</string>
<string name="sahurStatus_title">Aktifkan Alarm Sahur</string>
<string name="loc_succes_change">"Lokasi Utama berhasil diubah ke
"</string>
<string name="warning_zakat">PERHATIAN !\n \n Zakat yang
dikeluarkan per anggota keluarga adalah yang \n memenuhi syarat
sebagai berikut : \n 1. Beragama Islam\n 2. Keadaan Merdeka (Tidak
Menjadi Budak)\n 3. Baligh dan Berakal\n \n Zakat Fitrah ini wajib
dibayarkan seorang Muslim Setahun Sekali pada Awal atau Akhir Bulan
Ramadhan.</string>

<string name="app_intro_skip_button">Lewati</string>
<string name="app_intro_next_button">Lanjut</string>
<string name="app_intro_back_button">Kembali</string>
<string name="app_intro_done_button">Selesai</string>
<string name="your_location">Lokasimu:</string>
<string name="example_title">Untukmu yang Sedang Mencari
Ketenangan Jiwa</string>
<string name="join_with_shoumhome">Ayo bergabung bersama Shoum
Home!</string>
<string name="example_ustads">Ustadz K H Abdul Somad</string>
<string name="example_keterangan">LIVE</string>
<string name="example_tanggal">1/02/2020 18.00</string>

<string name="load_data">Mohon menunggu, kami sedang memuat
data...</string>
<string name="warning">warning</string>
<string name="example_error_message">Oops! Error\n\n[404] Not
Found</string>
<string name="refresh">Segarkan</string>
<string name="example_article_title">Hukum memberikan zakat
kepada kaum dhuafa secara online</string>
<string name="likes">suka</string>
<string name="example_likes">1200 suka</string>
<string name="post_date">Post Date</string>
<string name="example_post_date">12 Jan 2020 04:10:15</string>
<string name="example_text">Lorem ipsum dolor sit amet,
consectetur adipisicing elit. Hapunten salajengna</string>

<string name="play_video">Play Video</string>

```

```

    <string name="example_kajian_title">Untukmu yang sedang dalam
pelukan</string>
    <string name="example_ustadz">Ust. Fendi Riawan, LLC.,
MA.</string>
    <string name="example_mosque">Masjid Fatimatuzzahra</string>
    <string name="on_stage">Di Tempat</string>
    <string name="timestamp_announce">Waktu Pengumuman</string>
    <string name="timestamp_due">Waktu Pelaksanaan</string>
    <string name="description">Deskripsi:</string>
    <string name="by">oleh</string>
    <string name="post_picture">Post Picture</string>
    <string name="thumbnail">Thumbnail</string>
    <string name="read_article">Artikel Islami</string>
    <string
name="server">https://dwichan.intermediaamikom.org/</string>
    <string name="show_kajian">Kajian</string>

    <string name="leader">KETUA</string>
    <string name="member_one">ANGGOTA 1</string>
    <string name="member_two">ANGGOTA 2</string>
    <string name="about">Tentang</string>
    <string name="copyright">© 2020. Kajian.ID Team.</string>

    <string name="example_about_name">John Smith</string>
    <string name="example_about_nim">12.11.1234</string>
    <string name="label_about_grade">Kelas</string>
    <string name="example_about_grade">AB 1234 C</string>
    <string name="label_about_year_registered">Tahun
Angkatan</string>
    <string name="example_about_year_registered">2013</string>
    <string name="label_about_github">GitHub Username</string>
    <string name="example_about_github">admin123</string>
    <string name="label_about_email">Email</string>
    <string
name="example_about_email">john.smith@example.com</string>
    <string name="label_about_address">Alamat</string>
    <string name="example_about_address">1234 William Street, CA,
United States 12204</string>
    <string name="visit_my_github">Kunjungi Profil GitHub
Saya</string>
    <string name="github">GitHub</string>
    <string name="mosque_location">Lokasi Masjid</string>
    <string name="mosque_list">Daftar Masjid</string>
    <string name="nama_ustadz">ustadz abdul somad</string>
    <string name="sure">Yakin?</string>
    <string name="yes">Ya</string>
    <string name="no">Tidak</string>
    <string name="remove_reminder_confirm">Yakin ingin mematikan
pengingat dari kajian ini?</string>
    <string name="remove_article_confirm">Yakin ingin menghapus
artikel ini?</string>
    <string name="search">Cari</string>
    <string name="search_article">Cari artikel berdasarkan

```

```

judul/ustadz...</string>
    <string name="search_kajian">Cari kajian...</string>
    <string name="food">Makan</string>
    <string name="article">Artikel</string>
    <string name="kajian">Kajian</string>
    <string name="calc">Kalkulator</string>
    <string name="google_maps">Google Maps</string>
    <string name="choose_action_mosque">Pilih aksi yang akan
dilakukan dengan masjid ini.</string>
    <string name="view_on_maps">Lihat di Google Maps</string>
    <string name="cancel">Batalkan</string>
    <string name="example_address">Jalan Jatisari 13</string>
    <string name="search_mosque">Cari masjid...</string>
    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <string name="member_three">ANGGOTA 3</string>
    <string name="member_four">ANGGOTA 4</string>
    <string name="logo">LOGO</string>
    <string name="versinya">Version</string>
    <string name="share">Bagikan</string>

    <string name="notification_half_title">Sebentar lagi kajian %s
akan dimulai!</string>
    <string name="notification_half_message">Bersiap-siaplah!
Sebentar lagi, kajian %s yang dibawakan %s akan dimulai dalam waktu
30 menit ke depan!</string>
</resources>

```

## 1. Tema

### - themes.xml

```

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.KajianID"
parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">#FFC107</item>
        <item name="colorPrimaryVariant">#FFC107</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">#FFCC80</item>
        <item name="colorSecondaryVariant">#F57C00</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="1">?attr/colorPrimary</item>
        <!-- Customize your theme here. -->
    </style>
</resources>

```

### - themes.xml (night)

```

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.KajianID"

```

```

parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">#FFF59D</item>
    <item name="colorPrimaryVariant">#FBC02D</item>
    <item name="colorOnPrimary">@color/black</item>
    <!-- Secondary brand color. -->
    <item name="colorSecondary">#FFCC80</item>
    <item name="colorSecondaryVariant">#FFCC80</item>
    <item name="colorOnSecondary">@color/black</item>
    <!-- Status bar color. -->
    <item name="android:statusBarColor"
tools:targetApi="1">?attr/colorPrimary</item>
    <!-- Customize your theme here. -->
</style>
</resources>

```

#### m. AndroidManifest.xml

```

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

    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.VIBRATE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:roundIcon="@drawable/icon"
        android:supportsRtl="true"
        android:theme="@style/Theme.KajianID">
        <receiver
            android:name=".receiver.KajianAlarmReceiver"
            android:enabled="true"
            android:exported="true" />

        <activity android:name=".activities.AboutActivity" />
        <activity android:name=".activities.ArticleIslamiActivity" />
        <activity android:name=".activities.KajianActivity" />
        <activity android:name=".activities.MainActivity" />
        <activity android:name=".activities.MapResultActivity" />
        <activity android:name=".activities.MosqueLocationActivity"
/>

```

```

        <activity android:name=".activities.ReadArticleActivity" />
        <activity android:name=".activities.ReadArticleSQLActivity"
/>

        <activity android:name=".activities.ShowKajianActivity" />
        <activity android:name=".activities.ShowKajianSQLActivity" />

        <uses-library
            android:name="org.apache.http.legacy"
            android:required="false" />

        <activity android:name=".activities.SplashActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

n. build.gradle (Module)

```

plugins {
    id 'com.android.application'
    id 'kotlin-android'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.2"

    defaultConfig {
        applicationId "com.kajianid.android"
        minSdkVersion 21
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        multiDexEnabled true

        testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
    }
}

```

```

        targetCompatibility JavaVersion.VERSION_1_8
    }
    buildFeatures {
        viewBinding true
    }
}

dependencies {
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation
'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.navigation:navigation-fragment-
ktx:2.3.2'
    implementation 'androidx.navigation:navigation-ui-ktx:2.3.2'
    testImplementation 'junit:junit:4.13.1'
    implementation 'com.loopj.android:android-async-http:1.4.11'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-
core:3.3.0'
    implementation "androidx.core:core-ktx:1.3.2"
    implementation 'org.jetbrains.kotlin:kotlinx-coroutines-
android:1.3.9'
    implementation "org.jetbrains.kotlin:kotlin-stdlib-
jdk7:$kotlin_version"
    implementation 'com.github.bumptech.glide:glide:4.11.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'
    implementation 'de.hdodenhof:circleimageview:3.1.0'
    implementation 'com.google.android.gms:play-services-maps:17.0.0'
    implementation 'com.google.android.gms:play-services-
places:17.0.0'
    implementation 'com.google.android.gms:play-services-
location:17.0.0'
    implementation 'com.android.support:multidex:1.0.3'
}
repositories {
    mavenCentral()
}

```

### C. KAJIAN.ID USTADZ LAYOUT

```

- activity_about
- <?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="ui.about.AboutActivity">

        <com.google.android.material.appbar.AppBarLayout
            android:id="@+id/appBar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:elevation="0dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

        </com.google.android.material.appbar.AppBarLayout>

        <RelativeLayout
            android:id="@+id/rlAppName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="8dp">

            <ImageView
                android:id="@+id/appIcon"
                android:layout_width="60dp"
                android:layout_height="60dp"
                android:src="@drawable/icon" />

            <TextView
                android:id="@+id/appName"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"
                android:layout_toEndOf="@+id/appIcon"
                android:text="@string/app_name"
                android:textColor="?android:textColorPrimary"
                android:textSize="18sp"
                android:textStyle="bold" />

            <TextView
                android:id="@+id/appVer"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/appName"
                android:layout_marginStart="8dp"
                android:layout_toEndOf="@+id/appIcon"
                android:text="@string/app_version"
                android:textSize="14sp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/appVer"
                android:layout_marginStart="8dp"
                android:layout_toEndOf="@+id/appIcon"
                android:text="@string/copyright"
                android:textSize="14sp" />

```

```

</RelativeLayout>

<com.google.android.material.tabs.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/rlAppName"
    app:tabIndicatorAnimationDuration="50"
    app:tabIndicatorColor="#001FBD"
    app:tabSelectedTextColor="#001FBD" />
<!-- app:tabIndicatorAnimationDuration="50"-->
<!-- app:tabIndicatorColor="#001FBD"-->
<!-- app:tabSelectedTextColor="#001FBD"-->
<!-- android:background="?android:attr/colorPrimary"-->

<androidx.viewpager.widget.ViewPager
    android:id="@+id/vpAbout"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</LinearLayout>

```

- activity\_add\_update\_article

```

- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.article.AddUpdateArticleActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize" />

    </com.google.android.material.appbar.AppBarLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/tilArticleTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/appBar">

        <EditText
            android:id="@+id/edtArticleTitle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@android:color/transparent"
            android:hint="@string/article_title"
            android:importantForAutofill="no"
            android:inputType="text" />

    </com.google.android.material.textfield.TextInputLayout>

```

```

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/titArticleTitle">

            <EditText
                android:id="@+id/edtArticleContent"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@android:color/transparent"
                android:hint="@string/article_content"
                android:importantForAutofill="no"
                android:inputType="textCapSentences|textMultiLine"
            />

        </com.google.android.material.textfield.TextInputLayout>

        <RelativeLayout
            android:id="@+id/progressMessage"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:alpha=".8"
            android:background="@android:color/black"
            android:padding="16dp"
            android:visibility="gone">

            <ProgressBar
                android:id="@+id/progressBar"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:layout_centerVertical="true" />

            <TextView
                android:id="@+id/tvProgressMessage"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@id/progressBar"
                android:layout_centerHorizontal="true"
                android:layout_centerVertical="true"
                android:layout_marginTop="16dp"
                android:text="@string/save_please_wait"
                android:textAlignment="center"
                android:textColor="@android:color/white" />

        </RelativeLayout>

        <RelativeLayout
            android:id="@+id/errorMessage"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:alpha=".8"
            android:background="@android:color/black"
            android:padding="16dp"
            android:visibility="gone">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"

```

```

        android:layout_centerVertical="true"
        tools:ignore="UselessParent">

        <ImageView
            android:id="@+id/imgWarning"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_alignParentStart="true"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="0dp"
            android:layout_marginEnd="0dp"
            android:contentDescription="@string/warning"
            android:src="@drawable/ic_baseline_warning_24"

/>

        <TextView
            android:id="@+id/tvErrorMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/imgWarning"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:layout_marginTop="16dp"
            android:text="@string/example_error_message"
            android:textAlignment="center"
            android:textColor="@android:color/white" />

        <LinearLayout
            style="?android:attr/buttonBarStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/tvErrorMessage"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:layout_marginTop="16dp"
            android:orientation="horizontal">

            <Button
                android:id="@+id/btnCancelError"
                style="?android:attr/buttonBarButtonStyle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="16dp"
                android:text="@string/cancel" />

            <Button
                android:id="@+id/btnRetryError"
                style="?android:attr/buttonBarButtonStyle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="16dp"
                android:text="@string/retry" />

        </LinearLayout>

    </RelativeLayout>

</RelativeLayout>

</RelativeLayout>

```

```

- activity_add_update_kajian
- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:context="com.kajianid.ustadz.ui.kajian.AddUpdateKajianAct
    ivity">

    <com.google.android.material.appbar.AppBarLayout
      android:id="@+id/appBar"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize" />

    </com.google.android.material.appbar.AppBarLayout>

    <ScrollView
      android:id="@+id/svAddUpdateKajian"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_below="@id/appBar">

      <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_marginStart="16dp"
          android:layout_marginTop="16dp"
          android:layout_marginEnd="16dp"
          android:layout_marginBottom="8dp"
          android:text="@string/category" />

        <Spinner
          android:id="@+id/spKajianCategory"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_marginStart="16dp"
          android:layout_marginEnd="16dp"
          android:layout_marginBottom="16dp"

          android:entries="@array/kajian_category_entries" />

      <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/tilKajianYtLink"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"

```

```

        android:visibility="gone">

        <EditText
            android:id="@+id/edtKajianYtLink"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

            android:background="@android:color/transparent"
            android:hint="@string/youtube_link"
            android:importantForAutofill="no"
            android:inputType="text" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/tilKajianAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:visibility="visible">

        <EditText
            android:id="@+id/edtKajianAddress"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

            android:background="@android:color/transparent"
            android:hint="@string/address"
            android:importantForAutofill="no"

            android:inputType="textCapSentences|textMultiLine" />

    </com.google.android.material.textfield.TextInputLayout>

    <RelativeLayout
        android:id="@+id/rlKajianImage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/imgKajian"
            android:layout_width="match_parent"
            android:layout_height="250dp"
            android:background="@android:color/black"

            android:contentDescription="@string/kajian_poster" />

        <Button
            android:id="@+id/btnChooseKajianImage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBottom="@+id/imgKajian"
            android:layout_alignParentEnd="true"
            android:layout_margin="16dp"
            android:text="@string/select_image"
            tools:ignore="RelativeOverlap" />

```

```

        <TextView
            android:id="@+id/tvKajianImagePath"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:ellipsize="end"
            android:maxLines="1"
            android:text="@string/example_path"
            android:textColor="@android:color/white"
            android:visibility="gone" />

    </RelativeLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/tilKajianTitle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/edtKajianTitle"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

                android:background="@android:color/transparent"
                android:hint="@string/kajian_title"
                android:importantForAutofill="no"
                android:inputType="text" />

        </com.google.android.material.textfield.TextInputLayout>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:text="@string/mosque" />

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/tvMosqueId"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/mosque_null"
                android:visibility="gone" />

            <TextView
                android:id="@+id/tvMosqueName"
                android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:ellipsize="end"
        android:maxLines="1"
        android:text="@string/mosque_name_null"
        android:textSize="16sp" />

        <Button
            android:id="@+id/btnChooseMosque"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:text="@string/choose"
            tools:ignore="RelativeOverlap" />

    </RelativeLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/edtKajianDescription"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

            android:background="@android:color/transparent"
            android:hint="@string/description"
            android:importantForAutofill="no"

            android:inputType="textCapSentences|textMultiLine" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/edtKajianDateDue"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

            android:background="@android:color/transparent"
            android:focusable="false"
            android:hint="@string/date_due"
            android:importantForAutofill="no"
            android:inputType="none" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="70dp">

```



```

        <EditText
            android:id="@+id/edtKajianTimeDue"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

android:background="@android:color/transparent"
            android:focusable="false"
            android:hint="@string/time_due"
            android:importantForAutofill="no"
            android:inputType="none" />

</com.google.android.material.textfield.TextInputLayout>

    </LinearLayout>

</LinearLayout>

</ScrollView>

<RelativeLayout
    android:id="@+id/progressMessage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:alpha=".8"
    android:background="@android:color/black"
    android:padding="16dp"
    android:visibility="gone">

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true" />

    <TextView
        android:id="@+id/tvProgressMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/progressBar"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_marginTop="16dp"
        android:text="@string/save_please_wait"
        android:textAlignment="center"
        android:textColor="@android:color/white" />

</RelativeLayout>

<RelativeLayout
    android:id="@+id/errorMessage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:alpha=".8"
    android:background="@android:color/black"
    android:padding="16dp"
    android:visibility="gone">

    <RelativeLayout
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        tools:ignore="UselessParent">

        <ImageView
            android:id="@+id/imgWarning"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_alignParentStart="true"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="0dp"
            android:layout_marginEnd="0dp"
            android:contentDescription="@string/warning"
            android:src="@drawable/ic_baseline_warning_24"

/>

        <TextView
            android:id="@+id/tvErrorMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/imgWarning"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:layout_marginTop="16dp"
            android:text="@string/example_error_message"
            android:textAlignment="center"
            android:textColor="@android:color/white" />

        <LinearLayout
            style="?android:attr/buttonBarStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/tvErrorMessage"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:layout_marginTop="16dp"
            android:orientation="horizontal">

            <Button
                android:id="@+id/btnCancelError"
                style="?android:attr/buttonBarButtonStyle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="16dp"
                android:text="@string/cancel" />

            <Button
                android:id="@+id/btnRetryError"
                style="?android:attr/buttonBarButtonStyle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="16dp"
                android:text="@string/retry" />

        </LinearLayout>

    </RelativeLayout>

</RelativeLayout>

```

```
</RelativeLayout>
```

```
- activity_edit_identity
```

```
- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    tools:context="ui.credential.EditPasswordActivity">

    <com.google.android.material.textfield.TextInputLayout
      android:id="@+id/tilUstadzName"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginBottom="8dp">

      <EditText
        android:id="@+id/edtUstadzName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints=""
        android:hint="@string/name"
        android:inputType="textPersonName" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
      android:id="@+id/tilPhoneNumber"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/tilUstadzName"
      android:layout_marginBottom="8dp">

      <EditText
        android:id="@+id/edtPhoneNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints=""
        android:hint="@string/phone"
        android:inputType="phone" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
      android:id="@+id/tilUstadzAddress"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/tilPhoneNumber"
      android:layout_marginBottom="8dp">

      <EditText
        android:id="@+id/edtUstadzAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints=""
        android:hint="@string/address"
        android:inputType="textPostalAddress" />
```

```

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilEmailAddress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/tilUstadzAddress"
    android:layout_marginBottom="16dp">

    <EditText
        android:id="@+id/edtEmailAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints=""
        android:hint="@string/email"
        android:inputType="textEmailAddress" />

</com.google.android.material.textfield.TextInputLayout>

<RadioGroup
    android:id="@+id/rgGender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/tilEmailAddress"
    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/rdMale"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/male" />

    <RadioButton
        android:id="@+id/rdFemale"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/female" />

</RadioGroup>

<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/rgGender"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:visibility="gone" />

<RelativeLayout
    android:id="@+id/rlButtons"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/rgGender">

    <Button
        android:id="@+id/btnChange"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:text="@string/change" />

        <Button
            android:id="@+id/btnCancel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:text="@string/cancel" />

    </RelativeLayout>

</RelativeLayout>

```

- activity\_edit\_password

```

- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    tools:context="ui.credential.EditPasswordActivity">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/tilOldPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp">

        <EditText
            android:id="@+id/edtOldPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:autofillHints=""
            android:hint="@string/old_password"
            android:inputType="textPassword" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/tilNewPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/tilOldPassword"
        android:layout_marginBottom="8dp">

        <EditText
            android:id="@+id/edtNewPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:autofillHints=""
            android:hint="@string/new_password"
            android:inputType="textPassword" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/tilConfirmPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/tilNewPassword"

```

```

        android:layout_marginBottom="16dp">

        <EditText
            android:id="@+id/edtConfirmPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:autofillHints=""
            android:hint="@string/confirm_password"
            android:inputType="textPassword" />

    </com.google.android.material.textfield.TextInputLayout>

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tilConfirmPassword"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:visibility="gone" />

    <RelativeLayout
        android:id="@+id/rlButtons"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tilConfirmPassword">

        <Button
            android:id="@+id/btnChange"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/change" />

        <Button
            android:id="@+id/btnCancel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:text="@string/cancel" />

    </RelativeLayout>

</RelativeLayout>

```

```

- activity_login
- <?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp">

    <ScrollView
        android:id="@+id/scrollView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/progressBar"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

```

```

        <include
            android:id="@+id/contentLogin"
            layout="@layout/content_login" />

    </ScrollView>

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <LinearLayout
        android:id="@+id/llLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">

        <Button
            android:id="@+id/btn_register"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:background="@drawable/bt_aksi"
            android:backgroundTint="@color/abuabu"
            android:backgroundTintMode="src_atop"
            android:fontFamily="sans-serif-light"
            android:text="@string/register"
            android:textAllCaps="false"
            android:textStyle="bold" />

        <Button
            android:id="@+id/btnLogin"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:background="@drawable/bt_aksi"
            android:fontFamily="sans-serif-light"
            android:text="@string/login"
            android:textAllCaps="false"
            android:textStyle="bold" />

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

- activity_main
- <?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="ui.main.MainActivity">

        <com.google.android.material.appbar.AppBarLayout
            android:id="@+id/appBar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:elevation="0dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <com.google.android.material.tabs.TabLayout
                android:id="@+id/tabs"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@id/appBar"
                android:background="@android:color/transparent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/appBar"
                app:tabIndicatorAnimationDuration="50"
                app:tabIndicatorColor="#E91E63"
                app:tabSelectedTextColor="#E91E63" />

        </com.google.android.material.appbar.AppBarLayout>

        <androidx.viewpager.widget.ViewPager
            android:id="@+id/view_pager"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_below="@id/appBar"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/appBar" />

    </androidx.constraintlayout.widget.ConstraintLayout>

```

```

- activity_mosque_chooser
- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="450dp"
    android:orientation="vertical"
    tools:context="ui.mosque.MosqueChooserActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize" />

    <TextView

```



```

        android:id="@+id/tvMosqueNote"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/toolbar"
        android:paddingStart="16dp"
        android:paddingEnd="16dp"
        android:paddingBottom="8dp"
        android:text="@string/mosque_note"
        android:textStyle="bold" />

<TextView
    android:id="@+id/tv_no_mosque"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:paddingBottom="8dp"
    android:layout_below="@id/tvMosqueNote"
    android:textColor="?android:textColorLink"
    android:textStyle="bold"
    android:text="@string/mosque_unregistered"
    android:autoLink="all" />

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvMosqueList"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_above="@+id/btnBack"
    android:layout_below="@id/tv_no_mosque"
    tools:listitem="@layout/item_mosque" />

<RelativeLayout
    android:id="@+id/progressMessage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/tvMosqueNote"
    android:background="@android:color/white"
    android:padding="16dp"
    android:visibility="gone">

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/progressBar"
        android:layout_centerHorizontal="true"
        android:paddingTop="16dp"
        android:text="@string/load_data"
        android:textAlignment="center" />

</RelativeLayout>

<Button
    android:id="@+id/btnBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_margin="8dp"
        android:text="@string/back" />

```

```

</RelativeLayout>

```

- activity\_read\_article

```

- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.article.ReadArticleActivity">

    <com.google.android.material.appbar.AppBarLayout
      android:id="@+id/appBar"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize" />

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
      android:id="@+id/pullToRefresh"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_below="@id/appBar">

      <ScrollView
        android:id="@+id/svArticle"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <include
          android:id="@+id/contentReadArticle"
          layout="@layout/content_read_article" />

      </ScrollView>

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

    <RelativeLayout
      android:id="@+id/progressMessage"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_below="@id/appBar"
      android:layout_centerInParent="true"
      android:padding="30dp">

      <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true" />

```

```

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/progressBar"
            android:layout_centerHorizontal="true"
            android:paddingTop="16dp"
            android:text="@string/load_data"
            android:textAlignment="center" />

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/errorMessage"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar"
        android:layout_centerInParent="true"
        android:padding="30dp"
        android:visibility="visible">

        <ImageView
            android:id="@+id/imgWarning"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_centerHorizontal="true"
            android:contentDescription="@string/warning"
            android:src="@drawable/ic_baseline_warning_24" />

        <TextView
            android:id="@+id/tvErrorMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/imgWarning"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:text="@string/example_error_message"
            android:textAlignment="center" />

        <Button
            android:id="@+id/btnRefresh"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/tvErrorMessage"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:text="@string/refresh" />

    </RelativeLayout>

</RelativeLayout>

```

- activity\_settings

```

- <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.settings.SettingsActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </com.google.android.material.appbar.AppBarLayout>

    <FrameLayout
        android:id="@+id/settings"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar" />
</RelativeLayout>

```

- activity\_show\_kajian

```

- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.kajian.ShowKajianActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize" />

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/pullToRefresh"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar">

        <ScrollView
            android:id="@+id/svKajian"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <include
                android:id="@+id/contentShowKajian"
                layout="@layout/content_show_kajian" />

        </ScrollView>

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

    <RelativeLayout
        android:id="@+id/progressMessage"
        android:layout_width="match_parent"
        android:layout_height="match_parent"

```

```

        android:layout_below="@id/appBar"
        android:layout_centerInParent="true"
        android:padding="30dp">

        <ProgressBar
            android:id="@+id/progressBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/progressBar"
            android:layout_centerHorizontal="true"
            android:paddingTop="16dp"
            android:text="@string/load_data"
            android:textAlignment="center" />

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/errorMessage"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBar"
        android:layout_centerInParent="true"
        android:padding="30dp"
        android:visibility="visible">

        <ImageView
            android:id="@+id/imgWarning"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_centerHorizontal="true"
            android:contentDescription="@string/warning"
            android:src="@drawable/ic_baseline_warning_24" />

        <TextView
            android:id="@+id/tvErrorMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/imgWarning"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:text="@string/example_error_message"
            android:textAlignment="center" />

        <Button
            android:id="@+id/btnRefresh"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/tvErrorMessage"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:text="@string/refresh" />

    </RelativeLayout>

</RelativeLayout>

```

- activity\_splash

```
<?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="ui.intro.SplashActivity">

    <ImageView
        android:id="@+id/nama_slogan"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:contentDescription="@string/logo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/version"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nama_slogan"
    />

    <TextView
        android:id="@+id/version"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="40dp"
        android:text="@string/versinya"
        android:textColor="?attr/editTextColor"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/copyright"
        android:textColor="?attr/editTextColor"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/version"
        app:layout_constraintVertical_bias="0.3" />

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

- content\_login

```
<?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"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:padding="8dp">

<ImageView
    android:id="@+id/nama_slogan3"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginTop="100dp"
    android:layout_marginBottom="50dp"
    android:contentDescription="@string/logo"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:gravity="center"
    android:orientation="vertical"
    android:paddingBottom="60dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <TextView
        android:id="@+id/textlogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="18dp"
        android:layout_marginBottom="20dp"
        android:fontFamily="sans-serif-light"
        android:gravity="center_horizontal"
        android:text="@string/login_as_ustadz"
        android:textAlignment="center"
        android:textSize="25sp"
        android:textStyle="bold" />

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_marginBottom="10dp"
        android:background="@drawable/bg_edittext"
        android:paddingLeft="25dp"
        android:paddingRight="25dp">

        <EditText
            android:id="@+id/edtUsername"
            android:layout_width="300dp"
            android:layout_height="50dp"
            android:layout_centerHorizontal="true"
            android:background="@android:color/transparent"
            android:hint="@string/username"
            android:imeOptions="actionNext"
            android:importantForAutofill="no"

```

```

        android:inputType="text"
        android:maxLines="1"
        android:singleLine="true"
        android:textSize="14sp" />

</RelativeLayout>

<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginBottom="10dp"
    android:background="@drawable/bg_editttext"
    android:paddingLeft="25dp"
    android:paddingRight="25dp">

    <EditText
        android:id="@+id/edtPassword"
        android:layout_width="300dp"
        android:layout_height="50dp"
        android:layout_centerHorizontal="true"
        android:background="@android:color/transparent"
        android:hint="@string/password"
        android:imeOptions="actionDone"
        android:importantForAutofill="no"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true"
        android:textSize="14sp" />

</RelativeLayout>

<CheckBox
    android:id="@+id/showPass"
    android:layout_width="300dp"
    android:layout_height="20dp"
    android:text="@string/show_password"
    android:textSize="12sp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

- content_read_article
- <?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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <TextView
        android:id="@+id/tvArticleTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="16dp"
        android:text="@string/example_article_title"
        android:textSize="18sp"

```



```

        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<ImageView
    android:id="@+id/imgArticle"
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:contentDescription="@string/post_picture"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvArticleTitle" />

<ImageView
    android:id="@+id/imgLike"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:contentDescription="@string/likes"
    android:src="@drawable/ic_baseline_thumb_up_24"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imgArticle"
    android:visibility="gone" />

<TextView
    android:id="@+id/tvLikeCount"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="@string/example_likes"
    app:layout_constraintBottom_toBottomOf="@+id/imgLike"
    app:layout_constraintStart_toEndOf="@+id/imgLike"
    app:layout_constraintTop_toTopOf="@+id/imgLike"
    android:visibility="gone" />

<ImageView
    android:id="@+id/imgCalendar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="0dp"
    android:contentDescription="@string/post_date"
    android:src="@drawable/ic_baseline_calendar_today_24"
    app:layout_constraintBottom_toBottomOf="@+id/tvLikeCount"
    app:layout_constraintStart_toEndOf="@+id/tvLikeCount"
    app:layout_constraintTop_toTopOf="@+id/tvLikeCount" />

<TextView
    android:id="@+id/tvPostDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="@string/example_post_date"
    app:layout_constraintBottom_toBottomOf="@+id/imgCalendar"
    app:layout_constraintStart_toEndOf="@+id/imgCalendar"
    app:layout_constraintTop_toTopOf="@+id/imgCalendar" />

<TextView
    android:id="@+id/tvPostContent"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autoLink="all"
        android:paddingTop="16dp"
        android:paddingBottom="24dp"
        android:text="@string/example_text"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imgCalendar" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

- content\_show\_kajian

```

- <?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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <ImageView
      android:id="@+id/imgThumbnail"
      android:layout_width="match_parent"
      android:layout_height="200dp"
      android:contentDescription="@string/thumbnail"
      android:visibility="visible"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
      android:id="@+id/btnPlay"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:background="@android:color/transparent"
      android:contentDescription="@string/play_video"
      android:src="@drawable/ic_baseline_play_circle_outline_100"
      app:layout_constraintBottom_toBottomOf="@+id/imgThumbnail"
      app:layout_constraintEnd_toEndOf="@+id/imgThumbnail"
      app:layout_constraintStart_toStartOf="@+id/imgThumbnail"
      app:layout_constraintTop_toTopOf="@+id/imgThumbnail" />

    <TextView
      android:id="@+id/tvKajianTitle"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:paddingTop="16dp"
      android:text="@string/example_kajian_title"
      android:textSize="18sp"
      android:textStyle="bold"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/imgThumbnail" />

    <TextView
      android:id="@+id/tvMosqueAddress"
      android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:text="@string/example_mosque"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvKajianTitle" />

<TextView
    android:id="@+id/tvUstadzName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/by"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvMosqueAddress" />

<TextView
    android:id="@+id/tvCategory"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/on_stage"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvUstadzName" />

<TextView
    android:id="@+id/tvTimestampAnnounce"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/timestamp_announce"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvCategory" />

<TextView
    android:id="@+id/tvTimestampDue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/timestamp_due"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/tvTimestampAnnounce" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/description"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvTimestampDue" />

<TextView
    android:id="@+id/tvDescription"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/example_text"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView7" />

```

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

```
- fragment_article
- <?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=".ui.main.article.ArticleFragment">

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
      android:id="@+id/pullToRefresh"
      android:layout_width="match_parent"
      android:layout_height="0dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/relativeLayout">

      <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_articles"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="16dp"

        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        tools:listitem="@layout/item_kajian" />

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

    <RelativeLayout
      android:id="@+id/relativeLayout"
      android:layout_width="match_parent"
      android:layout_height="50dp"
      android:layout_margin="16dp"
      android:background="@drawable/bg_edittext"
      android:paddingLeft="25dp"
      android:paddingRight="25dp"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent">

      <EditText
        android:id="@+id/et_search_article"
        android:layout_width="300dp"
        android:layout_height="50dp"
        android:layout_centerHorizontal="true"
        android:autoFillHints=""
        android:background="@android:color/transparent"
        android:hint="@string/search_post"
        android:imeOptions="actionSearch"
        android:inputType="text"
        android:maxLines="1"
        android:singleLine="true"
        android:textSize="14sp" />
```

```

</RelativeLayout>

<com.google.android.material.floatingactionbutton.ExtendedFloatingActionButton
    android:id="@+id/extended_fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:text="@string/add_post"
    app:icon="@drawable/ic_baseline_post_add_24"
    app:layout_anchor="@id/pullToRefresh"
    app:layout_anchorGravity="bottom|end"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<RelativeLayout
    android:id="@+id/progressMessage"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:padding="30dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/relativeLayout">

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/progressBar"
        android:layout_centerHorizontal="true"
        android:paddingTop="16dp"
        android:text="@string/load_data"
        android:textAlignment="center" />

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

- fragment_developers
- <?xml version="1.0" encoding="utf-8"?>
  <ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.about.DevelopersFragment">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:layout_margin="8dp"
        android:orientation="vertical">

        <de.hdodenhof.circleimageview.CircleImageView
            android:id="@+id/civPhoto"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:layout_gravity="center"
            android:layout_margin="16dp"
            android:elevation="1dp" />

        <TextView
            android:id="@+id/tvName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/example_about_name"
            android:textColor="?android:textColorPrimary"
            android:textSize="20sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/tvNim"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/example_about_nim" />

        <TableLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp">

            <TableRow

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="8dp">

                        <TextView
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:layout_marginEnd="16dp"
                            android:text="@string/label_about_grade" />

                        <TextView
                            android:id="@+id/tvGrade"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:layout_weight="1"
                            android:text="@string/example_about_grade"
                            android:textAlignment="textEnd"
                            android:textColor="?android:textColorPrimary" />

                    </TableRow>

                <TableRow
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="8dp">

```

```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"

android:text="@string/label_about_year_registered" />

        <TextView
            android:id="@+id/tvYearRegistered"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"

android:text="@string/example_about_year_registered"
            android:textAlignment="textEnd"
            android:textColor="?android:textColorPrimary" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="8dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"
            android:text="@string/label_about_github" />

        <TextView
            android:id="@+id/tvGitHub"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/example_about_github"
            android:textAlignment="textEnd"
            android:textColor="?android:textColorPrimary" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="8dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"
            android:text="@string/label_about_email" />

        <TextView
            android:id="@+id/tvEmail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:autoLink="email"

```

```

        android:text="@string/example_about_email"
        android:textAlignment="textEnd"
        android:textColor="?android:textColorPrimary"
        android:textColorLink="?android:textColorPrimary"
    />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="8dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="16dp"
        android:text="@string/label_about_address" />

    <TextView
        android:id="@+id/tvAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/example_about_address"
        android:textAlignment="textEnd"
        android:textColor="?android:textColorPrimary" />

</TableRow>

</TableLayout>

<com.google.android.material.button.MaterialButton
    android:id="@+id/btnVisit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="36dp"
    android:text="@string/visit_my_github" />

</LinearLayout>

</ScrollView>

```

```

- fragment_kajian
- <?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=".ui.main.kajian.KajianFragment">

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/pullToRefresh"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"

```



```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/relativeLayout">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rv_info"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="16dp"

app:layout_behavior="@string/appbar_scrolling_view_behavior"
        tools:listitem="@layout/item_kajian" />

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

    <RelativeLayout
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_margin="16dp"
        android:background="@drawable/bg_edittext"
        android:paddingLeft="16dp"
        android:paddingRight="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <EditText
            android:id="@+id/et_search_kajian"
            android:layout_width="320dp"
            android:layout_height="50dp"
            android:layout_centerHorizontal="true"
            android:autoFillHints=""
            android:background="@android:color/transparent"
            android:hint="@string/search_kajian"
            android:imeOptions="actionSearch"
            android:inputType="text"
            android:maxLines="1"
            android:singleLine="true"
            android:textSize="14sp" />

    </RelativeLayout>

    <com.google.android.material.floatingactionbutton.ExtendedFloatingAct
ionButton
        android:id="@+id/fab_add_kajian"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:text="@string/add_kajian"
        app:icon="@drawable/ic_baseline_add_24"
        app:layout_anchor="@id/pullToRefresh"
        app:layout_anchorGravity="bottom|end"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <RelativeLayout
        android:id="@+id/progressMessage"

```

```

        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:background="@android:color/white"
        android:padding="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/pullToRefresh">

        <ProgressBar
            android:id="@+id/progressBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/progressBar"
            android:layout_centerHorizontal="true"
            android:paddingTop="16dp"
            android:text="@string/load_data"
            android:textAlignment="center" />

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

- fragment_mosque_list
- <?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="ui.mosque.MosqueListFragment">

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/pullToRefresh"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/relativeLayout">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rvMosque"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="16dp"

            app:layout_behavior="@string/appbar_scrolling_view_behavior"
            tools:listitem="@layout/item_mosque_list" />

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

```

```

<RelativeLayout
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_margin="16dp"
    android:background="@drawable/bg_edittext"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <EditText
        android:id="@+id/et_search_mosque"
        android:layout_width="320dp"
        android:layout_height="50dp"
        android:layout_centerHorizontal="true"
        android:autoFillHints=""
        android:background="@android:color/transparent"
        android:hint="@string/search_mosque"
        android:imeOptions="actionSearch"
        android:inputType="text"
        android:maxLines="1"
        android:singleLine="true"
        android:textSize="14sp" />

</RelativeLayout>

<com.google.android.material.floatingactionbutton.ExtendedFloatingActionButton
    android:id="@+id/fab_add_mosque"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:text="@string/add_mosque"
    app:icon="@drawable/ic_baseline_add_24"
    app:layout_anchor="@id/pullToRefresh"
    app:layout_anchorGravity="bottom|end"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<RelativeLayout
    android:id="@+id/progressMessage"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:background="@android:color/white"
    android:padding="30dp"
    android:visibility="gone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/pullToRefresh">

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true" />

```

```

        <TextView
            android:id="@+id/tvProgressMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/progressBar"
            android:layout_centerHorizontal="true"
            android:paddingTop="16dp"
            android:text="@string/load_data"
            android:textAlignment="center" />

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

- item_articles
- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/layout_data_kajian"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:focusable="true"
    android:paddingLeft="8dp"
    android:paddingRight="8dp">

    <androidx.cardview.widget.CardView
        android:id="@+id/cv_item_article"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        app:cardElevation="4dp"
        app:cardMaxElevation="4dp"
        app:cardUseCompatPadding="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp">

            <ImageView
                android:id="@+id/imgArticle"
                android:layout_width="match_parent"
                android:layout_height="150dp"
                android:contentDescription="@string/icon_kajian" />

            <TextView
                android:id="@+id/tv_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/imgArticle"
                android:ellipsize="end"
                android:maxLines="2"
                android:text="@string/example_article_title"
                android:textColor="?android:textColorPrimary"

```

```

        android:textSize="18sp"
        android:textStyle="bold" />

        <TextView
            android:id="@+id/tv_text_less"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tv_title"
            android:ellipsize="end"
            android:maxLines="2"
            android:text="@string/example_text" />

        <TextView
            android:id="@+id/tv_post_date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/tv_text_less"
            android:layout_alignParentEnd="true"
            android:layout_marginTop="8dp"
            android:text="@string/example_tanggal" />

    </RelativeLayout>

</androidx.cardview.widget.CardView>
</RelativeLayout>

```

```

- item_kajian
- <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/layout_data_kajian"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:focusable="true"
    android:paddingLeft="8dp"
    android:paddingRight="8dp">

    <androidx.cardview.widget.CardView
        android:id="@+id/cv_item_kajian"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        app:cardElevation="4dp"
        app:cardMaxElevation="4dp"
        app:cardUseCompatPadding="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp">

```

```

<ImageView
    android:id="@+id/img_kajian"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:contentDescription="@string/icon_kajian"
    android:src="@drawable/icon" />

<TextView
    android:id="@+id/tv_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/img_kajian"
    android:ellipsize="end"
    android:maxLines="1"
    android:text="@string/example_kajian_title"
    android:textColor="?android:textColorPrimary"
    android:textSize="18sp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/tv_mosque_address"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tv_title"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/img_kajian"
    android:ellipsize="end"
    android:maxLines="1"
    android:text="@string/example_mosque" />

<TextView
    android:id="@+id/tv_ket"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tv_mosque_address"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/img_kajian"
    android:text="@string/example_keterangan" />

<TextView
    android:id="@+id/tv_due"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tv_mosque_address"
    android:layout_alignParentEnd="true"
    android:text="@string/example_tanggal" />

</RelativeLayout>

</androidx.cardview.widget.CardView>
</RelativeLayout>

```

```

- item_mosque
- <?xml version="1.0" encoding="utf-8"?>
  <androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:clickable="true"
        android:elevation="1dp"
        android:focusable="true">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp">

            <TextView
                android:id="@+id/tvMosqueName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"
                android:layout_marginEnd="8dp"
                android:text="@string/example_mosque"
                android:textColor="?android:textColorPrimary"
                android:textSize="16sp"
                android:textStyle="bold"
                app:layout_constraintEnd_toStartOf="@+id/imgNext"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <TextView
                android:id="@+id/tvMosqueAddress"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"
                android:layout_marginEnd="8dp"
                android:ellipsize="end"
                android:maxLines="4"
                android:text="@string/example_address"
                app:layout_constraintEnd_toStartOf="@+id/imgNext"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/tvMosqueName"
            />

            <ImageView
                android:id="@+id/imgNext"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:contentDescription="@string/next"
                android:src="@drawable/ic_baseline_navigate_next_24"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

        </androidx.constraintlayout.widget.ConstraintLayout>

    </androidx.cardview.widget.CardView>

```

```

- item_mosque_list
- <?xml version="1.0" encoding="utf-8"?>
  <androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"

```

```

xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:clickable="true"
android:elevation="1dp"
android:focusable="true">

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <TextView
        android:id="@+id/tvMosqueName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:text="@string/example_mosque"
        android:textColor="?android:textColorPrimary"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvMosqueAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:ellipsize="end"
        android:maxLines="4"
        android:text="@string/example_address"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvMosqueName"
    />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

```

#### MENU

- menu\_about

```

- <?xml version="1.0" encoding="utf-8"?>
  <menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/menuGithub"
        android:title="@string/github"
        app:showAsAction="always" />
  </menu>

```

- menu\_add\_update



```
- <?xml version="1.0" encoding="utf-8"?>
  <menu xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/menuSave"
        android:icon="@drawable/ic_baseline_check_24"
        android:title="@string/save"
        android:visible="true"
        app:showAsAction="ifRoom" />

  </menu>
```

- menu\_article\_kajian

```
- <?xml version="1.0" encoding="utf-8"?>
  <menu xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/menuEdit"
        android:icon="@drawable/ic_baseline_edit_24"
        android:title="@string/edit"
        app:showAsAction="ifRoom" />

    <item
        android:id="@+id/menuDelete"
        android:icon="@drawable/ic_baseline_delete_24"
        android:title="@string/delete"
        app:showAsAction="ifRoom" />

    <item
        android:id="@+id/menuShare"
        android:icon="@drawable/ic_baseline_share_24"
        android:title="@string/menu_share"
        app:showAsAction="ifRoom" />

  </menu>
```

- menu\_main

```
- <?xml version="1.0" encoding="utf-8"?>
  <menu xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/menuSettings"
        android:title="@string/settings"
        app:showAsAction="never" />

    <item
        android:id="@+id/menuAbout"
        android:title="@string/about"
        app:showAsAction="never" />

  </menu>
```

- menu\_mosque\_chooser

```
- <?xml version="1.0" encoding="utf-8"?>
  <menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```

xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/search"
        android:icon="@drawable/ic_baseline_search_24"
        android:title="@string/search"
        app:actionViewClass="androidx.appcompat.widget.SearchView"
        app:showAsAction="collapseActionView|ifRoom" />

</menu>

```

JAVA

```

- Article.java
- package com.kajianid.ustadz.data;

import android.os.Parcel;
import android.os.Parcelable;

public class Article implements Parcelable {
    public static final Creator<Article> CREATOR = new
    Creator<Article>() {
        @Override
        public Article createFromParcel(Parcel in) {
            return new Article(in);
        }

        @Override
        public Article[] newArray(int size) {
            return new Article[size];
        }
    };
    private String id;
    private String title;
    private String content;
    private String post_date;
    private boolean hasImg;
    private String imgUrl;
    private int likes;

    public Article() {

    }

    protected Article(Parcel in) {
        id = in.readString();
        title = in.readString();
        content = in.readString();
        post_date = in.readString();
        hasImg = in.readByte() != 0;
        imgUrl = in.readString();
        likes = in.readInt();
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

```

```

    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }

    public String getPost_date() {
        return post_date;
    }

    public void setPost_date(String post_date) {
        this.post_date = post_date;
    }

    public boolean isHasImg() {
        return hasImg;
    }

    public void setHasImg(boolean hasImg) {
        this.hasImg = hasImg;
    }

    public String getImgUrl() {
        return imgUrl;
    }

    public void setImgUrl(String imgUrl) {
        this.imgUrl = imgUrl;
    }

    public int getLikes() {
        return likes;
    }

    public void setLikes(int likes) {
        this.likes = likes;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(id);
        dest.writeString(title);
    }

```

```

        dest.writeString(content);
        dest.writeString(post_date);
        dest.writeByte((byte) (hasImg ? 1 : 0));
        dest.writeString(imgUrl);
        dest.writeInt(likes);
    }
}

```

- Credential.java

```

- package com.kajianid.ustadz.data;

import android.os.Parcel;
import android.os.Parcelable;

public class Credential implements Parcelable {

    public static final Creator<Credential> CREATOR = new
    Creator<Credential>() {
        @Override
        public Credential createFromParcel(Parcel in) {
            return new Credential(in);
        }

        @Override
        public Credential[] newArray(int size) {
            return new Credential[size];
        }
    };
    String username = null;
    String password = null;

    public Credential() {
        // empty public constructor
    }

    protected Credential(Parcel in) {
        username = in.readString();
        password = in.readString();
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public int describeContents() {
        return 0;
    }
}

```

```

    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(username);
        dest.writeString(password);
    }
}

```

- Kajian.java

```

- package com.kajianid.ustadz.data;

import android.os.Parcel;
import android.os.Parcelable;

public class Kajian implements Parcelable {

    public static final Creator<Kajian> CREATOR = new
    Creator<Kajian>() {
        @Override
        public Kajian createFromParcel(Parcel in) {
            return new Kajian(in);
        }

        @Override
        public Kajian[] newArray(int size) {
            return new Kajian[size];
        }
    };

    private String id;
    private String title;
    private String mosqueId;
    private String mosqueName;
    private String place;
    private String address;
    private String date;
    private String description;
    private String imgResource;
    private String ytLink;

    public Kajian() {

    }

    protected Kajian(Parcel in) {
        id = in.readString();
        title = in.readString();
        mosqueId = in.readString();
        mosqueName = in.readString();
        place = in.readString();
        address = in.readString();
        date = in.readString();
        description = in.readString();
        imgResource = in.readString();
        ytLink = in.readString();
    }

    public String getId() {
        return id;
    }
}

```

```

    }

    public void setId(String id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getMosqueId() {
        return mosqueId;
    }

    public void setMosqueId(String mosqueId) {
        this.mosqueId = mosqueId;
    }

    public String getMosqueName() {
        return mosqueName;
    }

    public void setMosqueName(String mosqueName) {
        this.mosqueName = mosqueName;
    }

    public String getPlace() {
        return place;
    }

    public void setPlace(String place) {
        this.place = place;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {

```

```

        this.description = description;
    }

    public String getImgResource() {
        return imgResource;
    }

    public void setImgResource(String imgResource) {
        this.imgResource = imgResource;
    }

    public String getYtLink() {
        return ytLink;
    }

    public void setYtLink(String ytLink) {
        this.ytLink = ytLink;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(id);
        dest.writeString(title);
        dest.writeString(mosqueId);
        dest.writeString(mosqueName);
        dest.writeString(place);
        dest.writeString(address);
        dest.writeString(date);
        dest.writeString(description);
        dest.writeString(imgResource);
        dest.writeString(ytLink);
    }
}

```

- Mosque.java

```

- package com.kajianid.ustadz.data;

import android.os.Parcel;
import android.os.Parcelable;

public class Mosque implements Parcelable {
    public static final Creator<Mosque> CREATOR = new
    Creator<Mosque>() {
        @Override
        public Mosque createFromParcel(Parcel in) {
            return new Mosque(in);
        }

        @Override
        public Mosque[] newArray(int size) {
            return new Mosque[size];
        }
    };
    private int id;

```

```

private String mosqueName;
private String latLng;
private String address;

protected Mosque(Parcel in) {
    id = in.readInt();
    mosqueName = in.readString();
    latLng = in.readString();
    address = in.readString();
}

public Mosque() {

}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getMosqueName() {
    return mosqueName;
}

public void setMosqueName(String mosqueName) {
    this.mosqueName = mosqueName;
}

public String getLatLng() {
    return latLng;
}

public void setLatLng(String latLng) {
    this.latLng = latLng;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

@Override
public int describeContents() {
    return 0;
}

@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeInt(id);
    dest.writeString(mosqueName);
    dest.writeString(latLng);
    dest.writeString(address);
}

```



```
}  
}
```

- Ustadz.java

```
- package com.kajianid.ustadz.data;  
  
import android.os.Parcel;  
import android.os.Parcelable;  
  
public class Ustadz implements Parcelable {  
    public static final Creator<Ustadz> CREATOR = new  
    Creator<Ustadz>() {  
        @Override  
        public Ustadz createFromParcel(Parcel in) {  
            return new Ustadz(in);  
        }  
  
        @Override  
        public Ustadz[] newArray(int size) {  
            return new Ustadz[size];  
        }  
    };  
    private String name;  
    private String phone;  
    private String address;  
    private String email;  
    private String gender;  
  
    public Ustadz() {  
  
    }  
  
    protected Ustadz(Parcel in) {  
        name = in.readString();  
        phone = in.readString();  
        address = in.readString();  
        email = in.readString();  
        gender = in.readString();  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getPhone() {  
        return phone;  
    }  
  
    public void setPhone(String phone) {  
        this.phone = phone;  
    }  
  
    public String getAddress() {  
        return address;  
    }  
}
```

```

        public void setAddress(String address) {
            this.address = address;
        }

        public String getEmail() {
            return email;
        }

        public void setEmail(String email) {
            this.email = email;
        }

        public String getGender() {
            return gender;
        }

        public void setGender(String gender) {
            this.gender = gender;
        }

        @Override
        public int describeContents() {
            return 0;
        }

        @Override
        public void writeToParcel(Parcel dest, int flags) {
            dest.writeString(name);
            dest.writeString(phone);
            dest.writeString(address);
            dest.writeString(email);
            dest.writeString(gender);
        }
    }
}

```

```

- CredentialPreference.java
- package com.kajianid.ustadz.prefs;

import android.content.Context;
import android.content.SharedPreferences;

import com.kajianid.ustadz.data.Credential;

public class CredentialPreference {

    private static final String PREFERENCE_NAME =
"credential_preference";
    private static final String USERNAME = "username";
    private static final String PASSWORD = "password";

    private final SharedPreferences sharedPreferences;

    public CredentialPreference(Context context) {
        sharedPreferences =
context.getSharedPreferences(PREFERENCE_NAME, Context.MODE_PRIVATE);
    }

    public Credential getCredential() {

```

```

        Credential credential = new Credential();
        credential.setUsername(sharedPreferences.getString(USERNAME,
        ""));
        credential.setPassword(sharedPreferences.getString(PASSWORD,
        ""));
        return credential;
    }

    public void setCredential(Credential credential) {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString(USERNAME, credential.getUsername());
        editor.putString(PASSWORD, credential.getPassword());
        editor.apply();
    }
}

```

- AboutActivity.java

```

- package com.kajianid.ustadz.ui.about;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.databinding.ActivityAboutBinding;

import java.util.Objects;

public class AboutActivity extends AppCompatActivity {

    private ActivityAboutBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityAboutBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        AboutPagerAdapter adapter = new AboutPagerAdapter(this,
        getSupportFragmentManager());
        binding.vpAbout.setAdapter(adapter);
        binding.tabs.setupWithViewPager(binding.vpAbout);

        setSupportActionBar(binding.toolbar);

        Objects.requireNonNull(getSupportActionBar()).setTitle(getString(R.string.about));
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

```

```

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

@SuppressLint("NonConstantResourceId")
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            super.onBackPressed();
            break;
        case R.id.menuGithub:
            Intent i = new Intent(Intent.ACTION_VIEW);
            i.setData(Uri.parse("https://github.com/IF18A-Kajian-ID"));
            startActivity(i);
            break;
    }
    return super.onOptionsItemSelected(item);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    binding = null;
}
}

```

- AboutPagerAdapter.java

```

package com.kajianid.ustadz.ui.about;

import android.content.Context;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.StringRes;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;

import com.kajianid.ustadz.R;

public class AboutPagerAdapter extends FragmentPagerAdapter {

    private final Context context;

    @StringRes
    private final int[] tabTitles = {
        R.string.Leader,
        R.string.member_one,
        R.string.member_two,
        R.string.member_three,
        R.string.member_four
    };

    public AboutPagerAdapter(Context context, FragmentManager fm) {

```

```

        super(fm, BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
        this.context = context;
    }

    @NonNull
    @Override
    public Fragment getItem(int position) {
        return DevelopersFragment.newInstance(position);
    }

    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        return context.getResources().getString(tabTitles[position]);
    }

    @Override
    public int getCount() {
        return tabTitles.length;
    }
}

```

- DevelopersFragment.java

```

- package com.kajianid.ustadz.ui.about;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import com.bumptech.glide.Glide;
import com.kajianid.ustadz.databinding.FragmentDevelopersBinding;
import com.kajianid.ustadz.utils.DeveloperData;

public class DevelopersFragment extends Fragment {

    private static final String ARG_DEVELOPER_POSITION =
"arg_developer_position";
    private FragmentDevelopersBinding binding;

    public DevelopersFragment() {
        // Required empty public constructor
    }

    public static DevelopersFragment newInstance(int position) {
        DevelopersFragment fragment = new DevelopersFragment();
        Bundle args = new Bundle();

        args.putInt(ARG_DEVELOPER_POSITION, position);

        fragment.setArguments(args);
        return fragment;
    }
}

```

```

    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        binding = FragmentDevelopersBinding.inflate(inflater);
        return binding.getRoot();
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        int position = 1;
        if (getArguments() != null) {
            position =
                requireArguments().getInt(ARG_DEVELOPER_POSITION, 1);
        }

        Glide.with(this)
            .load(DeveloperData.drawableId[position])
            .override(150, 150)
            .centerCrop()
            .into(binding.civPhoto);

        binding.tvName.setText(DeveloperData.name[position]);
        binding.tvNim.setText(DeveloperData.nim[position]);
        binding.tvGrade.setText(DeveloperData.kelas[position]);

        binding.tvYearRegistered.setText(DeveloperData.yearRegister[position]);
    }

    binding.tvGitHub.setText(DeveloperData.githubUsername[position]);
    binding.tvEmail.setText(DeveloperData.email[position]);
    binding.tvAddress.setText(DeveloperData.address[position]);

    String githubUrl = "https://github.com/" +
        DeveloperData.githubUsername[position];
    binding.btnVisit.setOnClickListener(itView -> {
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(githubUrl));
        startActivity(i);
    });
}
}

```

```

- AddUpdateArticleActivity.java
- package com.kajianid.ustadz.ui.article;

import android.os.Bundle;
import android.view.Menu;

```

```

import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.kajianid.ustadz.R;
import
com.kajianid.ustadz.databinding.ActivityAddUpdateArticleBinding;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.kajianid.ustadz.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;

import java.util.Objects;

import cz.msebera.android.httpclient.Header;

public class AddUpdateArticleActivity extends AppCompatActivity {

    public static final String EXTRA_ARTICLE_ID = "extra_article_id";
    public static final String EXTRA_ARTICLE_TITLE =
"extra_article_title";
    public static final String EXTRA_ARTICLE_CONTENT =
"extra_article_content";
    public static final int RESULT_SAVE = 130;
    public static final int RESULT_UPDATE = 260;

    private String id = "";
    private boolean isProgress = false;

    private ActivityAddUpdateArticleBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityAddUpdateArticleBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        binding.progressBar.setVisibility(View.GONE);
        binding.errorMessage.setVisibility(View.GONE);

        setSupportActionBar(binding.toolbar);

        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);

        Bundle bundle = getIntent().getExtras();
        if (bundle != null) {
            id = bundle.getString(EXTRA_ARTICLE_ID, "");
            String title = bundle.getString(EXTRA_ARTICLE_TITLE, "");
            String content = bundle.getString(EXTRA_ARTICLE_CONTENT,
"");

            String titleWindow;
            if (StringHelper.isNullOrEmpty(id)) {

```

```

        titleWindow = getString(R.string.add_article);
        binding.edtArticleTitle.setText("");
        binding.edtArticleContent.setText("");
    } else {
        titleWindow = getString(R.string.update_article);
        binding.edtArticleTitle.setText(title);
        binding.edtArticleContent.setText(content);
    }
    getSupportActionBar().setTitle(titleWindow);
}

binding.btnCancelError.setOnClickListener(it -> {
    isProgress = false;
    binding.progressMessage.setVisibility(View.GONE);
    binding.errorMessage.setVisibility(View.GONE);
});

binding.btnRetryError.setOnClickListener(it ->
saveArticle());
}

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

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId() == android.R.id.home) {
        super.onBackPressed();
    } else if (item.getItemId() == R.id.menuSave) {
        saveArticle();
    }
    return true;
}

private void saveArticle() {
    if (StringHelper.isNullOrEmpty(id)) {
        isProgress = true;

        binding.progressMessage.setVisibility(View.VISIBLE);

        binding.tvProgressMessage.setText(R.string.save_please_wait);
        binding.errorMessage.setVisibility(View.GONE);

        String articleTitle =
binding.edtArticleTitle.getText().toString();
        String articleContent =
binding.edtArticleContent.getText().toString();

        if (StringHelper.isNullOrEmpty(articleTitle)) {
            Toast.makeText(this,
getString(R.string.title_must_fill), Toast.LENGTH_SHORT).show();
            binding.progressMessage.setVisibility(View.GONE);
            binding.errorMessage.setVisibility(View.GONE);
            isProgress = false;
        } else if (StringHelper.isNullOrEmpty(articleContent)) {
            Toast.makeText(this,

```



```

getString(R.string.content_must_fill), Toast.LENGTH_SHORT).show();
        binding.progressMessage.setVisibility(View.GONE);
        binding.errorMessage.setVisibility(View.GONE);
        isProgress = false;
    } else {
        String url = getString(R.string.server) +
"api/articles";
        CredentialPreference credential = new
CredentialPreference(this);
        AsyncHttpClient client = new AsyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("title", articleTitle);
        params.put("content", articleContent);
        params.put("ustadz_id",
credential.getCredential().getUsername());

        client.post(url, params, new
AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {

binding.progressMessage.setVisibility(View.GONE);

binding.errorMessage.setVisibility(View.GONE);
                Toast.makeText(AddUpdateArticleActivity.this,
getString(R.string.save_success), Toast.LENGTH_SHORT).show();

AddUpdateArticleActivity.this.setResult(RESULT_OK);
                finish();
            }

            @Override
            public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {

binding.progressMessage.setVisibility(View.GONE);

binding.errorMessage.setVisibility(View.VISIBLE);
                String errorMessage = "Oops! An error
occured.\n"
                    + "[" + statusCode + "]" + "
                    + error.getMessage();
                binding.tvErrorMessage.setText(errorMessage);
            }
        });
    }
} else {
    isProgress = true;

    binding.progressMessage.setVisibility(View.VISIBLE);

    binding.tvProgressMessage.setText(R.string.save_please_wait);
    binding.errorMessage.setVisibility(View.GONE);

    String articleTitle =
binding.edtArticleTitle.getText().toString();

```

```

        String articleContent =
binding.edtArticleContent.getText().toString();

        if (StringHelper.isNullOrEmpty(articleTitle)) {
            Toast.makeText(this,
getString(R.string.title_must_fill), Toast.LENGTH_SHORT).show();
            binding.progressMessage.setVisibility(View.GONE);
            binding.errorMessage.setVisibility(View.GONE);
            isProgress = false;
        } else if (StringHelper.isNullOrEmpty(articleContent)) {
            Toast.makeText(this,
getString(R.string.content_must_fill), Toast.LENGTH_SHORT).show();
            binding.progressMessage.setVisibility(View.GONE);
            binding.errorMessage.setVisibility(View.GONE);
            isProgress = false;
        } else {
            String url = getString(R.string.server) +
"api/articles/" + id;
            AsyncHttpClient client = new AsyncHttpClient();

            RequestParams params = new RequestParams();
            params.put("title", articleTitle);
            params.put("content", articleContent);

            client.put(url, params, new
AsyncHttpResponseHandler() {
                @Override
                public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {

                    binding.progressMessage.setVisibility(View.GONE);

                    binding.errorMessage.setVisibility(View.GONE);
                    Toast.makeText(AddUpdateArticleActivity.this,
getString(R.string.update_success), Toast.LENGTH_SHORT).show();

                    AddUpdateArticleActivity.this.setResult(RESULT_OK);
                    finish();
                }

                @Override
                public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {

                    binding.progressMessage.setVisibility(View.GONE);

                    binding.errorMessage.setVisibility(View.VISIBLE);
                    String errorMessage = "Oops! An error
occured.\n"
                        + "[" + statusCode + "] "
                        + error.getMessage();
                    binding.tvErrorMessage.setText(errorMessage);
                }
            });
        }
    }
}

```

```

        @Override
        public void onBackPressed() {
            if (!isProgress) super.onBackPressed();
        }

        @Override
        protected void onDestroy() {
            super.onDestroy();
            binding = null;
        }
    }
}

```

- ReadArticleActivity.kt

```

- package com.kajianid.ustadz.ui.article

import android.content.Intent
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ShareCompat
import androidx.lifecycle.ViewModelProvider
import com.bumptech.glide.Glide
import com.kajianid.ustadz.R
import com.kajianid.ustadz.databinding.ActivityReadArticleBinding
import com.loopj.android.http.AsyncHttpClient
import com.loopj.android.http.AsyncHttpResponseHandler
import cz.msebera.android.httpclient.Header
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import java.util.*

class ReadArticleActivity : AppCompatActivity() {
    private var binding: ActivityReadArticleBinding? = null
    private lateinit var readArticleViewModel: ReadArticleViewModel
    private var id: String? = ""
    private var title = ""
    private var content = ""
    private val isError = false

    companion object {
        const val EXTRA_ARTICLE_ID = "extra_article_id"
        const val RESULT_DELETE = 111
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityReadArticleBinding.inflate(layoutInflater)
        setContentView(binding?.root)
        setSupportActionBar(binding?.toolbar!!)

        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        supportActionBar?.title = getString(R.string.read_article)
    }
}

```

```

binding?.svArticle?.visibility = View.GONE
binding?.progressMessage?.visibility = View.VISIBLE
binding?.errorMessage?.visibility = View.GONE

val bundle = intent.extras
if (bundle != null) {
    id = bundle.getString(EXTRA_ARTICLE_ID)
    readArticleViewModel = ViewModelProvider(this,
        ViewModelProvider.NewInstanceFactory())
        .get(ReadArticleViewModel::class.java)
    readArticleViewModel.article.observe(this, {
        if (it["status"] == true) {
            title = it["title"].toString()
            content = it["content"].toString()
            binding?.contentReadArticle?.tvArticleTitle?.text
            = it["title"].toString()
            if (it["hasImg"] == true) {
                Glide.with(this)
                    .load(
                        getString(R.string.server) +
                            "assets/articles/" +
id + "." +
Objects.requireNonNull(it["extension"]).toString()
                    )
                .into(binding?.contentReadArticle?.imgArticle!!)
            } else {
binding?.contentReadArticle?.imgArticle?.visibility = View.GONE
            }
            val likes = it["like"].toString()
            binding?.contentReadArticle?.tvLikeCount?.text =
likes
            binding?.contentReadArticle?.tvPostDate?.text =
it["post_date"].toString()
            binding?.contentReadArticle?.tvPostContent?.text
            = it["content"].toString()
            binding?.progressMessage?.visibility = View.GONE
            binding?.errorMessage?.visibility = View.GONE
            binding?.svArticle?.visibility = View.VISIBLE
        } else {
            binding?.progressMessage?.visibility = View.GONE
            binding?.errorMessage?.visibility = View.VISIBLE
            val errorMessage = ""
            Error!
            [${it["code"]}]: ${it["message"]}
            """.trimIndent()
            binding?.tvErrorMessage?.text = errorMessage
        }
    })
    readArticleViewModel.setArticleAsync(this, id)
}

binding?.pullToRefresh?.setOnRefreshListener {
    binding?.svArticle?.visibility = View.GONE
    binding?.pullToRefresh?.isRefreshing = false
    binding?.errorMessage?.visibility = View.GONE

```

```

        binding?.progressMessage?.visibility = View.VISIBLE

        GlobalScope.launch(Dispatchers.Main) {
            val deferredArticles = async(Dispatchers.IO) {
                id.let {
                    readArticleViewModel.setArticle(applicationContext, it) }
            }
            val it = deferredArticles.await()
            if (it["status"] == true) {
                title = it["title"].toString()
                content = it["content"].toString()
                binding?.contentReadArticle?.tvArticleTitle?.text
                = it["title"].toString()
                if (it["hasImg"] == true) {
                    Glide.with(applicationContext)

                    .load(resources.getString(R.string.server) +
                        "assets/articles/${it["id"]}.${it["extension"]}")

                    .into(binding?.contentReadArticle?.imgArticle!!)
                } else {

binding?.contentReadArticle?.imgArticle?.visibility = View.GONE
                }
                val likes = it["like"].toString() + "
                ${resources.getString(R.string.likes)}"
                binding?.contentReadArticle?.tvLikeCount?.text =
                likes
                binding?.contentReadArticle?.tvPostDate?.text =
                it["post_date"].toString()
                binding?.contentReadArticle?.tvPostContent?.text
                = it["content"].toString()
                binding?.progressMessage?.visibility = View.GONE
                binding?.svArticle?.visibility = View.VISIBLE
            } else {
                binding?.progressMessage?.visibility = View.GONE
                binding?.errorMessage?.visibility = View.VISIBLE
                val errorMessage = ""
                Error!
                [${it["code"]}] : ${it["message"]}
                """.trimIndent()
                binding?.tvErrorMessage?.text = errorMessage
            }
        }
    }

    binding?.btnRefresh?.setOnClickListener { _ ->
        binding?.svArticle?.visibility = View.GONE
        binding?.errorMessage?.visibility = View.GONE
        binding?.progressMessage?.visibility = View.VISIBLE
        GlobalScope.launch(Dispatchers.Main) {
            val deferredArticles = async(Dispatchers.IO) {
                id.let {
                    readArticleViewModel.setArticle(applicationContext, it) }
            }
            val it = deferredArticles.await()
            if (it["status"] == true) {
                title = it["title"].toString()
                content = it["content"].toString()

```

```

        binding?.contentReadArticle?.tvArticleTitle?.text
= it["title"].toString()
        if (it["hasImg"] == true) {
            Glide.with(applicationContext)

.load(resources.getString(R.string.server) +
"assets/articles/${id}.${it["extension"]}")

.into(binding?.contentReadArticle?.imgArticle!!)
        } else {

binding?.contentReadArticle?.imgArticle?.visibility = View.GONE
        }
        val likes = it["like"].toString() + "
${resources.getString(R.string.likes)}"
        binding?.contentReadArticle?.tvLikeCount?.text =
likes
        binding?.contentReadArticle?.tvPostDate?.text =
it["post_date"].toString()
        binding?.contentReadArticle?.tvPostContent?.text
= it["content"].toString()
        binding?.progressMessage?.visibility = View.GONE
        binding?.svArticle?.visibility = View.VISIBLE
    } else {
        binding?.progressMessage?.visibility = View.GONE
        binding?.errorMessage?.visibility = View.VISIBLE
        val errorMessage = ""
            Error!
            [${it["code"]}] : ${it["message"]}
            """.trimIndent()
        binding?.tvErrorMessage?.text = errorMessage
    }
    }
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
        android.R.id.home -> super.onBackPressed()
        R.id.menuEdit -> {
            if (isError) {
                Toast.makeText(this,
resources.getString(R.string.update_error),
Toast.LENGTH_SHORT).show()
            } else {
                val i = Intent(this,
AddUpdateArticleActivity::class.java)

i.putExtra(AddUpdateArticleActivity.EXTRA_ARTICLE_ID, id)

i.putExtra(AddUpdateArticleActivity.EXTRA_ARTICLE_TITLE, title)

i.putExtra(AddUpdateArticleActivity.EXTRA_ARTICLE_CONTENT, content)
                startActivityForResult(i,
AddUpdateArticleActivity.RESULT_UPDATE)
            }
        }
        R.id.menuDelete -> {
            val alert = AlertDialog.Builder(this)

```

```

        alert.setTitle(resources.getString(R.string.sure))

        .setMessage(resources.getString(R.string.delete_article_message) + "
$title?")

        .setPositiveButton(resources.getString(R.string.yes)) { _, _ ->
            deleteArticle()
        }

        .setNegativeButton(resources.getString(R.string.no)) { _, _ ->
            /* no-op */
        }
        .setCancelable(false)
        val alertDialog = alert.create()
        alertDialog.show()
    }
    R.id.menuShare -> {
        val mimeType = "text/plain"
        ShareCompat.IntentBuilder.from(this).apply {
            setType(mimeType)
            setChooserTitle("Bagikan artikel ini sekarang!")
            setText("""
*${binding?.contentReadArticle?.tvArticleTitle?.text}*
_${binding?.contentReadArticle?.tvPostDate?.text}_
${binding?.contentReadArticle?.tvPostContent?.text}

Informasi ini dibagikan melalui aplikasi
${getString(R.string.app_name)}.
        """.trimIndent())
            startChooser()
        }
    }
}
return true
}

private fun deleteArticle() {
    val url = resources.getString(R.string.server) +
    "api/articles/$id"
    val client = AsyncHttpClient()
    client.delete(url, object : AsyncHttpResponseHandler() {
        override fun onSuccess(statusCode: Int, headers:
Array<out Header>?, responseBody: ByteArray?) {
            Toast.makeText(this@ReadArticleActivity,
resources.getString(R.string.delete_success_message),
Toast.LENGTH_SHORT).show()
            this@ReadArticleActivity.setResult(RESULT_DELETE,
Intent())
            finish()
        }

        override fun onFailure(statusCode: Int, headers:
Array<out Header>?, responseBody: ByteArray?, error: Throwable) {
            Toast.makeText(this@ReadArticleActivity,
                "Oops, an error occurred!\n[${statusCode}
${error.message}\n${responseBody.toString()}",
                Toast.LENGTH_SHORT).show()
        }
    })
}

```

```

    })
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_article_kajian, menu)
    return true
}

override fun onActivityResult(requestCode: Int, resultCode: Int,
data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)

    if (resultCode == RESULT_OK && requestCode ==
AddUpdateArticleActivity.RESULT_UPDATE) {
        binding?.svArticle?.visibility = View.GONE
        binding?.errorMessage?.visibility = View.GONE
        binding?.progressMessage?.visibility = View.VISIBLE
        GlobalScope.launch(Dispatchers.Main) {
            val deferredArticles = async(Dispatchers.IO) {
                id.let {
                    readArticleViewModel.setArticle(applicationContext, it)
                }
                val it = deferredArticles.await()
                if (it["status"] == true) {
                    title = it["title"].toString()
                    content = it["content"].toString()
                    binding?.contentReadArticle?.tvArticleTitle?.text
= it["title"].toString()
                    if (it["hasImg"] == true) {
                        Glide.with(applicationContext)

.load(resources.getString(R.string.server) +
"assets/articles/${id}.${it["extension"]}")

.into(binding?.contentReadArticle?.imgArticle!!)
                    } else {
                        binding?.contentReadArticle?.imgArticle?.visibility = View.GONE
                    }
                    val likes = it["like"].toString() + "
${resources.getString(R.string.likes)}"
                    binding?.contentReadArticle?.tvLikeCount?.text =
likes
                    binding?.contentReadArticle?.tvPostDate?.text =
it["post_date"].toString()
                    binding?.contentReadArticle?.tvPostContent?.text
= it["content"].toString()
                    binding?.progressMessage?.visibility = View.GONE
                    binding?.svArticle?.visibility = View.VISIBLE
                } else {
                    binding?.progressMessage?.visibility = View.GONE
                    binding?.errorMessage?.visibility = View.VISIBLE
                    val errorMessage = ""
                    Error!
                    [${it["code"]}]: ${it["message"]}
                    """.trimIndent()
                    binding?.tvErrorMessage?.text = errorMessage
                }
            }
        }
    }
}

```



```

    }
}
}

```

```

- ReadArticleViewModel.java
- package com.kajianid.ustadz.ui.article;

import android.content.Context;
import android.os.Looper;

import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.kajianid.ustadz.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

import cz.msebera.android.httpclient.Header;

public class ReadArticleViewModel extends ViewModel {

    private final MutableLiveData<Map<String, Object>> mData = new
    MutableLiveData<>();

    public LiveData<Map<String, Object>> getArticle() {
        return mData;
    }

    public Map<String, Object> setArticle(Context context, String id)
    {
        try {
            Looper.prepare();
        } catch (Exception ignored) {

        }
        Map<String, Object> resultMap = new HashMap<>();

        String url = context.getString(R.string.server) +
        "api/articles";
        CredentialPreference credential = new
        CredentialPreference(context);
        SyncHttpClient client = new SyncHttpClient();

```

```

        RequestParams params = new RequestParams();
        params.put("read", "1");
        params.put("ustadz_mode", "1");
        params.put("ustadz_id",
credential.getCredential().getUsername());
        params.put("article", id);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");
                    String like = resultObject.getString("likes");

                    JSONObject articleJSONObject =
list.getJSONObject(0);
                    String title =
articleJSONObject.getString("title");
                    boolean hasImg =
StringHelper.convertToBoolean(articleJSONObject.getString("has_img"))
;
                    String extension =
articleJSONObject.getString("extension");
                    String content =
articleJSONObject.getString("content");

                    String date =
articleJSONObject.getString("post_date");
                    Date datePost = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss", Locale.getDefault()).parse(date);
                    assert datePost != null;
                    String datePostFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(datePost);

                    resultMap.put("status", true);
                    resultMap.put("id", id);
                    resultMap.put("title", title);
                    resultMap.put("hasImg", hasImg);
                    resultMap.put("extension", extension);
                    resultMap.put("content", content);
                    resultMap.put("post_date", datePostFormatted);
                    resultMap.put("like", like);
                    mData.postValue(resultMap);
                } catch (JSONException | ParseException e) {
                    resultMap.put("status", false);
                    resultMap.put("code", 0);
                    resultMap.put("message", e.getLocalizedMessage()
+ " id: " + id);
                    mData.postValue(resultMap);
                    e.printStackTrace();
                }
            }
        }
    }

```

```

        @Override
        public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
            resultMap.put("status", false);
            resultMap.put("code", statusCode);
            resultMap.put("message", error.getLocalizedMessage()
+ " id: " + id);
            mData.postValue(resultMap);
        }
    });

    return resultMap;
}

    public Map<String, Object> setArticleAsync(Context context,
String id) {
        Map<String, Object> resultMap = new HashMap<>();

        String url = context.getString(R.string.server) +
"api/articles";
        CredentialPreference credential = new
CredentialPreference(context);
        AsyncHttpClient client = new AsyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("read", "1");
        params.put("ustadz_mode", "1");
        params.put("ustadz_id",
credential.getCredential().getUsername());
        params.put("article", id);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");
                    String like = resultObject.getString("likes");

                    JSONObject articleJSONObject =
list.getJSONObject(0);
                    String title =
articleJSONObject.getString("title");
                    boolean hasImg =
StringHelper.convertToBoolean(articleJSONObject.getString("has_img"));
                    ;
                    String extension =
articleJSONObject.getString("extension");
                    String content =
articleJSONObject.getString("content");

                    String date =
articleJSONObject.getString("post_date");
                    Date datePost = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss", Locale.getDefault()).parse(date);
                    assert datePost != null;

```

```

        String datePostFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(datePost);

        resultMap.put("status", true);
        resultMap.put("id", id);
        resultMap.put("title", title);
        resultMap.put("hasImg", hasImg);
        resultMap.put("extension", extension);
        resultMap.put("content", content);
        resultMap.put("post_date", datePostFormatted);
        resultMap.put("like", like);
        mData.postValue(resultMap);
    } catch (JSONException | ParseException e) {
        resultMap.put("status", false);
        resultMap.put("code", 0);
        resultMap.put("message", e.getLocalizedMessage()
+ " id: " + id);
        mData.postValue(resultMap);
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
    resultMap.put("status", false);
    resultMap.put("code", statusCode);
    resultMap.put("message", error.getLocalizedMessage()
+ " id: " + id);
    mData.postValue(resultMap);
}
});

return resultMap;
}
}

```

```

- EditIdentityActivity.java
- package com.kajianid.ustadz.ui.credential;

import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Ustadz;
import com.kajianid.ustadz.databinding.ActivityEditIdentityBinding;
import com.kajianid.ustadz.preferences.CredentialPreference;
import com.kajianid.ustadz.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;

import cz.msebera.android.httpclient.Header;

```

```

public class EditIdentityActivity extends AppCompatActivity {

    public static final String EXTRA_PARCEL_USTADZ =
"extra_parcel_ustadz";

    private ActivityEditIdentityBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityEditIdentityBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setFinishOnTouchOutside(false);

        binding.rlButtons.setVisibility(View.VISIBLE);
        binding.progressBar.setVisibility(View.GONE);
        binding.btnCancel.setOnClickListener(it -> finish());

        Bundle bundle = getIntent().getExtras();
        if (bundle != null) {
            Ustadz ustadz =
bundle.getParcelable(EXTRA_PARCEL_USTADZ);

            String name = ustadz.getName();
            String phone = ustadz.getPhone();
            String address = ustadz.getAddress();
            String email = ustadz.getEmail();
            String gender = ustadz.getGender();

            binding.edtUstadzName.setText(name);
            binding.edtPhoneNumber.setText(phone);
            binding.edtUstadzAddress.setText(address);
            binding.edtEmailAddress.setText(email);
            if (gender.equalsIgnoreCase("L"))
                binding.rgGender.check(R.id.rdMale);
            else
                binding.rgGender.check(R.id.rdFemale);

            binding.btnChange.setOnClickListener(it -> {
                boolean isError = false;
                if
(StringHelper.isNullOrEmpty(binding.edtUstadzName.getText().toString(
))) {
                    isError = true;

                binding.edtUstadzName.setError(getString(R.string.require_to_fill));
                }
                if
(StringHelper.isNullOrEmpty(binding.edtEmailAddress.getText().toStrin
g())) {
                    isError = true;

                binding.edtUstadzName.setError(getString(R.string.require_to_fill));
                } else if
(!StringHelper.isValidEmail(binding.edtEmailAddress.getText().toStrin
g())) {
                    isError = true;

```

```

binding.edtEmailAddress.setError(getString(R.string.invalid_email));
    }
    if
    (StringHelper.isNullOrEmpty(binding.edtPhoneNumber.getText().toString()
    ())) {
        isError = true;

binding.edtPhoneNumber.setError(getString(R.string.require_to_fill));
    }
    if (!isError) {
        CredentialPreference preference = new
        CredentialPreference(this);
        binding.rlButtons.setVisibility(View.GONE);
        binding.progressBar.setVisibility(View.VISIBLE);

        String ustadzGender =
        (binding.rgGender.getCheckedRadioButtonId() == R.id.rdMale) ? "L" :
        "P";

        String api = getString(R.string.server) +
        "api/ustadz/identity/" + preference.getCredential().getUsername();
        AsyncHttpClient client = new AsyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("name",
binding.edtUstadzName.getText().toString());
        params.put("phone",
binding.edtPhoneNumber.getText().toString());
        params.put("gender", ustadzGender);
        params.put("address",
binding.edtUstadzAddress.getText().toString());
        params.put("email",
binding.edtEmailAddress.getText().toString());

        client.put(api, params, new
        AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode,
Header[] headers, byte[] responseBody) {
                Toast.makeText(EditIdentityActivity.this,
getString(R.string.success_modify_identity),
Toast.LENGTH_SHORT).show();
                EditIdentityActivity.this.finish();
            }

            @Override
            public void onFailure(int statusCode,
Header[] headers, byte[] responseBody, Throwable error) {
                Toast.makeText(EditIdentityActivity.this,
getString(R.string.failed_modify_identity) + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
});
}

@Override
protected void onDestroy() {

```

```

        super.onDestroy();
        binding = null;
    }
}

```

- EditPasswordActivity.java

```

- package com.kajianid.ustadz.ui.credential;

import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Credential;
import com.kajianid.ustadz.databinding.ActivityEditPasswordBinding;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;

import cz.msebera.android.httpclient.Header;

public class EditPasswordActivity extends AppCompatActivity {

    private ActivityEditPasswordBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityEditPasswordBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setFinishOnTouchOutside(false);

        binding.btnCancel.setOnClickListener(it -> finish());

        binding.btnChange.setOnClickListener(it -> {
            CredentialPreference preference = new
CredentialPreference(this);
            String oldPassword =
preference.getCredential().getPassword();

            binding.edtOldPassword.setError(null);
            binding.edtNewPassword.setError(null);
            binding.edtConfirmPassword.setError(null);

            if
(binding.edtOldPassword.getText().toString().equals(oldPassword)) {
                if
(binding.edtNewPassword.getText().toString().length() < 6) {
                    // new password kurang dari 6

binding.edtNewPassword.setError(getString(R.string.new_password_not_m
eet_minimum));
                }
            }
        });
    }
}

```

```

(binding.edtConfirmPassword.getText().toString().length() < 6) {
    // confirm password kurang dari 6

binding.edtConfirmPassword.setError(getString(R.string.confirm_password_not_meet_minimum));
    }
    if
((binding.edtNewPassword.getText().toString().length() >= 6) &&
(binding.edtConfirmPassword.getText().toString().length() >= 6)) {
    if
(binding.edtNewPassword.getText().toString().equals(
binding.edtConfirmPassword.getText().toString()
)) {
        binding.rlButtons.setVisibility(View.GONE);

binding.progressBar.setVisibility(View.VISIBLE);

        String api = getString(R.string.server) +
"api/ustadz/credential";
        AsyncHttpClient client = new
AsyncHttpClient();
        RequestParams params = new RequestParams();
        params.put("username",
preference.getCredential().getUsername());
        params.put("old_pass",
binding.edtOldPassword.getText().toString());
        params.put("new_pass",
binding.edtNewPassword.getText().toString());

        client.put(api, params, new
AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode,
Header[] headers, byte[] responseBody) {
                Credential credential = new
Credential();

                credential.setUsername(preference.getCredential().getUsername());

                credential.setPassword(binding.edtNewPassword.getText().toString());

                Toast.makeText(EditPasswordActivity.this,
getString(R.string.success_change_pass), Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(int statusCode,
Header[] headers, byte[] responseBody, Throwable error) {

                Toast.makeText(EditPasswordActivity.this,
getString(R.string.failed_change_pass) + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    } else {

```



```

binding.edtConfirmPassword.setError(getString(R.string.confirm_password_not_same));
    }
    }
} else {

binding.edtOldPassword.setError(getString(R.string.wrong_old_pass));
}
});
}

@Override
protected void onDestroy() {
    super.onDestroy();
    binding = null;
}
}

```

```

- IntroActivity.kt
- package com.kajianid.ustadz.ui.intro

import android.Manifest
import android.content.Context
import android.content.Intent
import android.content.SharedPreferences
import android.graphics.Color
import android.os.Build
import android.os.Bundle
import android.widget.Toast
import androidx.annotation.RequiresApi
import androidx.fragment.app.Fragment
import com.github.appintro.AppIntro2
import com.github.appintro.AppIntroFragment
import com.github.appintro.AppIntroPageTransformerType
import com.kajianid.ustadz.R
import com.kajianid.ustadz.ui.login.LoginActivity

class IntroActivity : AppIntro2() {

    @RequiresApi(Build.VERSION_CODES.M)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Make sure you don't call setContentView!
        // Call addSlide passing your Fragments.
        // You can use AppIntroFragment to use a pre-built fragment
        addSlides()
        // Ask for required permission
        askForPermissions(
            permissions =
                arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE),
            slideNumber = 1, // first slide
            required = true)

        /* APPEARANCES */
        // Hide/Show the status Bar
        showStatusBar(true)
    }
}

```

```

        // Wizard Mode
        isWizardMode = true

        // You can customize your parallax parameters in the
        constructors.
        setTransformer(AppIntroPageTransformerType.Parallax(
            titleParallaxFactor = 1.0,
            imageParallaxFactor = -1.0,
            descriptionParallaxFactor = 2.0
        ))

        // Toggle Indicator Visibility
        isIndicatorEnabled = true
        // Change Indicator Color
        setIndicatorColor(
            selectedIndicatorColor = getColor(R.color.biru),
            unselectedIndicatorColor = getColor(R.color.abuabu)
        )
    }

    private fun addSlides() {
        addSlide(AppIntroFragment.newInstance(
            title = "Jalin hubungan bersama masyarakat",
            description = ""
                Buatlah pengumuman dan informasi terkait kajian
                yang akan Anda adakan!
            """).trimIndent(),
            titleColor = Color.BLACK,
            imageDrawable = R.drawable.intro1,
            descriptionColor = Color.BLACK,
            backgroundColor = Color.WHITE
        ))
        addSlide(AppIntroFragment.newInstance(
            title = "Bantu mereka meningkatkan pengetahuannya",
            description = ""
                Anda juga dapat membuat posting Islami yang akan
                disebarkan ke seluruh kalangan masyarakat melalui Kajian.ID!
            """).trimIndent(),
            titleColor = Color.BLACK,
            imageDrawable = R.drawable.intro3,
            descriptionColor = Color.BLACK,
            backgroundColor = Color.WHITE
        ))
    }

    override fun onUserDeniedPermission(permissionName: String) {
        // User pressed "Deny" on the permission dialog
        Toast.makeText(
            applicationContext,
            "Maaf, kami perlu izin untuk membaca berkas agar kami
            bisa mengunggah foto atau poster kajian ke server kami!",
            Toast.LENGTH_SHORT
        ).show()
    }

    override fun onDonePressed(currentFragment: Fragment?) {
        super.onDonePressed(currentFragment)
        // Decide what to do when the user clicks on "Done"
    }

```

```

        val sp: SharedPreferences =
getSharedPreferences("root_preferences", Context.MODE_PRIVATE)
        with(sp.edit()) {
            putBoolean("first", true)
            commit()
        }
        val home = Intent(this@IntroActivity,
LoginActivity::class.java)
        startActivity(home)
        finish()
    }
}

```

- SplashActivity.java

```

- package com.kajianid.ustadz.ui.intro;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.kajianid.ustadz.R;
import com.kajianid.ustadz.databinding.ActivitySplashBinding;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.kajianid.ustadz.ui.login.LoginActivity;
import com.kajianid.ustadz.ui.main.MainActivity;
import com.kajianid.ustadz.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;

import org.json.JSONException;
import org.json.JSONObject;

import cz.msebera.android.httpclient.Header;

public class SplashActivity extends AppCompatActivity {

    private ActivitySplashBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySplashBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        Glide.with(this)
            .load(R.drawable.icon)
            .into(binding.namaSlogan);

        SharedPreferences sharedPreferences =

```

```

getSharedPreferences("root_preferences", Context.MODE_PRIVATE);
    switch (sharedPreferences.getString("theme", "light")) {
        case "dark":

AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
            break;
        case "light":

AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);
            break;
        case "auto":

AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_FOLLOW_SYSTEM);
            break;
    }
    int loadTime = 2000;

    CredentialPreference credentialPreference = new
CredentialPreference(this);
    binding.progressBar.setVisibility(View.VISIBLE);
    String version = "Versi" + getString(R.string.app_version);
    binding.version.setText(version);

    new Handler(getMainLooper()).postDelayed(() -> {
        if (sharedPreferences.getBoolean("first", false)) {
            fetchLoginData(

credentialPreference.getCredential().getUsername(),

credentialPreference.getCredential().getPassword()
            );
        } else {
            Intent intro = new Intent(SplashActivity.this,
IntroActivity.class);
            startActivity(intro);
            SplashActivity.this.finish();
        }
    }, loadTime);
}

private void fetchLoginData(String username, String password) {
    if (!StringHelper.isNullOrEmpty(username) ||
!StringHelper.isNullOrEmpty(password)) {
        String api = getString(R.string.server) +
"api/ustadz/credential";
        AsyncHttpClient client = new AsyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("username", username);
        params.put("password", password);

        client.post(api, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
                try {

```

```

        String response = new String(responseBody);
        JSONObject responseObject = new
JSONObject(response);

        String getUsername =
responseObject.getString("username");
        Toast.makeText(
            SplashActivity.this,
            getString(R.string.welcome) +
getUsername,
            Toast.LENGTH_SHORT
        ).show();

        Intent i = new Intent(SplashActivity.this,
MainActivity.class);
        startActivity(i);
        SplashActivity.this.finish();
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
    switch (statusCode) {
        case 401:
            Toast.makeText(
                SplashActivity.this,
                getString(R.string.wrong_username_password),
                Toast.LENGTH_SHORT
            ).show();
            break;
        case 403:
            Toast.makeText(
                SplashActivity.this,
                getString(R.string.empty_username_password),
                Toast.LENGTH_SHORT
            ).show();
            break;
        default:
            Toast.makeText(
                SplashActivity.this,
                error.getLocalizedMessage(),
                Toast.LENGTH_SHORT
            ).show();
            break;
    }
    Intent i = new Intent(SplashActivity.this,
LoginActivity.class);
    startActivity(i);
    SplashActivity.this.finish();
}
    });
} else {
    Intent login = new Intent(SplashActivity.this,
LoginActivity.class);

```

```

        startActivity(login);
        SplashActivity.this.finish();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    binding = null;
}
}

```

- AddUpdateKajianActivity.kt

```

- package com.kajianid.ustadz.ui.kajian

import android.Manifest
import android.app.DatePickerDialog
import android.app.DatePickerDialog.OnDateSetListener
import android.app.TimePickerDialog
import android.app.TimePickerDialog.OnTimeSetListener
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Bundle
import android.os.Parcelable
import android.provider.MediaStore
import android.util.Log
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.*
import android.widget.AdapterView.OnItemClickListener
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import com.bumptech.glide.Glide
import com.kajianid.ustadz.R
import com.kajianid.ustadz.data.Kajian
import com.kajianid.ustadz.data.Mosque
import com.kajianid.ustadz.databinding.ActivityAddUpdateKajianBinding
import com.kajianid.ustadz.prefs.CredentialPreference
import com.kajianid.ustadz.ui.mosque.MosqueChooserActivity
import com.kajianid.ustadz.utils.StringHelper
import com.loopj.android.http.AsyncHttpClient
import com.loopj.android.http.AsyncHttpResponseHandler
import com.loopj.android.http.RequestParams
import cz.msebera.android.httpclient.Header
import java.io.File
import java.io.FileNotFoundException
import java.text.ParseException
import java.text.SimpleDateFormat
import java.util.*

class AddUpdateKajianActivity : AppCompatActivity() {
    private val calendar = Calendar.getInstance()
    private var binding: ActivityAddUpdateKajianBinding? = null
    private var isEditMode = false
    private val kajian = Kajian()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =

```

```

ActivityAddUpdateKajianBinding.inflate(layoutInflater)
    setContentView(binding?.root)
    binding?.progressBar?.visibility = View.GONE
    binding?.errorMessage?.visibility = View.GONE
    setSupportActionBar(binding!!.toolbar)
    supportActionBar?.title = getString(R.string.add_kajian)
    supportActionBar?.setDisplayHomeAsUpEnabled(true)
    val bundle = intent.extras
    if (bundle != null) {
        supportActionBar?.title =
            getString(R.string.update_kajian)
        val params =
            bundle.getParcelable<Parcelable>(EXTRA_PARCEL_KAJIAN) as Kajian?
        isEditMode = true
        kajian.id = params?.id
        kajian.title = params?.title
        kajian.date = params?.date
        kajian.address = params?.address
        kajian.place = params?.place
        kajian.description = params?.description
        kajian.ytLink = params?.ytLink
        kajian.imgResource = params?.imgResource
        kajian.mosqueId = params?.mosqueId
        kajian.mosqueName = params?.mosqueName
        when (kajian.place) {
            "Di Tempat" -> {
                binding?.spKajianCategory?.setSelection(0)
                binding?.rlKajianImage?.visibility = View.VISIBLE
                binding?.tilKajianAddress?.visibility =
                    View.VISIBLE
                binding?.tilKajianYtLink?.visibility = View.GONE
                if (kajian.imgResource != null) {
                    Glide.with(this)
                        .load(kajian.imgResource)
                        .into(binding!!.imgKajian)
                }

                binding?.edtKajianAddress?.setText(kajian.address)
            }
            "Video" -> {
                binding?.spKajianCategory?.setSelection(1)
                binding?.rlKajianImage?.visibility = View.GONE
                binding?.tilKajianAddress?.visibility = View.GONE
                binding?.tilKajianYtLink?.visibility =
                    View.VISIBLE
                binding?.edtKajianYtLink?.setText(kajian.ytLink)
            }
            "Live Streaming" -> {
            }
        }
        binding?.edtKajianTitle?.setText(kajian.title)
        binding?.tvMosqueId?.text = kajian.mosqueId
        binding?.tvMosqueName?.text = kajian.mosqueName

        binding?.edtKajianDescription?.setText(kajian.description)
        try {
            val dateParse = SimpleDateFormat("yyyy-MM-dd HH:mm",
                Locale.getDefault()).parse(kajian.date)!!
            val date = SimpleDateFormat("yyyy-MM-dd",

```

```

Locale.getDefault()).format(dateParse)
        val time = SimpleDateFormat("HH:mm",
Locale.getDefault()).format(dateParse)
        binding?.edtKajianDateDue?.setText(date)
        binding?.edtKajianTimeDue?.setText(time)
    } catch (e: ParseException) {
        e.printStackTrace()
    }
    binding?.spKajianCategory?.isEnabled = false
    binding?.btnChooseKajianImage?.isEnabled = false
}
binding?.btnChooseMosque?.setOnClickListener { it: View? ->
    val i = Intent(this, MosqueChooserActivity::class.java)
    i.putExtra(MosqueChooserActivity.EXTRA_IS_USTADZ_ONLY,
true)
        startActivityForResult(i,
MosqueChooserActivity.REQUEST_MOSQUE)
    }
    binding?.btnChooseKajianImage?.setOnClickListener { it: View?
->
        if (ContextCompat.checkSelfPermission(
            this,
            Manifest.permission.READ_EXTERNAL_STORAGE
        ) == PackageManager.PERMISSION_GRANTED) {
            val i = Intent(
                Intent.ACTION_PICK,
                MediaStore.Images.Media.EXTERNAL_CONTENT_URI
            )
            startActivityForResult(i, RESULT_LOAD_IMAGE)
        } else {
            Toast.makeText(
                this,
                getString(R.string.file_permission_denied),
                Toast.LENGTH_LONG
            ).show()
        }
    }
    binding?.btnRetryError?.setOnClickListener { it: View? ->
        try {
            postData()
        } catch (e: FileNotFoundException) {
            e.printStackTrace()
        }
    }
    binding?.btnCancelError?.setOnClickListener { it: View? ->
        binding?.progressBar?.visibility = View.GONE
        binding?.errorMessage?.visibility = View.GONE
    }
    binding?.spKajianCategory?.onItemSelectedListener = object :
OnItemSelectedListener {
        override fun onItemSelected(parent: AdapterView<*>?,
view: View, position: Int, id: Long) {
            if (position == 0) {
                binding?.rlKajianImage?.visibility = View.VISIBLE
                binding?.tilKajianAddress?.visibility =
View.VISIBLE
                binding?.tilKajianYtLink?.visibility = View.GONE
            } else {
                binding?.rlKajianImage?.visibility = View.GONE

```



```

binding?.tilKajianAddress?.visibility = View.GONE
binding?.tilKajianYtLink?.visibility =
View.VISIBLE
    }
}

    override fun onNothingSelected(parent: AdapterView<*>?) {
        // no-op
    }
}
val dateSetListener = OnDateSetListener { view: DatePicker?,
year: Int, month: Int, dayOfMonth: Int ->
    calendar[Calendar.YEAR] = year
    calendar[Calendar.MONTH] = month
    calendar[Calendar.DAY_OF_MONTH] = dayOfMonth
    val dateFormat = "yyyy-MM-dd"
    val sdf = SimpleDateFormat(dateFormat,
Locale.getDefault())

binding?.edtKajianDateDue?.setText(sdf.format(calendar.time))
}
binding?.edtKajianDateDue?.setOnClickListener { it: View? ->
    val dpd = DatePickerDialog(
        this,
        dateSetListener,
        calendar[Calendar.YEAR],
        calendar[Calendar.MONTH],
        calendar[Calendar.DAY_OF_MONTH]
    )
    dpd.setTitle(getString(R.string.choose_date_due))
    dpd.datePicker.minDate = System.currentTimeMillis() -
1000 // minimal hari ini
    dpd.show()
}
binding?.edtKajianTimeDue?.setOnClickListener { it: View? ->
    val mcurrentTime = Calendar.getInstance()
    val hourNow = mcurrentTime[Calendar.HOUR_OF_DAY]
    val minuteNow = mcurrentTime[Calendar.MINUTE]
    val timeSetListener = OnTimeSetListener { view:
TimePicker?, hourOfDay: Int, minute: Int ->
        val timeCalendarSet = Calendar.getInstance()
        timeCalendarSet[Calendar.HOUR_OF_DAY] = hourOfDay
        timeCalendarSet[Calendar.MINUTE] = minute
        val timeSet = SimpleDateFormat("HH:mm",
Locale.getDefault()).format(timeCalendarSet.time)
        binding?.edtKajianTimeDue?.setText(timeSet)
    }
    val tpd = TimePickerDialog(
        this,
        timeSetListener,
        hourNow,
        minuteNow,
        true
    )
    tpd.setTitle(getString(R.string.choose_time_due))
    tpd.show()
}
}

```

```

        override fun onCreateOptionsMenu(menu: Menu): Boolean {
            menuInflater.inflate(R.menu.menu_add_update, menu)
            return true
        }

        override fun onOptionsItemSelected(item: MenuItem): Boolean {
            if (item.itemId == android.R.id.home) super.onBackPressed()
            else if (item.itemId == R.id.menuSave) {
                try {
                    postData()
                } catch (e: FileNotFoundException) {
                    e.printStackTrace()
                }
            }
            return super.onOptionsItemSelected(item)
        }

        override fun onActivityResult(requestCode: Int, resultCode: Int,
            data: Intent?) {
            when (resultCode) {
                MosqueChooserActivity.REQUEST_MOSQUE -> if (data != null)
            {
                val mosque =
                data.getParcelableExtra<Mosque>(MosqueChooserActivity.EXTRA_MOSQUE_RE
                RESULT)!!
                binding?.tvMosqueId?.text = mosque.id.toString()
                binding?.tvMosqueName?.text = mosque.mosqueName
            }
            RESULT_OK -> if (requestCode == RESULT_LOAD_IMAGE && data
            != null) {
                val selectedImage = data.data
                Glide.with(this)
                    .load(selectedImage)
                    .into(binding?.imgKajian!!)
                val path = selectedImage!!.path!!.substring(5)
                binding?.tvKajianImagePath?.text = path
            }
        }
        super.onActivityResult(requestCode, resultCode, data)
    }

    @Throws(FileNotFoundException::class)
    private fun postData() {
        if (!isEditMode) {
            // save
            val selectedItem =
            binding!!.spKajianCategory.selectedItem.toString()
            val categories =
            resources.getStringArray(R.array.kajian_category_entries)
            var kajianTitle: String?
            var kajianAddress: String? = null
            var kajianYtLink: String? = null
            var kajianDescription: String?
            var kajianDateDue: String?
            var kajianTimeDue: String?
            if (selectedItem == categories[0]) {
                if
                (StringHelper.isNullOrEmpty(binding!!.edtKajianAddress.text.toString(
                ))) {

```

```

        Toast.makeText(this,
getString(R.string.kajian_address_empty), Toast.LENGTH_SHORT).show()
        binding!!.edtKajianAddress.requestFocus()
        return
    }
} else {
    if
    (StringHelper.isNullOrEmpty(binding!!.edtKajianYtLink.text.toString()
)) {
        Toast.makeText(this,
getString(R.string.kajian_yt_link_empty), Toast.LENGTH_SHORT).show()
        binding!!.edtKajianAddress.requestFocus()
        return
    }
}
    if
    (StringHelper.isNullOrEmpty(binding!!.edtKajianTitle.text.toString()
)) {
        Toast.makeText(this,
getString(R.string.kajian_title_empty), Toast.LENGTH_SHORT).show()
        binding!!.edtKajianTitle.requestFocus()
        return
    }
    if (binding!!.tvMosqueId.text.toString() ==
getString(R.string.example_id)) {
        Toast.makeText(this,
getString(R.string.kajian_mosque_empty), Toast.LENGTH_SHORT).show()
        binding!!.btnChooseMosque.requestFocus()
        return
    }
    if
    (StringHelper.isNullOrEmpty(binding!!.edtKajianDescription.text.toStr
ing())) {
        Toast.makeText(this,
getString(R.string.kajian_desc_empty), Toast.LENGTH_SHORT).show()
        binding!!.edtKajianTitle.requestFocus()
        return
    }
    if
    (StringHelper.isNullOrEmpty(binding!!.edtKajianDateDue.text.toString(
)) ||
StringHelper.isNullOrEmpty(binding!!.edtKajianTimeDue.text.toString()
)) {
        Toast.makeText(this,
getString(R.string.kajian_due_empty), Toast.LENGTH_SHORT).show()

        binding!!.svAddUpdateKajian.fullScroll(ScrollView.FOCUS_DOWN)
        return
    }
    binding!!.progressMessage.visibility = View.VISIBLE
    binding!!.errorMessage.visibility = View.GONE

    val kajianCategory: String
    kajianCategory = if (selectedItem == categories[0]) "Di
Tempat" else if (selectedItem == categories[1]) "Video" else if
(selectedItem == categories[2]) "Live Streaming" else throw
IllegalArgumentException("Invalid Kajian Category!")
    kajianTitle = binding!!.edtKajianTitle.text.toString()

```

```

        if (selectedItem == categories[0]) kajianAddress =
binding!!.edtKajianAddress.text.toString() else kajianYtLink =
binding!!.edtKajianYtLink.text.toString()
        val mosqueId =
binding!!.tvMosqueId.text.toString().toInt()
        kajianDescription =
binding!!.edtKajianDescription.text.toString()
        kajianDateDue =
binding!!.edtKajianDateDue.text.toString()
        kajianTimeDue =
binding!!.edtKajianTimeDue.text.toString()
        val kajianDateTimeDue = "$kajianDateDue $kajianTimeDue"

        val api = getString(R.string.server) + "api/kajian"
        val client = AsyncHttpClient()
        val credential = CredentialPreference(this)

        val params = RequestParams()
        params.put("title", kajianTitle)
        params.put("ustadz_id", credential.credential.username)
        params.put("mosque_id", mosqueId)
        if (selectedItem == categories[0]) {
            params.put("address", kajianAddress)
            if (binding!!.tvKajianImagePath.text.toString() !=
getString(R.string.example_path)) {
                val file =
File(binding!!.tvKajianImagePath.text.toString())
                if (file.isFile) params.put("file", file) else
Toast.makeText(this, "Not found: " + file.absolutePath,
Toast.LENGTH_LONG).show()
            }
        } else params.put("yt_url", kajianYtLink)
        params.put("place", kajianCategory)
        params.put("desc", kajianDescription)
        params.put("due", kajianDateTimeDue)
        client.setTimeout(60000)
        client.post(api, params, object :
AsyncHttpResponseHandler() {
            override fun onSuccess(statusCode: Int, headers:
Array<Header>, responseBody: ByteArray) {
                binding!!.progressMessage.visibility = View.GONE
                binding!!.errorMessage.visibility = View.GONE
                Toast.makeText(this@AddUpdateKajianActivity,
getString(R.string.add_kajian_success), Toast.LENGTH_SHORT).show()

                this@AddUpdateKajianActivity.setResult(RESULT_SAVE)
                finish()
            }

            override fun onFailure(statusCode: Int, headers:
Array<Header>, responseBody: ByteArray, error: Throwable) {
                error.printStackTrace()
                Log.e("ERROR", String(responseBody))
                binding!!.progressMessage.visibility = View.GONE
                binding!!.errorMessage.visibility = View.VISIBLE
                val errorMsg = ""
                Oops! An error occurred.
                [$statusCode]
                ${String(responseBody)}
            }
        })
    }
}

```

```

        """.trimIndent()
        binding!!.tvErrorMessage.text = errorMsg
    }
    })
} else {
    val selectedItem =
binding!!.spKajianCategory.selectedItem.toString()
    val categories =
resources.getStringArray(R.array.kajian_category_entries)
    var kajianTitle: String?
    var kajianAddress: String? = null
    var kajianYtLink: String? = null
    var kajianDescription: String?
    var kajianDateDue: String?
    var kajianTimeDue: String?
    if (selectedItem == categories[0]) {
        if
(StringHelper.isNullOrEmpty(binding!!.edtKajianAddress.text.toString(
))) {
            Toast.makeText(this,
getString(R.string.kajian_address_empty), Toast.LENGTH_SHORT).show()
            binding!!.edtKajianAddress.requestFocus()
            return
        }
    } else {
        if
(StringHelper.isNullOrEmpty(binding!!.edtKajianYtLink.text.toString(
))) {
            Toast.makeText(this,
getString(R.string.kajian_yt_link_empty), Toast.LENGTH_SHORT).show()
            binding!!.edtKajianAddress.requestFocus()
            return
        }
    }
    if
(StringHelper.isNullOrEmpty(binding!!.edtKajianTitle.text.toString(
))) {
        Toast.makeText(this,
getString(R.string.kajian_title_empty), Toast.LENGTH_SHORT).show()
        binding!!.edtKajianTitle.requestFocus()
        return
    }
    if (binding!!.tvMosqueId.text.toString() ==
getString(R.string.example_id)) {
        Toast.makeText(this,
getString(R.string.kajian_mosque_empty), Toast.LENGTH_SHORT).show()
        binding!!.btnChooseMosque.requestFocus()
        return
    }
    if
(StringHelper.isNullOrEmpty(binding!!.edtKajianDescription.text.toStr
ing())) {
        Toast.makeText(this,
getString(R.string.kajian_desc_empty), Toast.LENGTH_SHORT).show()
        binding!!.edtKajianTitle.requestFocus()
        return
    }
    if
(StringHelper.isNullOrEmpty(binding!!.edtKajianDateDue.text.toString(

```

```

)) ||

StringHelper.isNullOrEmpty(binding!!.edtKajianTimeDue.text.toString()
)) {
    Toast.makeText(this,
getString(R.string.kajian_due_empty), Toast.LENGTH_SHORT).show()

binding!!.svAddUpdateKajian.fullScroll(ScrollView.FOCUS_DOWN)
    return
}
binding!!.progressMessage.visibility = View.VISIBLE
binding!!.errorMessage.visibility = View.GONE
val kajianCategory: String
kajianCategory = if (selectedItem == categories[0]) "Di
Tempat" else if (selectedItem == categories[1]) "Video" else if
(selectedItem == categories[2]) "Live Streaming" else throw
IllegalArgumentException("Invalid Kajian Category!")
kajianTitle = binding!!.edtKajianTitle.text.toString()
if (selectedItem == categories[0]) kajianAddress =
binding!!.edtKajianAddress.text.toString() else kajianYtLink =
binding!!.edtKajianYtLink.text.toString()
val mosqueId =
binding!!.tvMosqueId.text.toString().toInt()
kajianDescription =
binding!!.edtKajianDescription.text.toString()
kajianDateDue =
binding!!.edtKajianDateDue.text.toString()
kajianTimeDue =
binding!!.edtKajianTimeDue.text.toString()
val kajianDateTimeDue = "$kajianDateDue $kajianTimeDue"
val api = getString(R.string.server) + "api/kajian/" +
kajian.id
val client = AsyncHttpClient()
val credential = CredentialPreference(this)
val params = RequestParams()
params.put("title", kajianTitle)
params.put("ustadz_id", credential.credential.username)
params.put("mosque_id", mosqueId)
if (selectedItem == categories[0]) {
    params.put("address", kajianAddress)
    if (binding!!.tvKajianImagePath.text.toString() ==
getString(R.string.example_path)) {
        val file =
File(binding!!.tvKajianImagePath.text.toString())
        if (file.isFile) params.put("file", file) else
Toast.makeText(this, "Not found: " + file.absolutePath,
Toast.LENGTH_LONG).show()
    }
} else params.put("yt_url", kajianYtLink)
params.put("place", kajianCategory)
params.put("desc", kajianDescription)
params.put("due", kajianDateTimeDue)
client.setTimeout(60000)
client.put(api, params, object :
AsyncHttpResponseHandler() {
    override fun onSuccess(statusCode: Int, headers:
Array<Header>, responseBody: ByteArray) {
        binding!!.progressMessage.visibility = View.GONE
        binding!!.errorMessage.visibility = View.GONE
    }
}

```

```

        Toast.makeText(this@AddUpdateKajianActivity,
getString(R.string.add_kajian_success), Toast.LENGTH_SHORT).show()

this@AddUpdateKajianActivity.setResult(RESULT_UPDATE)
        finish()
    }

    override fun onFailure(statusCode: Int, headers:
Array<Header>, responseBody: ByteArray, error: Throwable) {
        error.printStackTrace()
        Log.e("ERROR", String(responseBody))
        binding!!.progressMessage.visibility = View.GONE
        binding!!.errorMessage.visibility = View.VISIBLE
        val errorMsg = ""
            Oops! An error occurred.
            [$statusCode]
            ${String(responseBody)}
            """.trimIndent()
        binding!!.tvErrorMessage.text = errorMsg
    }
    })
}

override fun onDestroy() {
    super.onDestroy()
    binding = null
}

companion object {
    private const val RESULT_LOAD_IMAGE = 111
    const val RESULT_SAVE = 130
    const val RESULT_UPDATE = 260
    const val EXTRA_PARCEL_KAJIAN = "extra_parcel_kajian"
}
}

```

- ShowKajianActivity.kt

- package com.kajianid.ustadz.ui.kajian

```

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ShareCompat
import androidx.lifecycle.ViewModelProvider
import com.bumptech.glide.Glide
import com.kajianid.ustadz.R
import com.kajianid.ustadz.data.Kajian
import com.kajianid.ustadz.databinding.ActivityShowKajianBinding
import com.loopj.android.http.AsyncHttpClient
import com.loopj.android.http.AsyncHttpResponseHandler
import cz.msebera.android.httpclient.Header
import kotlinx.coroutines.Dispatchers

```

```

import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import java.util.*

class ShowKajianActivity : AppCompatActivity() {

    private var binding: ActivityShowKajianBinding? = null

    private lateinit var showKajianViewModel: ShowKajianViewModel
    private var kajian: Kajian = Kajian()
    private var id: String = ""

    companion object {
        const val EXTRA_KAJIAN_ID = "extra_kajian_id"

        const val RESULT_DELETE = 121
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityShowKajianBinding.inflate(layoutInflater)
        setContentView(binding?.root)

        binding?.svKajian?.visibility = View.GONE
        binding?.progressMessage?.visibility = View.VISIBLE
        binding?.errorMessage?.visibility = View.GONE

        setSupportActionBar(binding?.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        supportActionBar?.title =
            resources.getString(R.string.view_kajian)

        val bundle = intent.extras
        if (bundle != null) {
            id = bundle.getString(EXTRA_KAJIAN_ID, "").toString()

            showKajianViewModel = ViewModelProvider(this,
                ViewModelProvider.NewInstanceFactory())
                .get(ShowKajianViewModel::class.java)
            showKajianViewModel.kajian.observe(this, {
                if (it["status"] == true) {
                    binding?.progressMessage?.visibility = View.GONE
                    binding?.errorMessage?.visibility = View.GONE

                    val title = it["title"].toString()
                    val ustadzName = resources.getString(R.string.by)
                    + " " + it["ustadz_name"].toString()
                    val mosqueName = it["mosque_name"].toString()
                    val address = it["address"].toString()
                    val description = it["description"].toString()
                    val dateAnnounce =
                        resources.getString(R.string.timestamp_announce) + " " +
                        it["date_announce"].toString()
                    val dateDue =
                        resources.getString(R.string.timestamp_due) + " " + it["date_due"]

                    kajian.id = it["id"].toString()
                    kajian.title = it["title"].toString()
                }
            })
        }
    }
}

```



```

        kajian.mosqueId = it["mosque_id"].toString()
        kajian.mosqueName = it["mosque_name"].toString()
        kajian.place = it["category"].toString()
        kajian.ytLink = it["youtube_link"].toString()
        kajian.description = it["description"].toString()
        kajian.address = it["address"].toString()
        kajian.imgResource =
it["img_resource"].toString()
        kajian.date =
it["date_due_unformatted"].toString()

        binding?.contentShowKajian?.tvKajianTitle?.text =
title
        binding?.contentShowKajian?.tvUstadzName?.text =
ustadzName
        binding?.contentShowKajian?.tvDescription?.text =
description

        binding?.contentShowKajian?.tvTimestampAnnounce?.text = dateAnnounce
        binding?.contentShowKajian?.tvTimestampDue?.text
= dateDue

        if (it["category"] == "Di Tempat") {

binding?.contentShowKajian?.btnPlay?.visibility = View.GONE
        binding?.contentShowKajian?.tvCategory?.text
= it["category"].toString()

binding?.contentShowKajian?.tvMosqueAddress?.text = address
        } else {

binding?.contentShowKajian?.btnPlay?.visibility = View.VISIBLE
        val category =
it["category"].toString().toUpperCase(Locale.ROOT) + " - Courtesy of
YouTube"
        binding?.contentShowKajian?.tvCategory?.text
= category

binding?.contentShowKajian?.tvMosqueAddress?.text = mosqueName
        val uri = it["youtube_link"].toString()

binding?.contentShowKajian?.btnPlay?.setOnClickListener {
        val i = Intent(Intent.ACTION_VIEW,
Uri.parse(uri))
        startActivity(i)
    }

        if (it["img_resource"] != null) {
            Glide.with(this)
                .load(it["img_resource"].toString())
                .into(binding?.contentShowKajian?.imgThumbnail!!)
        } else {

binding?.contentShowKajian?.imgThumbnail?.visibility = View.GONE

binding?.contentShowKajian?.btnPlay?.visibility = View.GONE
        }
    }

```

```

        binding?.svKajian?.visibility = View.VISIBLE
    } else {
        binding?.progressMessage?.visibility = View.GONE
        binding?.errorMessage?.visibility = View.VISIBLE
        val errorMessage = ""
        Error!
        [${it["code"]}] : ${it["message"]}
        """.trimIndent()
        binding?.tvErrorMessage?.text = errorMessage
    }
})
showKajianViewModel.setKajianAsync(this, id)

binding?.pullToRefresh?.setOnRefreshListener {
    refreshData()
}

binding?.btnRefresh?.setOnClickListener {
    refreshData()
}
}

private fun refreshData() {
    binding?.pullToRefresh?.isRefreshing = false
    binding?.svKajian?.visibility = View.GONE
    binding?.progressMessage?.visibility = View.VISIBLE
    binding?.errorMessage?.visibility = View.GONE
    GlobalScope.launch(Dispatchers.Main) {
        val deferredKajian = async(Dispatchers.IO) {
            showKajianViewModel.setKajian(applicationContext, id)
        }
        val it = deferredKajian.await()
        if (it["status"] == true) {
            binding?.progressMessage?.visibility = View.GONE
            binding?.errorMessage?.visibility = View.GONE
            val title = it["title"].toString()
            val ustadzName = resources.getString(R.string.by) + " " + it["ustadz_name"].toString()
            val mosqueName = it["mosque_name"].toString()
            val address = it["address"].toString()
            val description = it["description"].toString()
            val dateAnnounce =
                resources.getString(R.string.timestamp_announce) + " " +
                it["date_announce"].toString()
            val dateDue =
                resources.getString(R.string.timestamp_due) + " " + it["date_due"]

            // write to model class
            kajian.id = it["id"].toString()
            kajian.title = it["title"].toString()
            kajian.mosqueId = it["mosque_id"].toString()
            kajian.mosqueName = it["mosque_name"].toString()
            kajian.place = it["category"].toString()
            kajian.ytLink = it["youtube_link"].toString()
            kajian.description = it["description"].toString()
            kajian.address = it["address"].toString()
            kajian.imgResource = it["img_resource"].toString()
            kajian.date = it["date_due_unformatted"].toString()
        }
    }
}

```

```

        title                binding?.contentShowKajian?.tvKajianTitle?.text =
                                binding?.contentShowKajian?.tvUstadzName?.text =
                                binding?.contentShowKajian?.tvDescription?.text =
                                binding?.contentShowKajian?.tvTimestampAnnounce?.text
                                binding?.contentShowKajian?.tvTimestampDue?.text =
                                dateAnnounce
                                dateDue

                                if (it["category"] == "Di Tempat") {
                                    binding?.contentShowKajian?.btnPlay?.visibility =
View.GONE
                                    binding?.contentShowKajian?.tvCategory?.text =
it["category"].toString()
                                    binding?.contentShowKajian?.tvMosqueAddress?.text
= address
                                } else {
                                    binding?.contentShowKajian?.btnPlay?.visibility =
View.VISIBLE
                                    val category =
it["category"].toString().toUpperCase(Locale.ROOT) + " - Courtesy of
YouTube"
                                    binding?.contentShowKajian?.tvCategory?.text =
category
                                    binding?.contentShowKajian?.tvMosqueAddress?.text
= mosqueName
                                    val uri = it["youtube_link"].toString()

binding?.contentShowKajian?.btnPlay?.setOnClickListener {
                                val i = Intent(Intent.ACTION_VIEW,
Uri.parse(uri))
                                startActivity(i)
                                }
                                }

                                if (it["img_resource"] != null) {
                                    Glide.with(applicationContext)
                                        .load(it["img_resource"].toString())
                                        .into(binding?.contentShowKajian?.imgThumbnail!!)
                                } else {

binding?.contentShowKajian?.imgThumbnail?.visibility = View.GONE
                                binding?.contentShowKajian?.btnPlay?.visibility =
View.GONE
                                }
                                binding?.svKajian?.visibility = View.VISIBLE
                                } else {
                                    binding?.progressMessage?.visibility = View.GONE
                                    binding?.errorMessage?.visibility = View.VISIBLE
                                    val errorMessage = ""
                                    Error!
                                    [${it["code"]}]: ${it["message"]}
                                    """".trimIndent()
                                    binding?.tvErrorMessage?.text = errorMessage
                                }
                                }

```

```

    }
}

private fun deleteKajian() {
    val url = resources.getString(R.string.server) +
"api/kajian/$id"
    val client = AsyncHttpClient()
    client.delete(url, object : AsyncHttpResponseHandler() {
        override fun onSuccess(statusCode: Int, headers:
Array<out Header>?, responseBody: ByteArray?) {
            Toast.makeText(this@ShowKajianActivity,
resources.getString(R.string.delete_kajian_success_message),
Toast.LENGTH_SHORT).show()
            this@ShowKajianActivity.setResult(RESULT_DELETE,
Intent())
            finish()
        }

        override fun onFailure(statusCode: Int, headers:
Array<out Header>?, responseBody: ByteArray?, error: Throwable) {
            Toast.makeText(this@ShowKajianActivity,
"Oops, an error occured!\n[$statusCode]
${error.message}\n${responseBody.toString()}",
Toast.LENGTH_SHORT).show()
        }
    })
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_article_kajian, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
        android.R.id.home -> super.onBackPressed()
        R.id.menuEdit -> {
            val i = Intent(this,
AddUpdateKajianActivity::class.java)
            i.putExtra(AddUpdateKajianActivity.EXTRA_PARCEL_KAJIAN, kajian)
            startActivityForResult(i,
AddUpdateKajianActivity.RESULT_UPDATE)
        }
        R.id.menuDelete -> {
            val alert = AlertDialog.Builder(this)
            alert.setTitle(resources.getString(R.string.sure))

            .setMessage(resources.getString(R.string.delete_kajian_message) + "
$title?")

            .setPositiveButton(resources.getString(R.string.yes)) { _, _ ->
                deleteKajian()
            }

            .setNegativeButton(resources.getString(R.string.no)) { _, _ ->
                /* no-op */
            }
            .setCancelable(false)
        }
    }
}

```

```

        val alertDialog = alert.create()
        alertDialog.show()
    }
    R.id.menuShare -> {
        val mimeType = "text/plain"
        ShareCompat.IntentBuilder.from(this).apply {
            setType(mimeType)
            setChooserTitle("Bagikan kajian ini sekarang!")
            setText("")
        }
        *Kajian ${binding?.contentShowKajian?.tvKajianTitle?.text}*
        Waktu Pelaksanaan:
        ${binding?.contentShowKajian?.tvTimestampDue?.text}

        Kajian ini bersifat ${kajian.place}

        Deskripsi:
        ${kajian.description}

        Informasi ini disebarkan melalui aplikasi
        ${getString(R.string.app_name)}.
        """.trimIndent()
        startChooser()
    }
}
}
return true
}
}

```

- ShowKajianViewModel.java

```

- package com.kajianid.ustadz.ui.kajian;

import android.content.Context;
import android.os.Looper;

import androidx.annotation.NonNull;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

import cz.msebera.android.httpclient.Header;

```

```

public class ShowKajianViewModel extends ViewModel {

    private final MutableLiveData<Map<String, Object>> mData = new
MutableLiveData<>();

    public LiveData<Map<String, Object>> getKajian() {
        return mData;
    }

    public Map<String, Object> setKajian(Context context, String id)
{
        try {
            Looper.prepare();
        } catch (Exception ignored) {

        }

        Map<String, Object> resultMap = new HashMap<>();

        String url = context.getString(R.string.server) +
"api/kajian";
        CredentialPreference credential = new
CredentialPreference(context);
        SyncHttpClient client = new SyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("read", "1");
        params.put("ustadz_mode", "1");
        params.put("ustadz_id",
credential.getCredential().getUsername());
        params.put("kajian", id);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");

                    JSONObject kajianJSONObject =
list.getJSONObject(0);
                    String title =
kajianJSONObject.getString("kajian_title");
                    String ustadzName =
kajianJSONObject.getString("ustadz_name");
                    String mosqueId =
kajianJSONObject.getString("mosque_id");
                    String mosqueName =
kajianJSONObject.getString("mosque_name");
                    String address =
kajianJSONObject.getString("address");
                    String category =
kajianJSONObject.getString("place");
                    String youtubeLink =
kajianJSONObject.getString("youtube_link");
                    String description =

```

```

kajianJSONObject.getString("description");
        String imgResource =
kajianJSONObject.getString("img_resource");
        String dateAnnounce =
kajianJSONObject.getString("date_announce");
        String dateDue =
kajianJSONObject.getString("date_due");

        DateFormat dateAnnouncedD = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        String dateAnnounceFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateAnnouncedD.parse(dateAnnounce));

        DateFormat dateDueD = new SimpleDateFormat("yyyy-
MM-dd HH:mm:ss", Locale.getDefault());
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDueD.parse(dateDue));

        resultMap.put("status", true);
        resultMap.put("id", id);
        resultMap.put("title", title);
        resultMap.put("ustadz_name", ustadzName);
        resultMap.put("mosque_id", mosqueId);
        resultMap.put("mosque_name", mosqueName);
        resultMap.put("address", address);
        resultMap.put("category", category);
        resultMap.put("youtube_link", youtubeLink);
        resultMap.put("description", description);
        resultMap.put("img_resource", imgResource);
        resultMap.put("date_announce",
dateAnnounceFormatted);
        resultMap.put("date_due", dateDueFormatted);
        resultMap.put("date_due_unformatted", dateDue);
        mData.postValue(resultMap);
    } catch (JSONException | ParseException e) {
        resultMap.put("status", false);
        resultMap.put("code", 0);
        resultMap.put("message", e.getLocalizedMessage()
+ " id: " + id);
        mData.postValue(resultMap);
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
    resultMap.put("status", false);
    resultMap.put("code", statusCode);
    resultMap.put("message", error.getLocalizedMessage()
+ " id: " + id);
    mData.postValue(resultMap);
}
});

return resultMap;
}

```

```

        public Map<String, Object> setKajianAsync(@NonNull Context
context, String id) {
            Map<String, Object> resultMap = new HashMap<>();

            String url = context.getString(R.string.server) +
"api/kajian";
            CredentialPreference credential = new
CredentialPreference(context);
            AsyncHttpClient client = new AsyncHttpClient();

            RequestParams params = new RequestParams();
            params.put("read", "1");
            params.put("ustadz_mode", "1");
            params.put("ustadz_id",
credential.getCredential().getUsername());
            params.put("kajian", id);

            client.get(url, params, new AsyncHttpResponseHandler() {
                @Override
                public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                    try {
                        String result = new String(responseBody);
                        JSONObject resultObject = new JSONObject(result);
                        JSONArray list =
resultObject.getJSONArray("data");

                        JSONObject kajianJSONObject =
list.getJSONObject(0);
                        String title =
kajianJSONObject.getString("kajian_title");
                        String ustadzName =
kajianJSONObject.getString("ustadz_name");
                        String mosqueId =
kajianJSONObject.getString("mosque_id");
                        String mosqueName =
kajianJSONObject.getString("mosque_name");
                        String address =
kajianJSONObject.getString("address");
                        String category =
kajianJSONObject.getString("place");
                        String youtubeLink =
kajianJSONObject.getString("youtube_link");
                        String description =
kajianJSONObject.getString("description");
                        String imgResource =
kajianJSONObject.getString("img_resource");
                        String dateAnnounce =
kajianJSONObject.getString("date_announce");
                        String dateDue =
kajianJSONObject.getString("date_due");

                        DateFormat dateAnnounced = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
                        assert dateAnnounced != null;
                        String dateAnnounceFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateAnnounced.parse(dateAnnounce));

```



```

        DateFormat dateDueD = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        assert dateDueD != null;
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDueD.parse(dateDue));

        resultMap.put("status", true);
        resultMap.put("id", id);
        resultMap.put("title", title);
        resultMap.put("ustadz_name", ustadzName);
        resultMap.put("mosque_id", mosqueId);
        resultMap.put("mosque_name", mosqueName);
        resultMap.put("address", address);
        resultMap.put("category", category);
        resultMap.put("youtube_link", youtubeLink);
        resultMap.put("description", description);
        resultMap.put("img_resource", imgResource);
        resultMap.put("date_announce",
dateAnnounceFormatted);
        resultMap.put("date_due", dateDueFormatted);
        resultMap.put("date_due_unformatted", dateDue);
        mData.postValue(resultMap);
    } catch (JSONException | ParseException e) {
        resultMap.put("status", false);
        resultMap.put("code", 0);
        resultMap.put("message", e.getLocalizedMessage()
+ " id: " + id);

        mData.postValue(resultMap);
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
    resultMap.put("status", false);
    resultMap.put("code", statusCode);
    resultMap.put("message", error.getLocalizedMessage()
+ " id: " + id);
    mData.postValue(resultMap);
}
});

return resultMap;
}
}

```

- LoginActivity.java

```

- package com.kajianid.ustadz.ui.login;

import android.content.Intent;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.text.method.HideReturnsTransformationMethod;
import android.text.method.PasswordTransformationMethod;

```

```

import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Credential;
import com.kajianid.ustadz.databinding.ActivityLoginBinding;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.kajianid.ustadz.ui.main.MainActivity;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;

import org.json.JSONException;
import org.json.JSONObject;

import cz.msebera.android.httpclient.Header;

public class LoginActivity extends AppCompatActivity {

    private ActivityLoginBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityLoginBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        Glide.with(this)
            .load(R.drawable.icon)
            .into(binding.contentLogin.namaSlogan3);

        binding.contentLogin.showPass.setOnClickListener(itView -> {
            if (binding.contentLogin.showPass.isChecked())

binding.contentLogin.edtPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
            else

binding.contentLogin.edtPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
        });

        binding.btnRegister.setOnClickListener(it -> {
            String text = "Permisi, saya ingin mengajukan diri\nmendaftar sebagai Ustadz di aplikasi Kajian.ID.\n" +
                "Berikut informasi data diri saya:\n\n" +
                "Nama: \n" +
                "Email: \n" +
                "Nomor Telepon: \n" +
                "Jenis Kelamin: \n" +
                "Alamat Tempat Tinggal: \n" +
                "Password Akun: \n\n" +
                "Terimakasih.";

            Uri whatsappUri = Uri.parse("https://wa.me/" +
                getString(R.string.whatsapp_target) + "?text=" + text);

```

```

        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(whatsappUri);
        startActivity(i);
    });

    binding.btnLogin.setOnClickListener(itView -> {
        binding.btnLogin.setVisibility(View.GONE);
        binding.btnRegister.setVisibility(View.GONE);
        binding.progressBar.setVisibility(View.VISIBLE);

        CredentialPreference preference = new
        CredentialPreference(this);
        String api = getString(R.string.server) +
        "api/ustadz/credential";
        AsyncHttpClient client = new AsyncHttpClient();

        String username =
        binding.contentLogin.edtUsername.getText().toString();
        String password =
        binding.contentLogin.edtPassword.getText().toString();

        RequestParams params = new RequestParams();
        params.put("username", username);
        params.put("password", password);

        client.post(api, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[]
            headers, byte[] responseBody) {
                try {
                    String response = new String(responseBody);
                    JSONObject responseObject = new
                    JSONObject(response);

                    Credential credential = new Credential();
                    credential.setUsername(username);
                    credential.setPassword(password);
                    preference.setCredential(credential);

                    String getUsername =
                    responseObject.getString("username");
                    Toast.makeText(
                        LoginActivity.this,
                        getString(R.string.welcome) +
                        getUsername,
                        Toast.LENGTH_SHORT
                    ).show();

                    Intent i = new Intent(LoginActivity.this,
                    MainActivity.class);
                    startActivity(i);
                    LoginActivity.this.finish();
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        })

        @Override

```

```

        public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
            binding.progressBar.setVisibility(View.GONE);
            binding.btnLogin.setVisibility(View.VISIBLE);
            switch (statusCode) {
                case 403:
                    Toast.makeText(
                        LoginActivity.this,
                        getString(R.string.wrong_username_password),
                        Toast.LENGTH_SHORT
                    ).show();
                    break;
                case 401:
                    Toast.makeText(
                        LoginActivity.this,
                        getString(R.string.empty_username_password),
                        Toast.LENGTH_SHORT
                    ).show();
                    break;
                default:
                    Toast.makeText(
                        LoginActivity.this,
                        error.getLocalizedMessage(),
                        Toast.LENGTH_SHORT
                    ).show();
                    break;
            }
        }
    });
});
}

@Override
protected void onDestroy() {
    super.onDestroy();
    binding = null;
}
}

```

- ArticleAdapter.java

```

- package com.kajianid.ustadz.ui.main.article;

import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.kajianid.ustadz.data.Article;
import com.kajianid.ustadz.databinding.ItemArticlesBinding;
import com.kajianid.ustadz.ui.article.ReadArticleActivity;

import java.util.ArrayList;

```

```

public class ArticleAdapter extends
RecyclerView.Adapter<ArticleAdapter.ArticleViewHolder> {

    private final ArrayList<Article> mData = new ArrayList<>();

    public void setArticles(ArrayList<Article> articles) {
        mData.clear();
        mData.addAll(articles);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public ArticleAdapter.ArticleViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        ItemArticlesBinding binding =
ItemArticlesBinding.inflate(LayoutInflater.from(parent.getContext()))
;
        return new ArticleViewHolder(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull
ArticleAdapter.ArticleViewHolder holder, int position) {
        holder.bind(mData.get(position));
    }

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

    public static class ArticleViewHolder extends
RecyclerView.ViewHolder {

        private final ItemArticlesBinding binding;

        public ArticleViewHolder(@NonNull ItemArticlesBinding
binding) {
            super(binding.getRoot());
            this.binding = binding;
        }

        public void bind(Article article) {
            String title = article.getTitle();
            String content = article.getContent();
            String postDate = article.getPost_date();
            String id = article.getId();

            binding.tvTitle.setText(title);
            binding.tvTextLess.setText(content);
            binding.tvPostDate.setText(postDate);

            if (!article.isHasImg()) {
                binding.imgArticle.setVisibility(View.GONE);
            } else {
                Glide.with(binding.getRoot().getContext())
                    .load(article.getImgUrl())
                    .into(binding.imgArticle);
            }
        }
    }
}

```

```

    }

    binding.getRoot().setOnClickListener(itView -> {
        Intent i = new Intent(binding.getRoot().getContext(),
ReadArticleActivity.class);
        i.putExtra(ReadArticleActivity.EXTRA_ARTICLE_ID, id);
        binding.getRoot().getContext().startActivity(i);
    });
    }
}
}

```

- ArticleFragment.java

```

- package com.kajianid.ustadz.ui.main.article

import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.inputmethod.EditorInfo
import android.widget.Toast
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.ustadz.databinding.FragmentArticleBinding
import com.kajianid.ustadz.ui.article.AddUpdateArticleActivity
import com.kajianid.ustadz.ui.article.ReadArticleActivity
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import org.json.JSONObject

class ArticleFragment : Fragment() {
    private lateinit var binding: FragmentArticleBinding
    private var adapter: ArticleAdapter = ArticleAdapter()
    private lateinit var articleViewModel: ArticleViewModel

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        adapter = ArticleAdapter()
        adapter.notifyDataSetChanged()
    }

    override fun onCreateView(inflater: LayoutInflater, container:
ViewGroup?,
                                savedInstanceState: Bundle?): View {
        // Inflate the layout for this fragment
        binding = FragmentArticleBinding.inflate(inflater)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState:
Bundle?) {
        super.onViewCreated(view, savedInstanceState)
    }
}

```

```

        binding.rvArticles.layoutManager =
            LinearLayoutManager(context)
        binding.rvArticles.adapter = adapter

        binding.pullToRefresh.visibility = View.GONE
        binding.pullToRefresh.setOnRefreshListener {
            GlobalScope.launch(Dispatchers.Main) {
                val deferredStatus = async(Dispatchers.IO) {

                    articleViewModel.setArticle(requireActivity().applicationContext,
                                                binding.etSearchArticle.text.toString())
                }
                val status = deferredStatus.await()
                if (status != null) {
                    binding.pullToRefresh.isRefreshing = false
                    val parse = JSONObject(status)
                    Toast.makeText(context,
                        parse.getString("message"), Toast.LENGTH_SHORT).show()
                } else {
                    binding.pullToRefresh.isRefreshing = false
                    adapter.notifyDataSetChanged()
                }
            }
        }

        binding.etSearchArticle.setOnEditorActionListener { _,
            actionId, _ ->
                if (actionId == EditorInfo.IME_ACTION_SEARCH) {
                    fetchArticles()
                }

                true
            }

        articleViewModel = ViewModelProvider(this,
            ViewModelProvider.NewInstanceFactory())
            .get(ArticleViewModel::class.java)
        articleViewModel.article.observe(viewLifecycleOwner, {
            if (it != null) {
                adapter.setArticles(it)
                binding.pullToRefresh.visibility = View.VISIBLE
                binding.progressMessage.visibility = View.GONE
            }
        })
        articleViewModel.setArticleAsync(requireActivity(), adapter,
            "")

        binding.extendedFab.setOnClickListener {
            val i = Intent(activity?.applicationContext,
                AddUpdateArticleActivity::class.java)
            startActivityForResult(i,
                AddUpdateArticleActivity.RESULT_SAVE)
        }

        private fun fetchArticles() {
            binding.pullToRefresh.visibility = View.GONE
            binding.progressMessage.visibility = View.VISIBLE
            GlobalScope.launch(Dispatchers.Main) {

```

```

        val deferredStatus = async(Dispatchers.IO) {
            articleViewModel.setArticle(requireActivity().applicationContext,
                binding.etSearchArticle.text.toString())
        }
        val status = deferredStatus.await()
        if (status != null) {
            binding.pullToRefresh.visibility = View.VISIBLE
            binding.progressMessage.visibility = View.GONE
            val parse = JSONObject(status)
            Toast.makeText(context, parse.getString("message"),
                Toast.LENGTH_SHORT).show()
        } else {
            binding.pullToRefresh.visibility = View.VISIBLE
            binding.progressMessage.visibility = View.GONE
            adapter.notifyDataSetChanged()
        }
    }
}

override fun onActivityResult(requestCode: Int, resultCode: Int,
    data: Intent?) {
    when (resultCode) {
        AddUpdateArticleActivity.RESULT_SAVE -> fetchArticles()
        ReadArticleActivity.RESULT_DELETE -> fetchArticles()
    }
    super.onActivityResult(requestCode, resultCode, data)
}
}

```

- ArticleViewModel.java

```

- package com.kajianid.ustadz.ui.main.article;

import android.content.Context;
import android.os.Looper;
import android.widget.Toast;

import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Article;
import com.kajianid.ustadz prefs.CredentialPreference;
import com.kajianid.ustadz.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;

```



```

import java.util.Locale;

import cz.msebera.android.httpclient.Header;

public class ArticleViewModel extends ViewModel {

    private final MutableLiveData<ArrayList<Article>> listArticle =
new MutableLiveData<>();

    public String setArticle(Context context, String searchQuery) {
        try {
            Looper.prepare();
        } catch (Exception ignored) {

        }
        CredentialPreference credentialPreference = new
CredentialPreference(context);
        final String[] ret = {null};
        String url =
context.getResources().getString(R.string.server) + "api/articles";
        ArrayList<Article> listItems = new ArrayList<>();
        SyncHttpClient client = new SyncHttpClient();

        RequestParams params = new RequestParams();
        params.put("read", "0");
        params.put("ustadz_mode", "1");
        params.put("ustadz_id",
credentialPreference.getCredential().getUsername());
        params.put("query", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");

                    for (int i = 0; i < list.length(); i++) {
                        JSONObject jsonObject =
list.getJSONObject(i);
                        Article article = new Article();

                        article.setId(jsonObject.getString("id"));

                        article.setTitle(jsonObject.getString("title"));

                        article.setContent(jsonObject.getString("content"));

                        article.setHasImg(StringHelper.convertToBoolean(jsonObject.getString(
"has_img")));

                        if (article.isHasImg()) {
                            String extension =
jsonObject.getString("extension");

                            article.setImgUrl(context.getResources().getString(R.string.server) +

```

```

"assets/articles/" +
                                article.getId() + "." +
extension);
                                }

                                String date =
jsonObject.getString("post_date");
                                DateFormat dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
                                assert dateDue != null;
                                String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue.parse(date));

                                article.setPost_date(dateDueFormatted);

                                listItems.add(article);
                                }
                                listArticle.postValue(listItems);
                                ret[0] = null;
                                } catch (JSONException | ParseException e) {
                                Toast.makeText(context, e.getMessage(),
Toast.LENGTH_LONG).show();
                                e.printStackTrace();
                                }
                                }

                                @Override
                                public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
                                error.printStackTrace();
                                Toast.makeText(
                                context,
                                "Failure!\n" +
                                error.getMessage() +
                                "\n" +
                                Arrays.toString(responseBody),
                                Toast.LENGTH_LONG
                                ).show();
                                ret[0] = Arrays.toString(responseBody);
                                }
                                });

                                return ret[0];
                                }

                                public boolean setArticleAsync(Context context, ArticleAdapter
adapter, String searchQuery) {
                                CredentialPreference credentialPreference = new
CredentialPreference(context);
                                final boolean[] ret = {false};
                                String url =
context.getResources().getString(R.string.server) + "api/articles";
                                ArrayList<Article> listItems = new ArrayList<>();
                                AsyncHttpClient client = new AsyncHttpClient();

                                RequestParams params = new RequestParams();
                                params.put("read", "0");
                                params.put("ustadz_mode", "1");

```

```

        params.put("ustadz_id",
credentialPreference.getCredential().getUsername());
        params.put("query", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");

                    for (int i = 0; i < list.length(); i++) {
                        JSONObject jsonObject =
list.getJSONObject(i);
                        Article article = new Article();

                        article.setId(jsonObject.getString("id"));

                        article.setTitle(jsonObject.getString("title"));

                        article.setContent(jsonObject.getString("content"));

                        article.setHasImg(StringHelper.convertToBoolean(jsonObject.getString(
"has_img")));

                        if (article.isHasImg()) {
                            String extension =
jsonObject.getString("extension");

                            article.setImgUrl(context.getResources().getString(R.string.server) +
"assets/articles/" +
                                article.getId() + "." +
extension);
                        }

                        String date =
jsonObject.getString("post_date");
                        DateFormat dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
                        assert dateDue != null;
                        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue.parse(date));

                        article.setPost_date(dateDueFormatted);

                        listItems.add(article);
                        adapter.notifyDataSetChanged();
                    }
                    listArticle.postValue(listItems);
                    ret[0] = true;
                } catch (JSONException | ParseException e) {
                    Toast.makeText(context, e.getMessage(),
Toast.LENGTH_LONG).show();
                    e.printStackTrace();
                }
            }
        });

```

```

    }

    @Override
    public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
        Toast.makeText(
            context,
            "Failure!\n" +
                error.getMessage() +
                "\n" +
                Arrays.toString(responseBody),
            Toast.LENGTH_LONG
        ).show();
        error.printStackTrace();
        ret[0] = false;
    }
});

return ret[0];
}

public LiveData<ArrayList<Article>> getArticle() {
    return listArticle;
}
}

```

```

- KajianAdapter.java
- package com.kajianid.ustadz.ui.main.kajian;

import android.content.Intent;
import android.view.LayoutInflater;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Kajian;
import com.kajianid.ustadz.databinding.ItemKajianBinding;
import com.kajianid.ustadz.ui.kajian.ShowKajianActivity;
import com.kajianid.ustadz.utils.StringHelper;

import java.util.ArrayList;

public class KajianAdapter extends
RecyclerView.Adapter<KajianAdapter.KajianViewHolder> {

    private ArrayList<Kajian> data = new ArrayList<>();

    @NonNull
    @Override
    public KajianViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        ItemKajianBinding binding =

```

```

ItemKajianBinding.inflate(LayoutInflater.from(parent.getContext()));
        return new KajianViewHolder(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull KajianViewHolder holder,
int position) {
        holder.bind(data.get(position));
    }

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

    public void setData(ArrayList<Kajian> kajian) {
        this.data.clear();
        this.data.addAll(kajian);
        notifyDataSetChanged();
    }

    public static class KajianViewHolder extends
RecyclerView.ViewHolder {

        private ItemKajianBinding binding;

        public KajianViewHolder(@NonNull ItemKajianBinding binding) {
            super(binding.getRoot());
            this.binding = binding;
        }

        public void bind(Kajian kajian) {
            String mosqueOrAddress = (kajian.getPlace().equals("Di
Tempat")) ? kajian.getAddress() : kajian.getMosqueName();
            binding.tvTitle.setText(kajian.getTitle());
            binding.tvKet.setText(kajian.getPlace());
            binding.tvDue.setText(kajian.getDate());
            binding.tvMosqueAddress.setText(mosqueOrAddress);

            if (!StringHelper.isNullOrEmpty(kajian.getImgResource()))
            {
                Glide.with(binding.getRoot().getContext())
                    .load(kajian.getImgResource())
                    .into(binding.imgKajian);
            } else {
                Glide.with(binding.getRoot().getContext())
                    .load(R.drawable.intro3)
                    .into(binding.imgKajian);
            }

            binding.getRoot().setOnClickListener(it -> {
                Intent i = new Intent(binding.getRoot().getContext(),
ShowKajianActivity.class);
                i.putExtra(ShowKajianActivity.EXTRA_KAJIAN_ID,
kajian.getId());
                binding.getRoot().getContext().startActivity(i);
            });
        }
    }

```

```
}
}
```

```
- KajianFragment.kt
- package com.kajianid.ustadz.ui.main.kajian

import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.inputmethod.EditorInfo
import android.widget.Toast
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.ustadz.databinding.FragmentKajianBinding
import com.kajianid.ustadz.ui.kajian.AddUpdateKajianActivity
import com.kajianid.ustadz.ui.kajian.ShowKajianActivity
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import org.json.JSONObject

class KajianFragment : Fragment() {
    private var adapter = KajianAdapter()
    private lateinit var kajianViewModel: KajianViewModel
    private lateinit var binding: FragmentKajianBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        adapter = KajianAdapter()
        adapter.notifyDataSetChanged()
    }

    override fun onCreateView(inflater: LayoutInflater, container:
    ViewGroup?,
                                savedInstanceState: Bundle?): View {
        // Inflate the layout for this fragment
        binding = FragmentKajianBinding.inflate(inflater)
        return binding.root
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)

        // recview adapter
        binding.rvInfo.layoutManager = LinearLayoutManager(context)
        binding.rvInfo.adapter = adapter

        binding.pullToRefresh.setOnRefreshListener {
            GlobalScope.launch(Dispatchers.Main) {
                val deferredStatus = async(Dispatchers.IO) {
                    kajianViewModel.setKajian(requireActivity().applicationContext,
                                                binding.etSearchKajian.text.toString())
                }
            }
        }
    }
}
```

```

        val status = deferredStatus.await()
        if (status != null) {
            binding.pullToRefresh.isRefreshing = false
            val parse = JSONObject(status)
            Toast.makeText(context,
parse.getString("message"), Toast.LENGTH_SHORT).show()
        } else {
            binding.pullToRefresh.isRefreshing = false
            adapter.notifyDataSetChanged()
        }
    }
}

binding.etSearchKajian.setOnEditorActionListener { _,
actionId, _ ->
    if (actionId == EditorInfo.IME_ACTION_SEARCH) {
        binding.progressMessage.visibility = View.VISIBLE
        GlobalScope.launch(Dispatchers.Main) {
            val deferredStatus = async(Dispatchers.IO) {

kajianViewModel.setKajian(requireActivity().applicationContext,

binding.etSearchKajian.text.toString())
            }
            val status = deferredStatus.await()
            if (status != null) {
                binding.progressMessage.visibility =

View.GONE

                val parse = JSONObject(status)
                Toast.makeText(context,
parse.getString("message"), Toast.LENGTH_SHORT).show()
            } else {
                binding.progressMessage.visibility =

View.GONE

                adapter.notifyDataSetChanged()
            }
        }
    }
}

true
}

kajianViewModel = ViewModelProvider(this,
ViewModelProvider.NewInstanceFactory())
    .get(KajianViewModel::class.java)
kajianViewModel.kajian.observe(viewLifecycleOwner, {
    if (it != null) {
        adapter.setData(it)
        binding.progressMessage.visibility = View.GONE
    }
})

kajianViewModel.setKajianAsync(requireActivity().applicationContext,
adapter, "")

binding.fabAddKajian.setOnClickListener {
    val i = Intent(activity?.applicationContext,
AddUpdateKajianActivity::class.java)
    startActivityForResult(i,

```

```

        AddUpdateKajianActivity.RESULT_SAVE)
    }
}

private fun fetchKajian() {
    binding.progressMessage.visibility = View.VISIBLE
    GlobalScope.launch(Dispatchers.Main) {
        val deferredStatus = async(Dispatchers.IO) {

kajianViewModel.setKajian(requireActivity().applicationContext,
                            binding.etSearchKajian.text.toString())
        }
        val status = deferredStatus.await()
        if (status != null) {
            binding.progressMessage.visibility = View.GONE
            val parse = JSONObject(status)
            Toast.makeText(context, parse.getString("message"),
Toast.LENGTH_SHORT).show()
        } else {
            binding.progressMessage.visibility = View.GONE
            adapter.notifyDataSetChanged()
        }
    }
}

override fun onActivityResult(requestCode: Int, resultCode: Int,
data: Intent?) {
    when (resultCode) {
        AddUpdateKajianActivity.RESULT_SAVE -> fetchKajian()
        ShowKajianActivity.RESULT_DELETE -> fetchKajian()
    }
    super.onActivityResult(requestCode, resultCode, data)
}
}

```

- KajianViewModel.java

```

- package com.kajianid.ustadz.ui.main.kajian;

import android.content.Context;
import android.os.Looper;
import android.widget.Toast;

import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Kajian;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

```



```

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Locale;

import cz.msebera.android.httpclient.Header;

public class KajianViewModel extends ViewModel {

    private final MutableLiveData<ArrayList<Kajian>> listKajian = new
    MutableLiveData<>();

    public LiveData<ArrayList<Kajian>> getKajian() {
        return listKajian;
    }

    public String setKajian(Context context, String searchQuery) {
        try {
            Looper.prepare();
        } catch (Exception ignored) {
        }

        CredentialPreference credential = new
        CredentialPreference(context);
        String[] ret = {null};
        String url = context.getString(R.string.server) +
        "api/kajian";
        ArrayList<Kajian> listItems = new ArrayList<>();
        SyncHttpClient client = new SyncHttpClient();

        // Request Parameters
        RequestParams params = new RequestParams();
        params.put("read", "0");
        params.put("ustadz_mode", "1");
        params.put("ustadz_id",
        credential.getCredential().getUsername()); // replace admin with
        Ustadz ID
        params.put("q", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
            byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
                    resultObject.getJSONArray("data");

                    for (int i = 0; i < list.length(); i++) {
                        JSONObject jsonObject =
                        list.getJSONObject(i);
                        Kajian kajian = new Kajian();

                        kajian.setId(jsonObject.getString("id"));

                        kajian.setTitle(jsonObject.getString("kajian_title"));

```

```

kajian.setPlace(jsonObject.getString("place"));

kajian.setAddress(jsonObject.getString("address"));

kajian.setMosqueId(jsonObject.getString("mosque_id"));

kajian.setMosqueName(jsonObject.getString("mosque_name"));

        String date =
jsonObject.getString("date_due");
        DateFormat dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue.parse(date));

        kajian.setDate(dateDueFormatted);

kajian.setImgResource(jsonObject.getString("img_resource"));

        listItems.add(kajian);
    }

    listKajian.postValue(listItems);
    ret[0] = null;
} catch (JSONException | ParseException e) {
    e.printStackTrace();
    Toast.makeText(context, "Failure!\n" +
e.getMessage() + "\n{onSuccess(), client.get()}",
Toast.LENGTH_LONG).show();
    ret[0] = e.getMessage();
}
}

@Override
public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
    error.printStackTrace();
    Toast.makeText(context, "Failure!\n" +
error.getMessage() + "\n{onFailure(), client.get()}",
Toast.LENGTH_LONG).show();
    String response = new String(responseBody);
    ret[0] = response;
}

});

return ret[0];
}

public boolean setKajianAsync(Context context, KajianAdapter
adapter, String searchQuery) {
    CredentialPreference credential = new
CredentialPreference(context);
    boolean[] ret = {false};
    String url = context.getString(R.string.server) +
"api/kajian";
    ArrayList<Kajian> listItems = new ArrayList<>();
    AsyncHttpClient client = new AsyncHttpClient();

```

```

        // Request Parameters
        RequestParams params = new RequestParams();
        params.put("read", "0");
        params.put("ustadz_mode", "1");
        params.put("ustadz_id",
credential.getCredential().getUsername()); // replace admin with
Ustadz ID
        params.put("q", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
resultObject.getJSONArray("data");

                    for (int i = 0; i < list.length(); i++) {
                        JSONObject jsonObject =
list.getJSONObject(i);
                        Kajian kajian = new Kajian();

                        kajian.setId(jsonObject.getString("id"));

                        kajian.setTitle(jsonObject.getString("kajian_title"));
                        kajian.setPlace(jsonObject.getString("place"));
                        kajian.setAddress(jsonObject.getString("address"));
                        kajian.setMosqueId(jsonObject.getString("mosque_id"));
                        kajian.setMosqueName(jsonObject.getString("mosque_name"));

                        String date =
jsonObject.getString("date_due");
                        DateFormat dateDue = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
                        String dateDueFormatted = new
SimpleDateFormat("dd MMM yyyy HH:mm",
Locale.getDefault()).format(dateDue.parse(date));

                        kajian.setDate(dateDueFormatted);

                        kajian.setImgResource(jsonObject.getString("img_resource"));

                        listItems.add(kajian);
                        adapter.notifyDataSetChanged();
                    }

                    listKajian.postValue(listItems);

                    ret[0] = true;
                } catch (JSONException | ParseException e) {
                    e.printStackTrace();
                    Toast.makeText(context, "Failure!\n" +

```

```

e.getMessage() + "\n{onSuccess(), client.get()}",
Toast.LENGTH_LONG).show();
        ret[0] = false;
    }
}

@Override
public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
    error.printStackTrace();
    //String response = Arrays.toString(responseBody);
    Toast.makeText(context, "Failure!\n" +
error.getMessage() + "\n", Toast.LENGTH_LONG).show();
    ret[0] = false;
}

});

return ret[0];
}
}

```

```

- MainActivity.java
- package com.kajianid.ustadz.ui.main;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.databinding.ActivityMainBinding;
import com.kajianid.ustadz.ui.about.AboutActivity;
import com.kajianid.ustadz.ui.settings.SettingsActivity;

import java.util.Objects;

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        MainPagerAdapter mainPagerAdapter = new
MainPagerAdapter(this, getSupportFragmentManager());
        binding.viewPager.setAdapter(mainPagerAdapter);
        binding.tabs.setupWithViewPager(binding.viewPager);
    }
}

```

```

        setSupportActionBar(binding.toolbar);

Objects.requireNonNull(getSupportActionBar()).setElevation(0f);

Objects.requireNonNull(getSupportActionBar()).setTitle(getString(R.string.app_name));
    }

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

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == R.id.menuSettings) {
            Intent i = new Intent(this, SettingsActivity.class);
            startActivity(i);
        } else if (item.getItemId() == R.id.menuAbout) {
            Intent i = new Intent(this, AboutActivity.class);
            startActivity(i);
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        binding = null;
    }
}

```

- MainPagerAdapter.java

```

- package com.kajianid.ustadz.ui.main;

import android.content.Context;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.StringRes;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.ui.main.article.ArticleFragment;
import com.kajianid.ustadz.ui.main.kajian.KajianFragment;
import com.kajianid.ustadz.ui.mosque.MosqueListFragment;

public class MainPagerAdapter extends FragmentPagerAdapter {

    private final Context context;

    @StringRes
    private final int[] tabTitles = {

```

```

        R.string.tab_article,
        R.string.tab_kajian,
        R.string.tab_mosque
    };

    public MainPagerAdapter(Context context, @NonNull FragmentManager
fm) {
        super(fm, BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
        this.context = context;
    }

    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        return context.getResources().getString(tabTitles[position]);
    }

    @NonNull
    @Override
    public Fragment getItem(int position) {
        Fragment fragment = null;
        switch (position) {
            case 0:
                fragment = new ArticleFragment();
                break;
            case 1:
                fragment = new KajianFragment();
                break;
            case 2:
                fragment = new MosqueListFragment();
                break;
        }
        return fragment;
    }

    @Override
    public int getCount() {
        return 3;
    }
}

```

- MosqueChooserActivity.java

```

- package com.kajianid.ustadz.ui.mosque;

import android.app.SearchManager;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.View;
import android.view.Window;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SearchView;
import androidx.lifecycle.ViewModelProvider;
import androidx.recyclerview.widget.LinearLayoutManager;

```

```

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.databinding.ActivityMosqueChooserBinding;

import java.util.Objects;

public class MosqueChooserActivity extends AppCompatActivity {

    public static final String EXTRA_MOSQUE_RESULT =
"extra_mosque_result";
    public static final String EXTRA_IS_USTADZ_ONLY =
"extra_is_ustadz_only";
    public static final int REQUEST_MOSQUE = 100;
    private MosqueChooserAdapter mosqueChooserAdapter;
    private MosqueChooserViewModel mosqueChooserViewModel;
    private ActivityMosqueChooserBinding binding;
    private boolean isUstadzOnly = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        supportRequestWindowFeature(Window.FEATURE_NO_TITLE);
        binding =
ActivityMosqueChooserBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setFinishOnTouchOutside(false);

        setSupportActionBar(binding.toolbar);

Objects.requireNonNull(getSupportActionBar()).setTitle(R.string.choos
e_mosque);

        mosqueChooserAdapter = new MosqueChooserAdapter();
        mosqueChooserAdapter.notifyDataSetChanged();

        binding.rvMosqueList.setLayoutManager(new
LinearLayoutManager(this));
        binding.rvMosqueList.setAdapter(mosqueChooserAdapter);

        binding.progressMessage.setVisibility(View.VISIBLE);

        binding.btnBack.setOnClickListener(ignored -> {
            finish();
        });

        binding.tvNoMosque.setOnClickListener(it -> {
            String text = "Permisi, saya ingin mendaftarkan suatu
tempat untuk dijadikan Masjid di aplikasi Kajian.ID.\n" +
                "Berikut informasi data diri saya:\n\n" +
                "Nama: \n" +
                "Email: \n" +
                "Nama Lokasi: \n" +
                "Alamat Lokasi: \n" +
                "Keperluan: \n\n" +
                "Dan untuk selanjutnya, lokasi tersebut akan saya
bagikan (share location) melalui WhatsApp ini.\n\n" +
                "Terimakasih.";

            Uri whatsappUri = Uri.parse("https://wa.me/" +
getString(R.string.whatsapp_target) + "?text=" + text);

```

```

        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(whatsappUri);
        startActivity(i);
    });

    Bundle bundle = getIntent().getExtras();
    if (bundle != null) {
        isUstadzOnly = bundle.getBoolean(EXTRA_IS_USTADZ_ONLY,
true);

        mosqueChooserViewModel = new ViewModelProvider(this, new
ViewModelProvider.NewInstanceFactory())
            .get(MosqueChooserViewModel.class);
        mosqueChooserViewModel.getMosques().observe(this, mosques
-> {
            if (mosques != null) {
                binding.progressMessage.setVisibility(View.GONE);
                mosqueChooserAdapter.setMosques(mosques);
            }
        });
        mosqueChooserViewModel.setMosqueAsync(
            this,
            mosqueChooserAdapter,
            "",
            isUstadzOnly
        );
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_mosque_chooser, menu);

    SearchManager searchManager = (SearchManager)
getSystemService(Context.SEARCH_SERVICE);
    SearchView searchView = (SearchView)
menu.findItem(R.id.search).getActionView();

    searchView.setSearchableInfo(searchManager.getSearchableInfo(getCompo
nentName()));
    searchView.setQueryHint(getString(R.string.search_mosque));
    searchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            binding.progressMessage.setVisibility(View.VISIBLE);
            mosqueChooserViewModel.setMosqueAsync(
                MosqueChooserActivity.this,
                mosqueChooserAdapter,
                query,
                isUstadzOnly);
            searchView.clearFocus();
            return true;
        }
    })

    @Override

```



```

        public boolean onQueryTextChanged(String newText) {
            /* no-op */
            return true;
        }
    });

    return true;
}

@Override
protected void onDestroy() {
    super.onDestroy();
    binding = null;
}
}

```

- MosqueChooserAdapter.java

```

- package com.kajianid.ustadz.ui.mosque;

import android.app.Activity;
import android.content.Intent;
import android.location.Address;
import android.location.Geocoder;
import android.view.LayoutInflater;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.gms.maps.model.LatLng;
import com.kajianid.ustadz.data.Mosque;
import com.kajianid.ustadz.databinding.ItemMosqueBinding;
import com.kajianid.ustadz.utils.StringHelper;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

public class MosqueChooserAdapter extends
RecyclerView.Adapter<MosqueChooserAdapter.MosqueChooserViewHolder> {

    private final ArrayList<Mosque> mosques = new ArrayList<>();

    public void setMosques(ArrayList<Mosque> mosques) {
        this.mosques.clear();
        this.mosques.addAll(mosques);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public MosqueChooserAdapter.MosqueChooserViewHolder
    onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        ItemMosqueBinding binding =
        ItemMosqueBinding.inflate(LayoutInflater.from(parent.getContext()));
        return new MosqueChooserViewHolder(binding);
    }
}

```

```

        @Override
        public void onBindViewHolder(@NonNull
MosqueChooserAdapter.MosqueChooserViewHolder holder, int position) {
            try {
                holder.bind(mosques.get(position));
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

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

        public static class MosqueChooserViewHolder extends
RecyclerView.ViewHolder {

            private final ItemMosqueBinding binding;

            public MosqueChooserViewHolder(@NonNull ItemMosqueBinding
itemView) {
                super(itemView.getRoot());
                this.binding = itemView;
            }

            public void bind(Mosque mosque) throws IOException {
                binding.tvMosqueName.setText(mosque.getMosqueName());
                if (StringHelper.isNullOrEmpty(mosque.getAddress())) {
                    String latLng = mosque.getLatLng();
                    String[] latLngArray = latLng.split(",");
                    double latitude = Double.parseDouble(latLngArray[0]);
                    double longitude =
Double.parseDouble(latLngArray[1]);
                    LatLng location = new LatLng(latitude, longitude);
                    Geocoder geocoder = new
Geocoder(binding.getRoot().getContext(), Locale.getDefault());
                    List<Address> addresses = geocoder.getFromLocation(
                        location.latitude,
                        location.longitude,
                        1
                    );

                    String address = addresses.get(0).getAddressLine(0);
                    String city = addresses.get(0).getLocality();
                    String state = addresses.get(0).getAdminArea();
                    String country = addresses.get(0).getCountryName();
                    String postalCode = addresses.get(0).getPostalCode();

                    String fullAddress = address + ", " + city + ", " +
state + ", " + country + " " + postalCode;
                    binding.tvMosqueAddress.setText(fullAddress);
                } else
binding.tvMosqueAddress.setText(mosque.getAddress());

                binding.getRoot().setOnClickListener(it -> {
                    Intent i = new Intent();
                    i.putExtra(MosqueChooserActivity.EXTRA_MOSQUE_RESULT,
mosque);
                });
            }
        }
    }
}

```

```

        ((Activity)
binding.getRoot().getContext()).setResult(MosqueChooserActivity.REQUE
ST_MOSQUE, i);
        ((Activity) binding.getRoot().getContext()).finish();
    });
    }
}
}

```

- MosqueChooserViewModel.java

```

- package com.kajianid.ustadz.ui.mosque;

import android.content.Context;
import android.os.Looper;
import android.widget.Toast;

import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Mosque;
import com.kajianid.ustadz.prefs.CredentialPreference;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import com.loopj.android.http.RequestParams;
import com.loopj.android.http.SyncHttpClient;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

import cz.msebera.android.httpclient.Header;

public class MosqueChooserViewModel extends ViewModel {

    private MutableLiveData<ArrayList<Mosque>> listMosques = new
MutableLiveData<>();

    public LiveData<ArrayList<Mosque>> getMosques() {
        return listMosques;
    }

    public String setMosques(Context context, String searchQuery,
boolean isUstadzOnly) {
        try {
            Looper.prepare();
        } catch (Exception ignored) {
        }
        final String[] ret = {null};
        CredentialPreference credential = new
CredentialPreference(context);
        String endpoint = (isUstadzOnly) ? "api/mosques/ustadz" :
"api/mosques/user";
        String url =
context.getResources().getString(R.string.server) + endpoint;
        ArrayList<Mosque> listItems = new ArrayList<>();

```

```

        SyncHttpClient client = new SyncHttpClient();

        RequestParams params = new RequestParams();
        if (isUstadzOnly) params.put("ustadz_id",
            credential.getCredential().getUsername());
        params.put("q", searchQuery);

        client.get(url, params, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers,
                byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new JSONObject(result);
                    JSONArray list =
                        resultObject.getJSONArray("data");

                    for (int i = 0; i < list.length(); i++) {
                        JSONObject mosqueJson =
                            list.getJSONObject(i);
                        Mosque mosque = new Mosque();

                        int mosqueId = (isUstadzOnly) ?
                            mosqueJson.getInt("mosque_id") : mosqueJson.getInt("id");
                        mosque.setId(mosqueId);

                        mosque.setMosqueName(mosqueJson.getString("name"));
                        mosque.setAddress(mosqueJson.getString("address"));
                        mosque.setLatLng(mosqueJson.getString("lat_lng"));

                        listItems.add(mosque);
                    }

                    listMosques.postValue(listItems);
                    ret[0] = null;
                } catch (JSONException e) {
                    e.printStackTrace();
                    String expected = new String(responseBody);
                    ret[0] = expected;
                }
            }

            @Override
            public void onFailure(int statusCode, Header[] headers,
                byte[] responseBody, Throwable error) {
                error.printStackTrace();
                String expected = new String(responseBody);
                ret[0] = expected;
            }
        });

        return ret[0];
    }

    public boolean setMosqueAsync(Context context,
        MosqueListAdapter adapter,
        String searchQuery,

```

```

        Boolean isUstadzOnly) {
            final boolean[] ret = {false};
            CredentialPreference credential = new
            CredentialPreference(context);
            String endpoint = (isUstadzOnly) ? "api/mosques/ustadz" :
            "api/mosques/user";
            String url =
            context.getResources().getString(R.string.server) + endpoint;
            ArrayList<Mosque> listItems = new ArrayList<>();
            AsyncHttpClient client = new AsyncHttpClient();

            RequestParams params = new RequestParams();
            if (isUstadzOnly) params.put("ustadz_id",
            credential.getCredential().getUsername());
            params.put("q", searchQuery);

            client.get(url, params, new AsyncHttpResponseHandler() {
                @Override
                public void onSuccess(int statusCode, Header[] headers,
                byte[] responseBody) {
                    try {
                        String result = new String(responseBody);
                        JSONObject resultObject = new JSONObject(result);
                        JSONArray list =
                        resultObject.getJSONArray("data");

                        for (int i = 0; i < list.length(); i++) {
                            JSONObject mosqueJson =
                            list.getJSONObject(i);
                            Mosque mosque = new Mosque();

                            int mosqueId = (isUstadzOnly) ?
                            mosqueJson.getInt("mosque_id") : mosqueJson.getInt("id");
                            mosque.setId(mosqueId);

                            mosque.setMosqueName(mosqueJson.getString("name"));
                            mosque.setAddress(mosqueJson.getString("address"));
                            mosque.setLatLng(mosqueJson.getString("lat_lng"));

                            listItems.add(mosque);
                            adapter.notifyDataSetChanged();
                        }

                        listMosques.postValue(listItems);
                        ret[0] = true;
                    } catch (JSONException e) {
                        e.printStackTrace();
                        Toast.makeText(context, "Failure!\n" +
                        e.getMessage() + "\n${onSuccess()}\n${url}",
                        Toast.LENGTH_SHORT).show();
                        ret[0] = false;
                    }
                }

                @Override
                public void onFailure(int statusCode, Header[] headers,
                byte[] responseBody, Throwable error) {

```

```

        error.printStackTrace();
        String expected = new String(responseBody);
        Toast.makeText(context, "Failure!\n" +
error.getMessage() + "\n" + expected, Toast.LENGTH_SHORT).show();
        ret[0] = false;
    }
});

return ret[0];
}

public boolean setMosqueAsync(Context context,
                             MosqueChooserAdapter adapter,
                             String searchQuery,
                             Boolean isUstadzOnly) {
    final boolean[] ret = {false};
    CredentialPreference credential = new
CredentialPreference(context);
    String endpoint = (isUstadzOnly) ? "api/mosques/ustadz" :
"api/mosques/user";
    String url =
context.getResources().getString(R.string.server) + endpoint;
    ArrayList<Mosque> listItems = new ArrayList<>();
    AsyncHttpClient client = new AsyncHttpClient();

    RequestParams params = new RequestParams();
    if (isUstadzOnly) params.put("ustadz_id",
credential.getCredential().getUsername());
    params.put("q", searchQuery);

    client.get(url, params, new AsyncHttpResponseHandler() {
        @Override
        public void onSuccess(int statusCode, Header[] headers,
byte[] responseBody) {
            try {
                String result = new String(responseBody);
                JSONObject resultObject = new JSONObject(result);
                JSONArray list =
resultObject.getJSONArray("data");

                for (int i = 0; i < list.length(); i++) {
                    JSONObject mosqueJson =
list.getJSONObject(i);
                    Mosque mosque = new Mosque();

                    int mosqueId = (isUstadzOnly) ?
mosqueJson.getInt("mosque_id") : mosqueJson.getInt("id");
                    mosque.setId(mosqueId);

                    mosque.setMosqueName(mosqueJson.getString("name"));

                    mosque.setAddress(mosqueJson.getString("address"));

                    mosque.setLatLng(mosqueJson.getString("lat_lng"));

                    listItems.add(mosque);
                    adapter.notifyDataSetChanged();
                }
            }
        }
    });
}

```

```

        listMosques.postValue(listItems);
        ret[0] = true;
    } catch (JSONException e) {
        e.printStackTrace();
        Toast.makeText(context, "Failure!\n" +
e.getMessage() + "\n${onSuccess()}\n${url}",
Toast.LENGTH_SHORT).show();
        ret[0] = false;
    }
}

@Override
public void onFailure(int statusCode, Header[] headers,
byte[] responseBody, Throwable error) {
    error.printStackTrace();
    String expected = new String(responseBody);
    Toast.makeText(context, "Failure!\n" +
error.getMessage() + "\n" + expected, Toast.LENGTH_SHORT).show();
    ret[0] = false;
}
});

return ret[0];
}
}

```

- MosqueListAdapter.java

```

- package com.kajianid.ustadz.ui.mosque;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.content.Intent;
import android.location.Address;
import android.location.Geocoder;
import android.net.Uri;
import android.view.LayoutInflater;
import android.view.ViewGroup;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.gms.maps.model.LatLng;
import com.kajianid.ustadz.R;
import com.kajianid.ustadz.data.Mosque;
import com.kajianid.ustadz.databinding.ItemMosqueListBinding;
import com.kajianid.ustadz.preferences.CredentialPreference;
import com.kajianid.ustadz.utils.StringHelper;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import cz.msebera.android.httpclient.Header;

```

```

public class MosqueListAdapter extends
RecyclerView.Adapter<MosqueListAdapter.MosqueChooserViewHolder> {

    private final ArrayList<Mosque> mosques = new ArrayList<>();

    public void setMosques(ArrayList<Mosque> mosques) {
        this.mosques.clear();
        this.mosques.addAll(mosques);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public MosqueListAdapter.MosqueChooserViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        ItemMosqueListBinding binding =
ItemMosqueListBinding.inflate(LayoutInflater.from(parent.getContext()
));
        return new MosqueChooserViewHolder(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull
MosqueListAdapter.MosqueChooserViewHolder holder, int position) {
        try {
            holder.bind(mosques.get(position), position);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

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

    public class MosqueChooserViewHolder extends
RecyclerView.ViewHolder {

        private final ItemMosqueListBinding binding;

        public MosqueChooserViewHolder(@NonNull ItemMosqueListBinding
itemView) {
            super(itemView.getRoot());
            this.binding = itemView;
        }

        @SuppressWarnings("QueryPermissionsNeeded")
        public void bind(Mosque mosque, int position) throws
IOException {
            binding.tvMosqueName.setText(mosque.getMosqueName());
            if (StringHelper.isNullOrEmpty(mosque.getAddress())) {
                String latLng = mosque.getLatLng();
                String[] latLngArray = latLng.split(",");
                double latitude = Double.parseDouble(latLngArray[0]);
                double longitude =
Double.parseDouble(latLngArray[1]);
                LatLng location = new LatLng(latitude, longitude);
                Geocoder geocoder = new

```



```

Geocoder(binding.getRoot().getContext(), Locale.getDefault());
    List<Address> addresses = geocoder.getFromLocation(
        location.latitude,
        location.longitude,
        1
    );

    String address = addresses.get(0).getAddressLine(0);
    String city = addresses.get(0).getLocality();
    String state = addresses.get(0).getAdminArea();
    String country = addresses.get(0).getCountryName();
    String postalCode = addresses.get(0).getPostalCode();

    String fullAddress = address + ", " + city + ", " +
state + ", " + country + " " + postalCode;
    binding.tvMosqueAddress.setText(fullAddress);
} else
binding.tvMosqueAddress.setText(mosque.getAddress());

binding.getRoot().setOnClickListener(it -> new
AlertDialog.Builder(binding.getRoot().getContext())
    .setMessage(R.string.choose_action_mosque)
    .setPositiveButton(R.string.view_on_maps, (a, b)
-> {
        Intent mapIntent = new Intent(
            Intent.ACTION_VIEW,
            Uri.parse(mosque.getLatLng() + "?q="
+ mosque.getLatLng() + "(" + mosque.getMosqueName() + ")")
        );

        mapIntent.setPackage("com.google.android.apps.maps");
        if
        (mapIntent.resolveActivity(binding.getRoot().getContext().getPackageManager()) != null)

        binding.getRoot().getContext().startActivity(mapIntent);
        else

        Toast.makeText(binding.getRoot().getContext(), "Aplikasi
com.google.android.apps.maps tidak ada", Toast.LENGTH_SHORT).show();
        })
        .setNegativeButton(R.string.delete, (a, b) -> new
AlertDialog.Builder(binding.getRoot().getContext())

        .setMessage(binding.getRoot().getContext().getString(R.string.delete_
mosque_confirm) + mosque.getMosqueName() + "?")
        .setPositiveButton(R.string.yes, (c, d) -
> {
            CredentialPreference preference = new
CredentialPreference(binding.getRoot().getContext());
            String ustadzId =
preference.getCredential().getUsername();
            String api =
binding.getRoot().getContext().getString(R.string.server) +
"api/mosques/" + mosque.getId() + "/" + ustadzId;
            AsyncHttpClient client = new
AsyncHttpClient();
            client.delete(api, new
AsyncHttpResponseHandler() {

```

```

                                @Override
                                public void onSuccess(int
statusCode, Header[] headers, byte[] responseBody) {
                                    mosques.remove(position);
                                    notifyDataSetChanged();
                                    Toast.makeText(

binding.getRoot().getContext(),

binding.getRoot().getContext().getString(R.string.delete_mosque_succe
ss) + mosque.getMosqueName() + "!",
                                    Toast.LENGTH_SHORT
                                    ).show();
                                }

                                @Override
                                public void onFailure(int
statusCode, Header[] headers, byte[] responseBody, Throwable error) {
                                    Toast.makeText(

binding.getRoot().getContext(),

binding.getRoot().getContext().getString(R.string.delete_mosque_faile
d) + mosque.getMosqueName() + "!",
                                    Toast.LENGTH_SHORT
                                    ).show();
                                }
                            });
                        })
                        .setNegativeButton(R.string.no, null)
                        .create().show()
                        .setNeutralButton(R.string.cancel, null)
                        .create().show();
                    }
                }
            }
        }
    }
}

```

- MosqueListFragment.kt

- package com.kajianid.ustadz.ui.mosque

```

import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.inputmethod.EditorInfo
import android.widget.Toast
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import com.kajianid.ustadz.R
import com.kajianid.ustadz.data.Mosque
import com.kajianid.ustadz.databinding.FragmentMosqueListBinding
import com.kajianid.ustadz.preferences.CredentialPreference
import com.loopj.android.http.AsyncHttpClient
import com.loopj.android.http.AsyncHttpResponseHandler
import com.loopj.android.http.RequestParams
import cz.msebera.android.httpclient.Header
import kotlinx.coroutines.Dispatchers

```

```

import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import org.json.JSONObject

class MosqueListFragment : Fragment() {

    private var mosqueListAdapter: MosqueListAdapter =
        MosqueListAdapter()
    private lateinit var mosqueListViewModel: MosqueChooserViewModel
    private lateinit var binding: FragmentMosqueListBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        mosqueListAdapter = MosqueListAdapter()
        mosqueListAdapter.notifyDataSetChanged()
    }

    override fun onCreateView(inflater: LayoutInflater, container:
        ViewGroup?,
                               savedInstanceState: Bundle?): View {
        // Inflate the layout for this fragment
        binding = FragmentMosqueListBinding.inflate(inflater)
        return binding.root
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)

        binding.rvMosque.layoutManager = LinearLayoutManager(context)
        binding.rvMosque.adapter = mosqueListAdapter

        mosqueListViewModel = ViewModelProvider(this,
            ViewModelProvider.NewInstanceFactory())
            .get(MosqueChooserViewModel::class.java)
        mosqueListViewModel.mosques.observe(viewLifecycleOwner, {
            if (it != null) {
                mosqueListAdapter.setMosques(it)
                binding.progressMessage.visibility = View.GONE
            }
        })
        context?.let { mosqueListViewModel.setMosqueAsync(it,
            mosqueListAdapter, "", true) }

        binding.pullToRefresh.setOnRefreshListener {
            GlobalScope.launch(Dispatchers.Main) {
                val deferredStatus = async(Dispatchers.IO) {
                    mosqueListViewModel.setMosques(requireActivity().applicationContext,
                        binding.etSearchMosque.text.toString(),
                        true)
                }
                val status = deferredStatus.await()
                if (status != null) {
                    binding.pullToRefresh.isRefreshing = false
                    val parse = JSONObject(status)
                    Toast.makeText(context,
                        parse.getString("message"), Toast.LENGTH_SHORT).show()
                }
            }
        }
    }
}

```

```

        } else {
            binding.pullToRefresh.isRefreshing = false
            mosqueListAdapter.notifyDataSetChanged()
        }
    }
}

binding.etSearchMosque.setOnEditorActionListener { _,
actionId, _ ->
    if (actionId == EditorInfo.IME_ACTION_SEARCH) {
        binding.progressMessage.visibility = View.VISIBLE
        GlobalScope.launch(Dispatchers.Main) {
            val deferredStatus = async(Dispatchers.IO) {
                mosqueListViewModel.setMosques(requireActivity().applicationContext,
                binding.etSearchMosque.text.toString(), true)
            }
            val status = deferredStatus.await()
            if (status != null) {
                binding.pullToRefresh.isRefreshing = false
                val parse = JSONObject(status)
                Toast.makeText(context,
                parse.getString("message"), Toast.LENGTH_SHORT).show()
            } else {
                binding.pullToRefresh.isRefreshing = false
                mosqueListAdapter.notifyDataSetChanged()
            }
        }
    }
}
true
}

binding.fabAddMosque.setOnClickListener {
    val i = Intent(context,
MosqueChooserActivity::class.java)
    i.putExtra(MosqueChooserActivity.EXTRA_IS_USTADZ_ONLY,
false)
    startActivityForResult(i,
MosqueChooserActivity.REQUEST_MOSQUE)
}

override fun onActivityResult(requestCode: Int, resultCode: Int,
data: Intent?) {
    when (resultCode) {
        MosqueChooserActivity.REQUEST_MOSQUE -> {
            if (data != null) {
                val credential =
CredentialPreference(requireActivity().applicationContext!!)

                val mosque =
data.getParcelableExtra<Mosque>(MosqueChooserActivity.EXTRA_MOSQUE_RE
SULT)!!

                val progressMsg =
resources.getString(R.string.progress_add_mosque) + mosque.mosqueName
                binding.progressMessage.visibility = View.VISIBLE
                binding.tvProgressMessage.text = progressMsg
            }
        }
    }
}

```

```

        val mosqueId = mosque.id
        val ustadzId = credential.credential.username

        val api = resources.getString(R.string.server) +
"api/mosques"

        val client = AsyncHttpClient()

        val params = RequestParams()
        params.put("id", mosqueId)
        params.put("ustadz_id", ustadzId)

        client.post(api, params, object :
AsyncHttpResponseHandler() {
            override fun onSuccess(statusCode: Int,
headers: Array<out Header>?, responseBody: ByteArray?) {
                Toast.makeText(context,
resources.getString(R.string.add_mosque_success) + mosque.mosqueName,
Toast.LENGTH_SHORT).show()

                binding.progressMessage.visibility =
View.VISIBLE

                GlobalScope.launch(Dispatchers.Main) {
                    val deferredStatus =
async(Dispatchers.IO) {

mosqueListViewModel.setMosques(requireActivity().applicationContext,
binding.etSearchMosque.text.toString(), true)
                    }
                    val status = deferredStatus.await()
                    if (status != null) {

binding.pullToRefresh.isRefreshing = false
                        val parse = JSONObject(status)
                        Toast.makeText(context,
parse.getString("message"), Toast.LENGTH_SHORT).show()
                    } else {

binding.pullToRefresh.isRefreshing = false

mosqueListAdapter.notifyDataSetChanged()
                    }
                }
            }

            override fun onFailure(statusCode: Int,
headers: Array<out Header>?, responseBody: ByteArray?, error:
Throwable?) {
                Toast.makeText(context,
resources.getString(R.string.add_mosque_failed) + mosque.mosqueName,
Toast.LENGTH_SHORT).show()

                binding.progressMessage.visibility =
View.GONE
            }
        })
    }
}
super.onActivityResult(requestCode, resultCode, data)

```

```
}  
}
```

- SettingActivity.java

```
- package com.kajianid.ustadz.ui.settings;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.MenuItem;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.preference.ListPreference;  
import androidx.preference.Preference;  
import androidx.preference.PreferenceFragmentCompat;  
  
import com.google.android.material.snackbar.BaseTransientBottomBar;  
import com.google.android.material.snackbar.Snackbar;  
import com.kajianid.ustadz.R;  
import com.kajianid.ustadz.data.Credential;  
import com.kajianid.ustadz.data.Ustadz;  
import com.kajianid.ustadz.databinding.ActivitySettingsBinding;  
import com.kajianid.ustadz.prefs.CredentialPreference;  
import com.kajianid.ustadz.ui.credential.EditIdentityActivity;  
import com.kajianid.ustadz.ui.credential.EditPasswordActivity;  
import com.kajianid.ustadz.ui.login.LoginActivity;  
import com.loopj.android.http.AsyncHttpClient;  
import com.loopj.android.http.AsyncHttpResponseHandler;  
  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.Objects;  
  
import cz.msebera.android.httpclient.Header;  
  
public class SettingsActivity extends AppCompatActivity {  
  
    private ActivitySettingsBinding binding;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        binding =  
ActivitySettingsBinding.inflate(getLayoutInflater());  
        setContentView(binding.getRoot());  
        Objects.requireNonNull(getSupportFragmentManager())  
            .beginTransaction()  
            .replace(R.id.settings, new SettingsFragment())  
            .commit();  
        setSupportActionBar(binding.toolbar);  
    }  
}
```

```

Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);

Objects.requireNonNull(getSupportActionBar()).setTitle(getString(R.string.settings));
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home)
            super.onBackPressed();
        return true;
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        binding = null;
    }

    public static class SettingsFragment extends
    PreferenceFragmentCompat {

        @Override
        public void onCreatePreferences(Bundle savedInstanceState,
        String rootKey) {
            setPreferencesFromResource(R.xml.root_preferences,
            rootKey);

            ListPreference theme = findPreference("theme");
            Preference changePassword =
            findPreference("changePassword");
            Preference changeIdentity =
            findPreference("changeIdentity");
            Preference logout = findPreference("logout");

            theme.setOnPreferenceChangeListener((preference,
            newValue) -> {
                switch (newValue.toString()) {
                    case "dark":

                        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);

                        break;
                    case "light":

                        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);

                        break;
                }

                return true;
            });

            changeIdentity.setOnPreferenceClickListener(it -> {
                CredentialPreference preference = new
                CredentialPreference(requireContext());
                Snackbar.make(requireView().getRootView(),

```

```

getString(R.string.please_wait),
BaseTransientBottomBar.LENGTH_SHORT).show();
        String api = getString(R.string.server) +
"api/ustadz/identity/" + preference.getCredential().getUsername();
        AsyncHttpClient client = new AsyncHttpClient();

        client.get(api, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[]
headers, byte[] responseBody) {
                try {
                    String result = new String(responseBody);
                    JSONObject resultObject = new
JSONObject(result);

                                Ustadz ustadz = new Ustadz();

ustadz.setName(resultObject.getString("name"));
ustadz.setAddress(resultObject.getString("address"));
ustadz.setEmail(resultObject.getString("email"));
ustadz.setPhone(resultObject.getString("phone"));
ustadz.setGender(resultObject.getString("gender"));

                                Intent i = new Intent(getContext(),
EditIdentityActivity.class);
i.putExtra(EditIdentityActivity.EXTRA_PARCEL_USTADZ, ustadz);
                                startActivity(i);
                                } catch (JSONException e) {

Snackbar.make(requireView().getRootView(),
getString(R.string.failed_load_data) + ": Failed Parsing",
BaseTransientBottomBar.LENGTH_SHORT).show();
                                }
                            }

            @Override
            public void onFailure(int statusCode, Header[]
headers, byte[] responseBody, Throwable error) {
                Snackbar.make(requireView().getRootView(),
getString(R.string.failed_load_data),
BaseTransientBottomBar.LENGTH_SHORT).show();
            }
        });

        return true;
    });

    changePassword.setOnPreferenceClickListener(it -> {
        Intent i = new Intent(getContext(),
EditPasswordActivity.class);
        startActivity(i);

        return true;
    });
}

```



```

        logout.setOnPreferenceClickListener(it -> {
            new AlertDialog.Builder(requireContext())
                .setTitle(getString(R.string.sure))

                .setMessage(getString(R.string.logout_confirm))
                .setPositiveButton(getString(R.string.yes),
(a, b) -> {
                    CredentialPreference preference = new
CredentialPreference(requireContext());
                    Credential credential = new Credential();

                    credential.setUsername("");
                    credential.setPassword("");
                    preference.setCredential(credential);

                    Toast.makeText(requireContext(),
getString(R.string.goodbye), Toast.LENGTH_SHORT).show();

                    Intent i = new Intent(getContext(),
LoginActivity.class);

                    i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(i);
                    requireActivity().finishAffinity();
                })
                .setNegativeButton(getString(R.string.no),
null)

                .create()
                .show();

            return true;
        });
    }
}

```

```

- DeveloperData.java
- package com.kajianid.ustadz.utils;

import androidx.annotation.DrawableRes;
import com.kajianid.ustadz.R;

public class DeveloperData {
    @DrawableRes
    public static final int[] drawableId = {
        R.drawable.dwichan,
        R.drawable.maratun,
        R.drawable.fendi,
        R.drawable.diyah, // Diyah Primasari
        R.drawable.bayu
    };

    public static final String[] name = {
        "Dwi Candra Permana",
        "Mar Atun Solehah",
        "Fendi Riawan",
        "Diyah Primasari",
    }
}

```

```

        "Muhammad Rizky Bayu Pratama",
    };

    public static final String[] nim = {
        "18.11.0004",
        "18.11.0017",
        "18.11.0268",
        "18.11.0289",
        "18.11.0258",
    };

    public static final String[] kelas = {
        "IF 2018 A",
        "IF 2018 A",
        "IF 2018 A",
        "IF 2018 A",
        "IF 2018 A",
    };

    public static final String[] yearRegister = {
        "2018",
        "2018",
        "2018",
        "2018",
        "2018",
    };

    public static final String[] githubUsername = {
        "dwichan0905",
        "maratun1725",
        "fendi-riawan",
        "DiyahPrimasari0289",
        "rizkybayu1411",
    };

    public static final String[] email = {
        "dwichan@outlook.com",
        "marhatunsoleha17@gmail.com",
        "fendiriawan0102@gmail.com",
        "diyahprimasari24@gmail.com",
        "rizkybayu1411@gmail.com",
    };

    public static final String[] address = {
        "Jalan Mandor Martinem Roy 1 Nomor 99 RT 005 RW 006 Desa  
Rejasari Kecamatan Langensari Kota Banjar",
        "Desa Losari Rt 04 Rw 04 Kecamatan Rawalo Kabupaten  
Banyumas",
        "Kemutug Lor RT 04 RW 03 Kecamatan Baturraden",
        "Kasih, RT 004 RW 003, Kecamatan Kertanegara, Kabupaten  
Purbalingga",
        "Desa Papringan Kec. Banyumas, Kabupaten Banyumas, Jawa  
Tengah",
    };
}

```

- StringHelper.java

```
- package com.kajianid.ustadz.utils;

import android.util.Patterns;

public class StringHelper {
    public static boolean isNullOrEmpty(String what) {
        return (what == null) || (what.trim().equals("")) ||
what.equals("null");
    }

    public static boolean isValidEmail(String email) {
        if (email.equals(null)) return false;
        return Patterns.EMAIL_ADDRESS.matcher(email).matches();
    }

    public static boolean convertToBoolean(String what) {
        return what.equals("1") || what.equalsIgnoreCase("TRUE");
    }
}
```

- AndroidManifest.xml

```
- <?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.kajianid.ustadz">

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

    <application
        android:allowBackup="true"
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:requestLegacyExternalStorage="true"
        android:roundIcon="@drawable/icon"
        android:supportsRtl="true"
        android:theme="@style/Theme.KajianID">
        <activity
            android:name=".ui.mosque.MosqueChooserActivity"
            android:theme="@style/MosqueChooserDialog" />
        <activity android:name=".ui.kajian.ShowKajianActivity" />
        <activity android:name=".ui.settings.SettingsActivity" />
        <activity android:name=".ui.article.ReadArticleActivity" />
        <activity android:name=".ui.login.LoginActivity" />
        <activity android:name=".ui.intro.IntroActivity" />
        <activity
            android:name=".ui.credential.EditPasswordActivity"
            android:label="@string/change_password"
            android:theme="@style/ChangePasswordDialog" />
        <activity
            android:name=".ui.credential.EditIdentityActivity"
            android:label="@string/change_identity"
```

```

        android:theme="@style/ChangeIdentityDialog" />
        <activity android:name=".ui.kajian.AddUpdateKajianActivity"
/>
        <activity android:name=".ui.article.AddUpdateArticleActivity"
/>
        <activity android:name=".ui.main.MainActivity" />
        <activity android:name=".ui.about.AboutActivity" />
        <activity android:name=".ui.intro.SplashActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **A. KESIMPULAN**

Aplikasi KAJIAN.ID USER merupakan aplikasi yang dapat digunakan oleh masyarakat untuk mengetahui info kajian rutin yang telah di jadwalkan, agar masyarakat tetap dapat melaksanakan kajian rutin secara online dengan aman dan mudah. Tak hanya itu, aplikasi ini juga dilengkapi dengan artikel islami yang dapat diunduh dan dibaca secara offline.

Aplikasi KAJIAN.ID USTADZ merupakan aplikasi yang dapat digunakan oleh ustadz/ustadzah untuk memudahkan penyampaian informasi kajian secara online. dan memudahkan para ustadz/ustadzah pula dalam membagikan artikel-artikel Islamiah.

Fitur-fitur yang terdapat dalam aplikasi KAJIAN.ID USER yaitu:

- Menyimpan artikel atau dan pengingat kajian Islami
- Menghapus artikel atau kajian yang telah disimpan
- Melihat masjid atau lokasi yang biasanya diadakan kajian.
- Tema terang/gelap yang membantu pengguna aplikasi
- Mengurangi cahaya yang ditangkap oleh mata
- Membagikan artikel atau kajian ke media sosial.

Fitur-fitur yang terdapat dalam aplikasi KAJIAN.ID USER yaitu:

- Menambah, mengubah, atau menghapus artikel Islami
- Menambah, mengubah, atau menghapus kajian yang akan atau telah dilaksanakan
- Mengelola masjid atau lokasi yang biasanya diadakan kajian.
- Tema terang/gelap yang membantu pengguna aplikasi ini mengurangi cahaya yang ditangkap oleh mata
- Mengunggah alamat Video atau Live Streaming dari YouTube
- Membagikan artikel atau kajian ke media sosial.

#### **B. SARAN**

Adapun saran-saran yang diberikan untuk penyempurnaan dan pengembangan aplikasi ini adalah sebagai berikut:

- a. Aplikasi ini sudah dapat digunakan dengan baik tanpa adanya error atau bug. Namun masih ada beberapa tambahan untuk pengembangan aplikasi kedepannya agar nantinya aplikasi ini dapat dikembangkan menjadi aplikasi yang lebih kompleks.
- b. Diharapkan untuk perkembangan kedepannya, aplikasi ini dapat ditambahkan dari sisi admin. Karena terkendala waktu maka kami perancang aplikasi ini belum dapat menyempurnakan aplikasi ini.