**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

[DAFTAR ISI 1](#_Toc61795784)

[BAB I: PENDAHULUAN 2](#_Toc61795785)

[A. LATAR BELAKANG 2](#_Toc61795786)

[B. RUMUSAN MASALAH 2](#_Toc61795787)

[C. TUJUAN 2](#_Toc61795788)

[D. MANFAAT 2](#_Toc61795789)

[BAB II: PEMBAHASAN 3](#_Toc61795790)

[A. KAJIAN.ID USER 3](#_Toc61795791)

[B. KAJIAN.ID USTADZ 7](#_Toc61795792)

[BAB III: *SOURCE CODE* 23](#_Toc61795793)

[A. ALAMAT *REPOSITORY* 23](#_Toc61795794)

[B. KAJIAN.ID USER 23](#_Toc61795795)

[C. KAJIAN.ID USTADZ 143](#_Toc61795796)

[BAB IV: KESIMPULAN DAN SARAN 144](#_Toc61795797)

[A. KESIMPULAN 144](#_Toc61795798)

[B. SARAN 144](#_Toc61795799)

# BAB I: PENDAHULUAN

## LATAR BELAKANG

Di masa pandemi ini, kegiatan sehari-hari yang dilakukan masyarakat terbatasi. 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.

## RUMUSAN MASALAH

1. aaa
2. bbb
3. ccc

## TUJUAN

1. aaa
2. bbb
3. ccc

## MANFAAT

# BAB II: PEMBAHASAN

## 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.

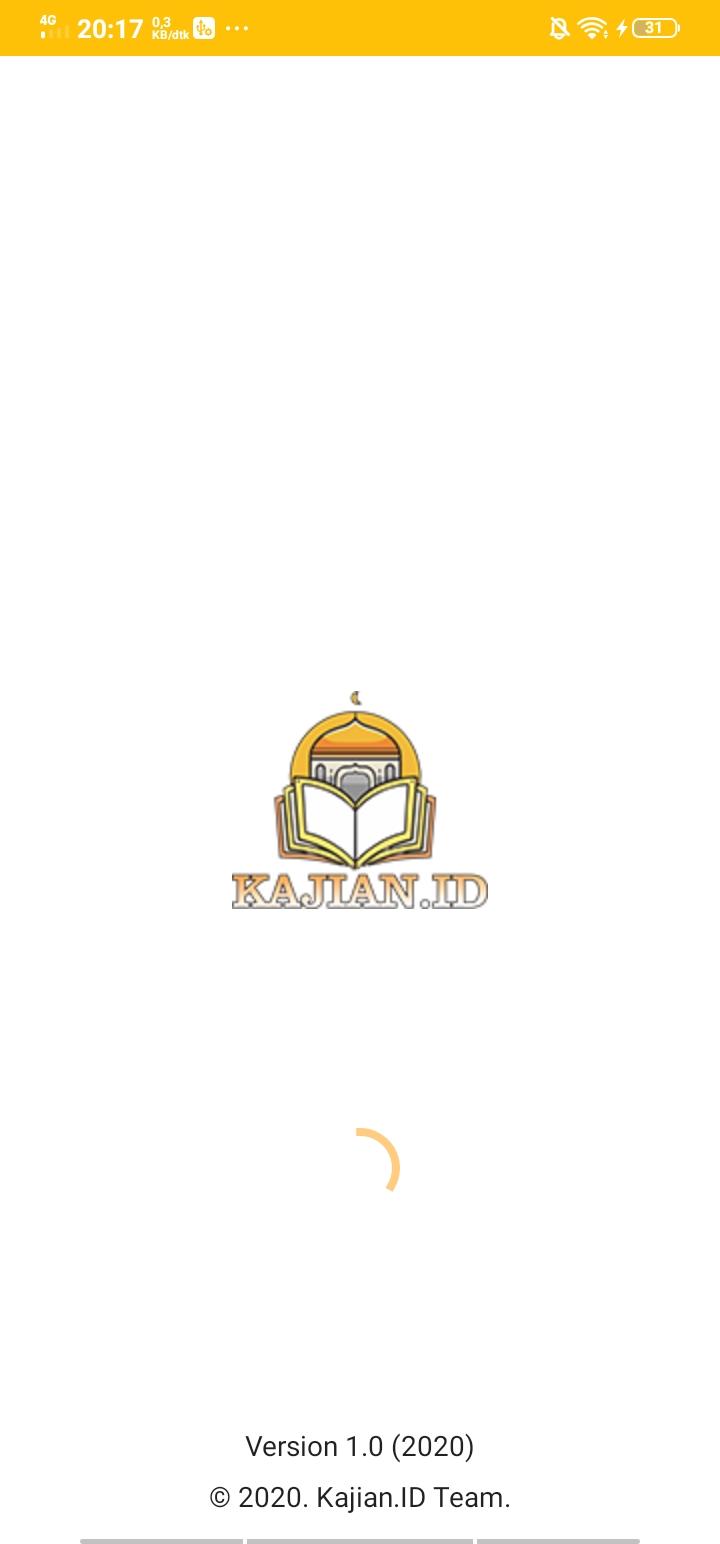
Berikut fitur-fitur yang dibawakan di Kajian.ID:

* 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.

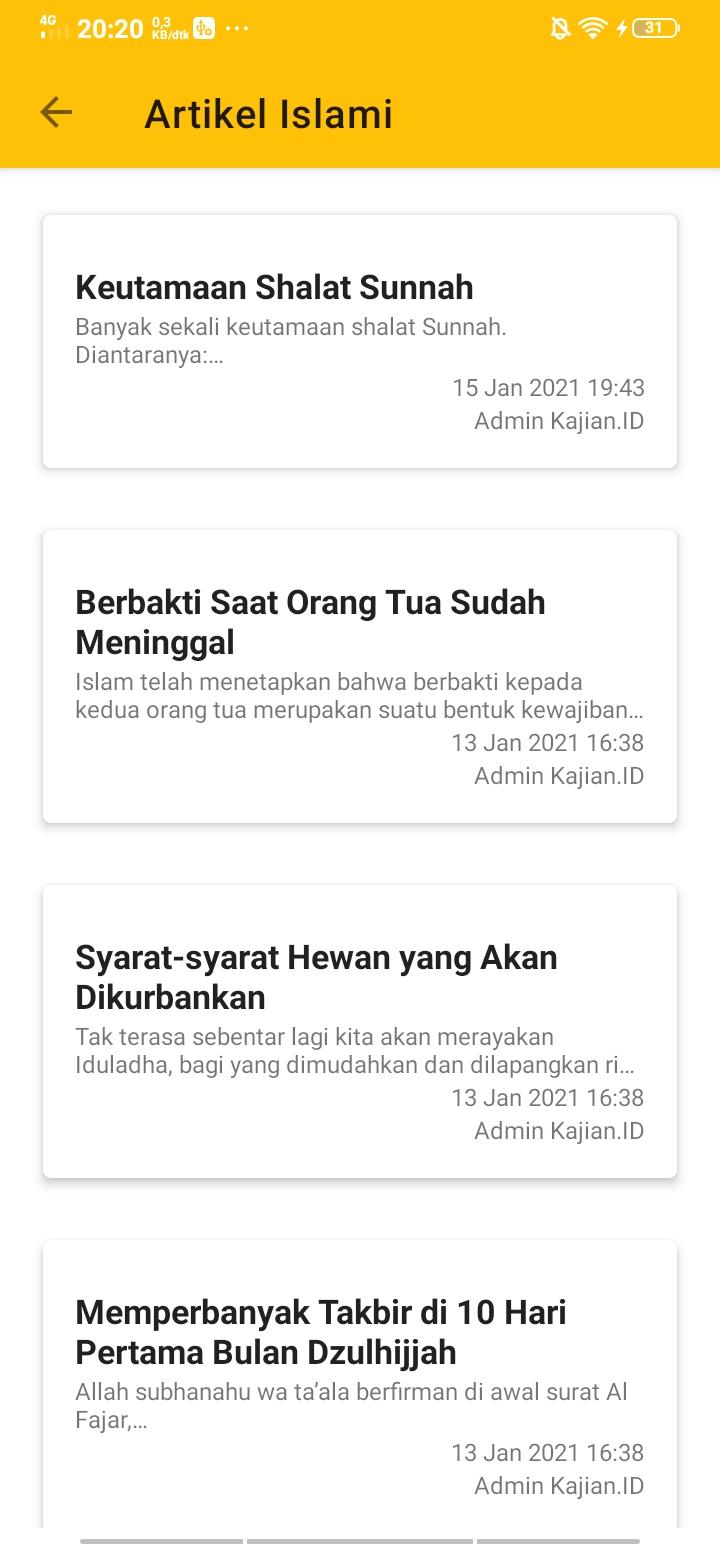
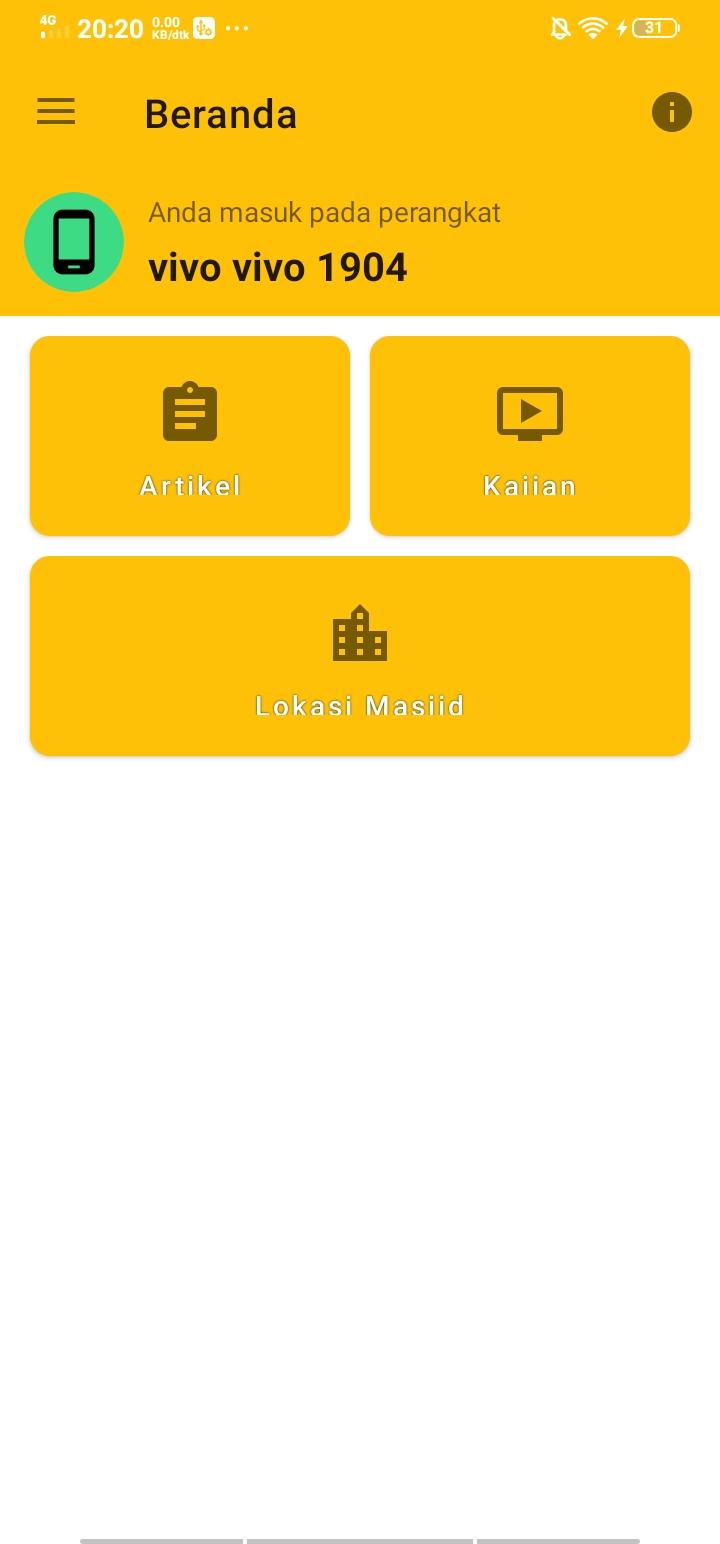
1. Langkah awal sebelum menggunakan aplikasi

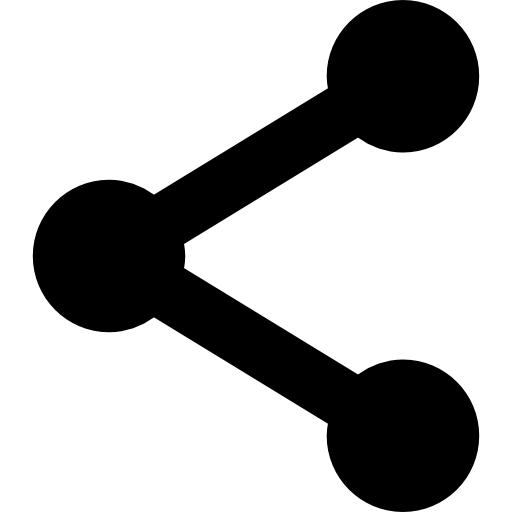
Saat pertama kali menginstal aplikasi ini, Anda akan dikenali sekilas mengenai aplikasi ini. Tunggu loading tampilan awal aplikasi membuka halaman selanjutnya, sudah selesai.

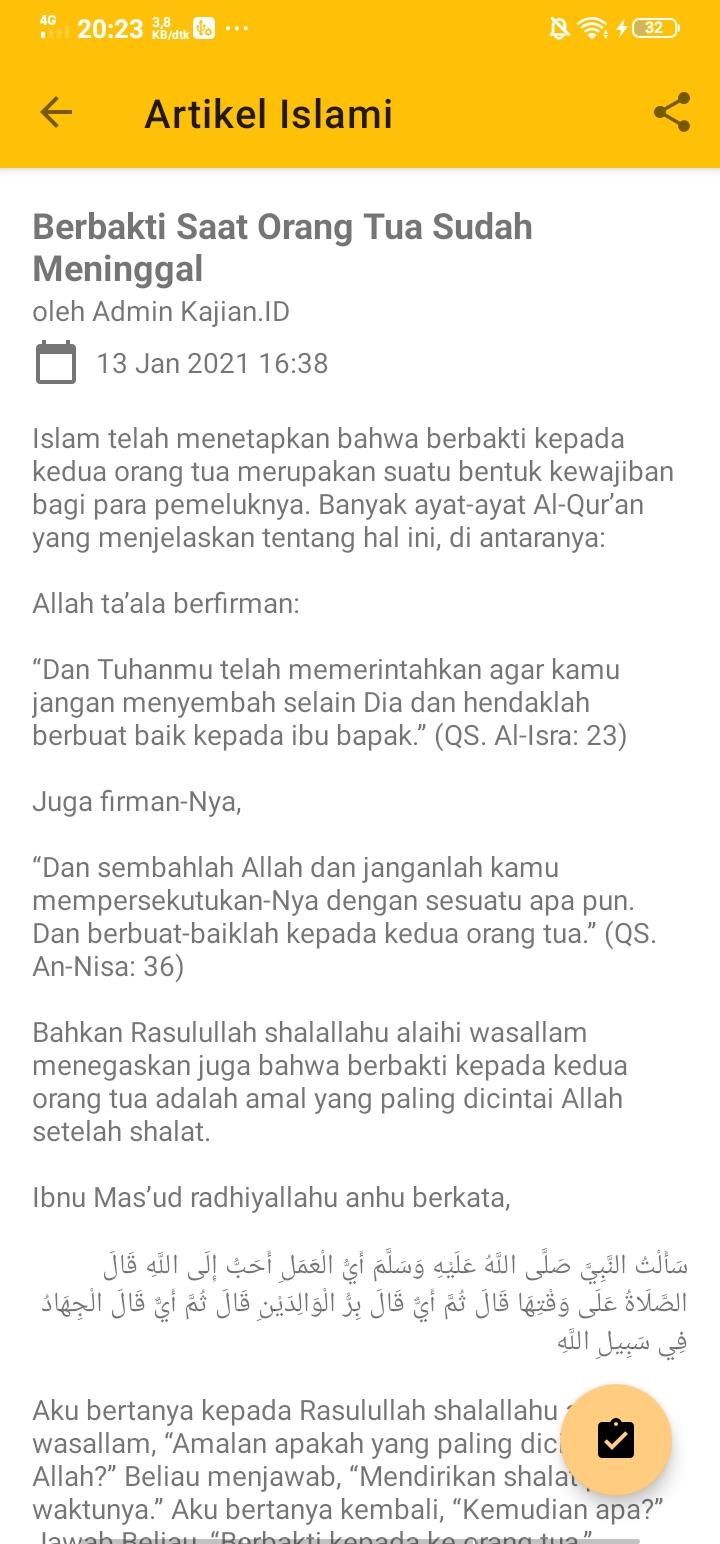


1. Cara membaca artikel, menyimpan artikel dan menyebarluaskan artikel Anda.

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

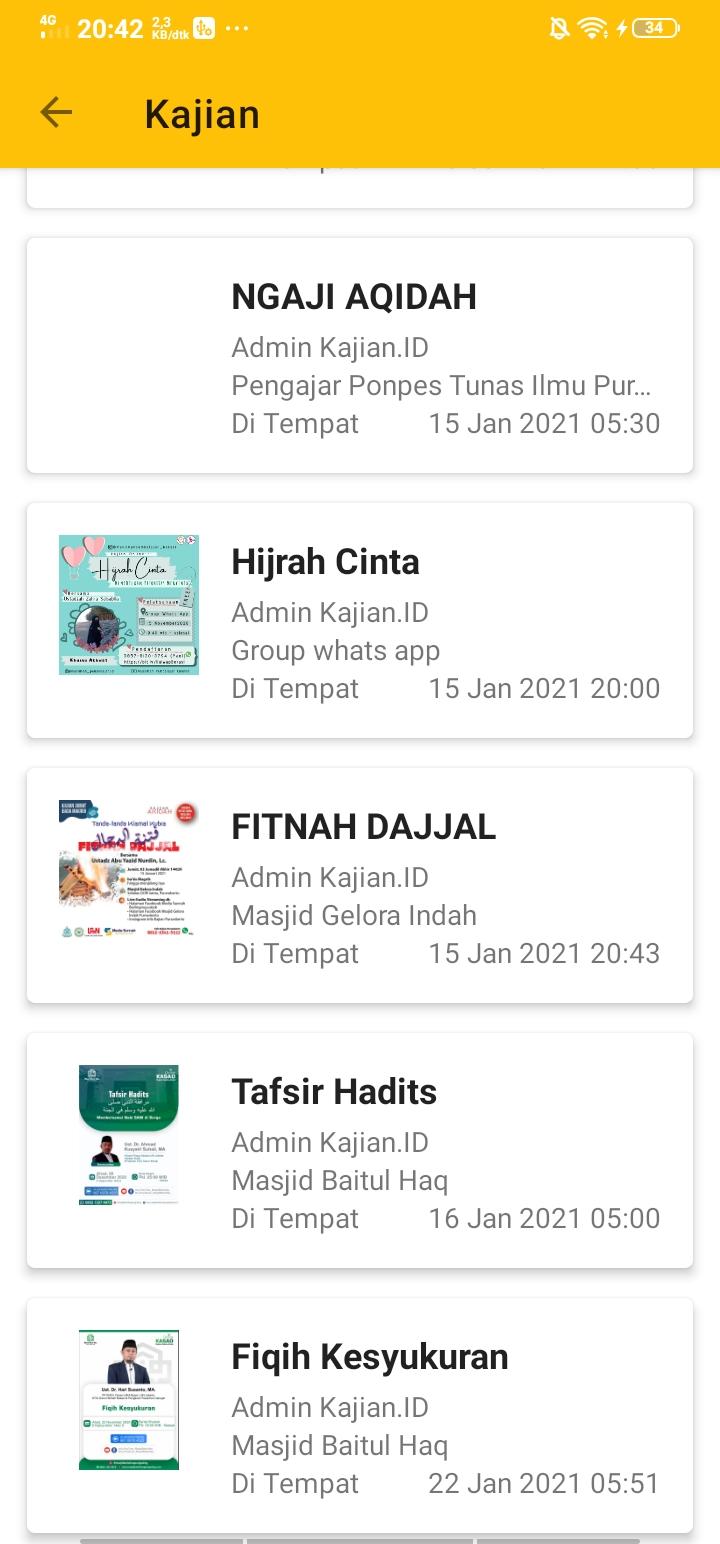
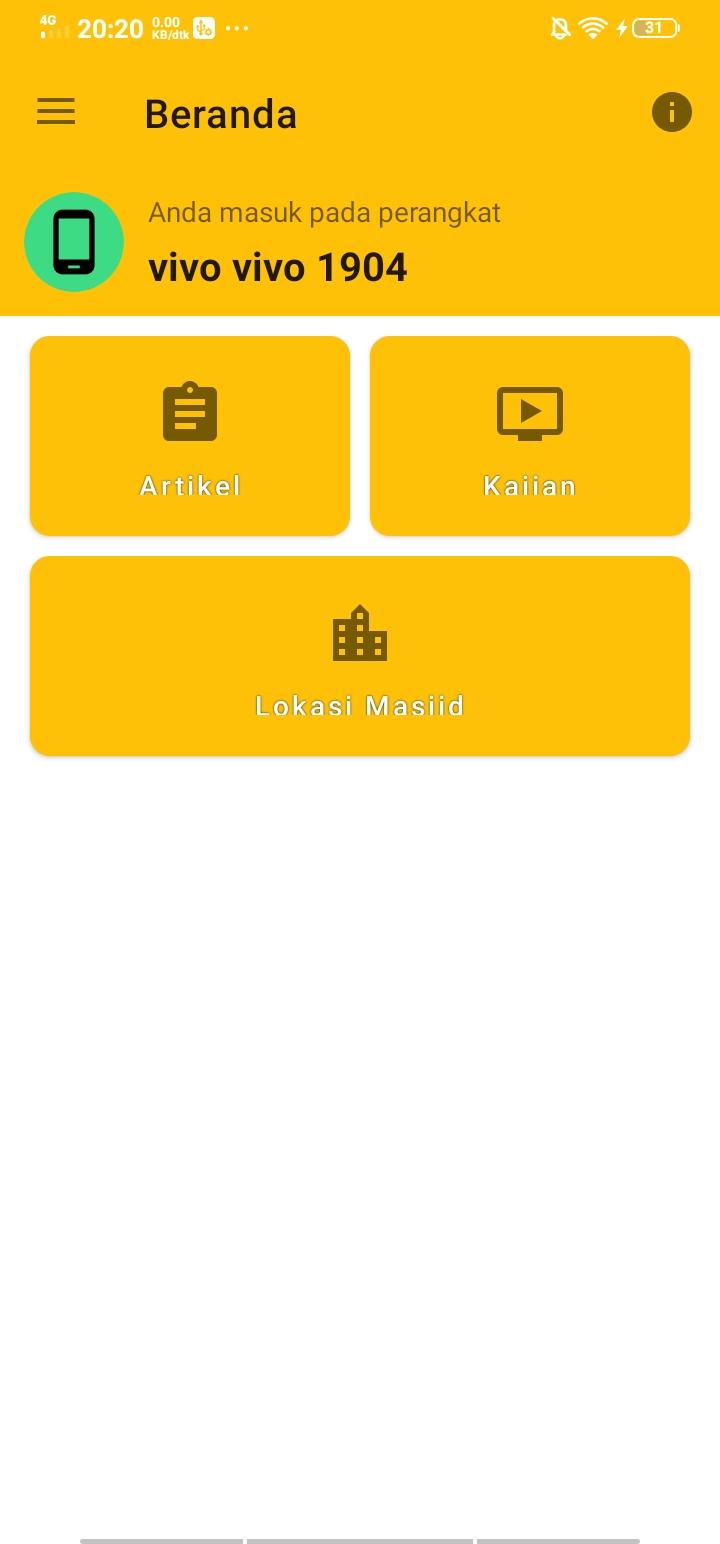


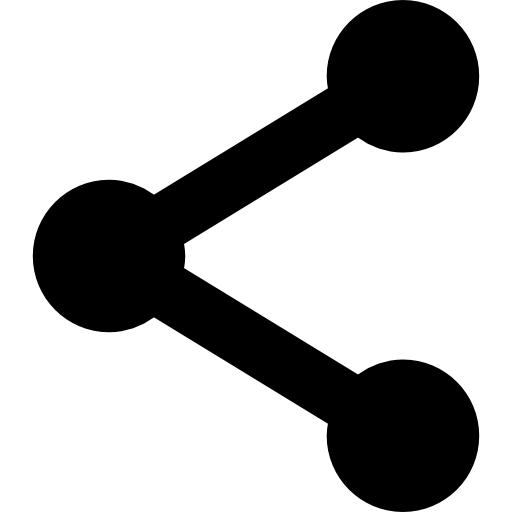
Untuk Menyimpan artikel yang akan disimpan tekan tombol “Simpan Post” di pojok kanan bawah layar. dan untuk membagikan atau menyebarluaskan artikel yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca artikel dan pilih ke aplikasi mana artikel Anda akan dibagikan.



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

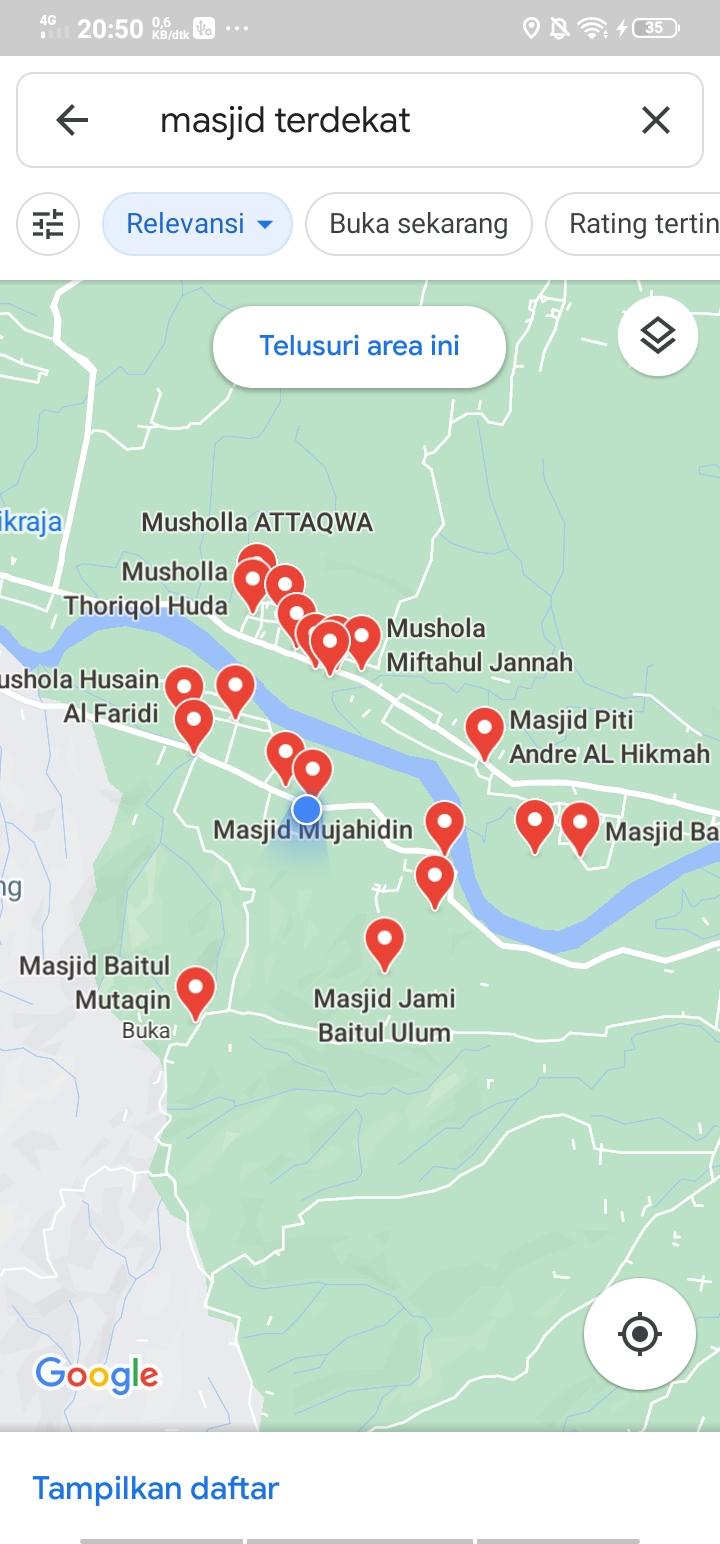
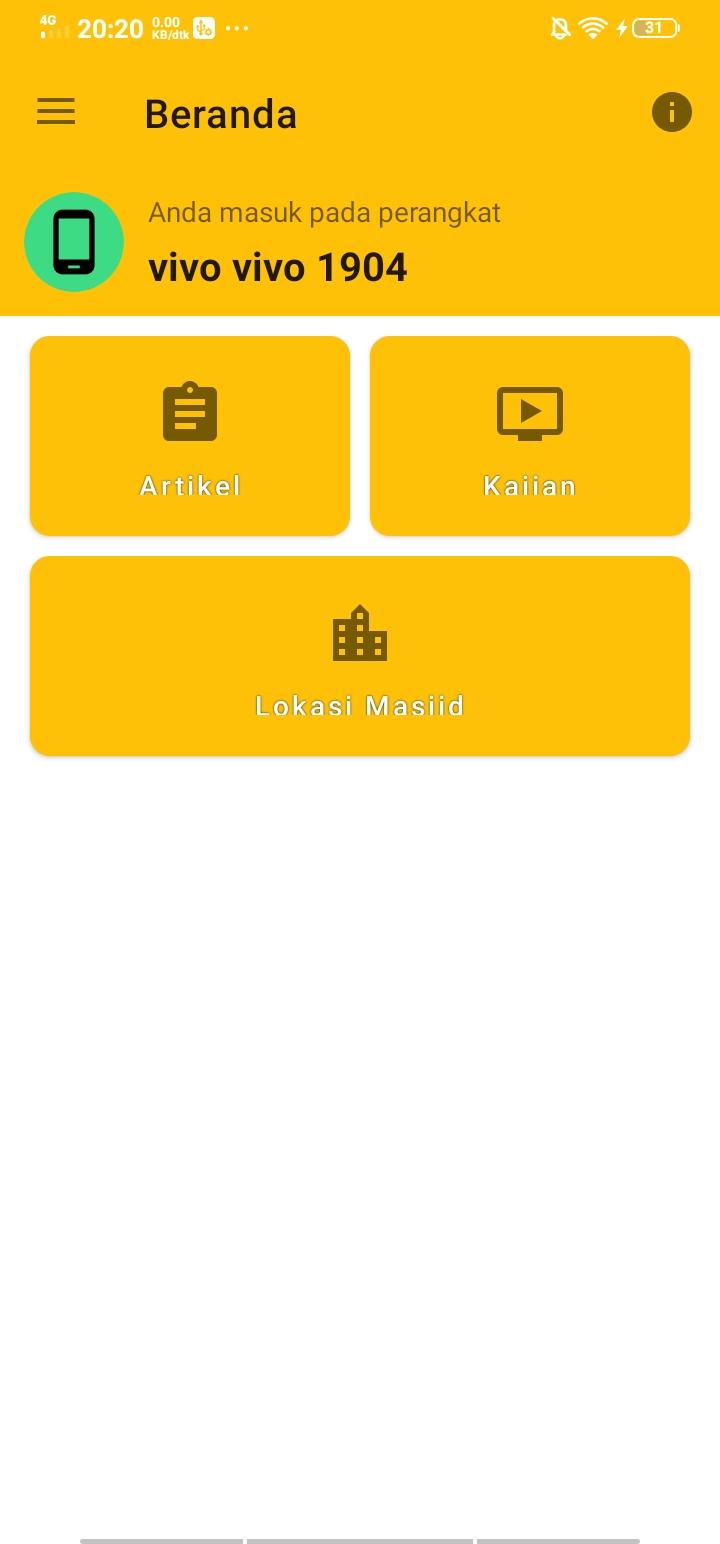
Di layar utama, pilih tab “Kajian” lalu pilih kajian yang akan dipilih.



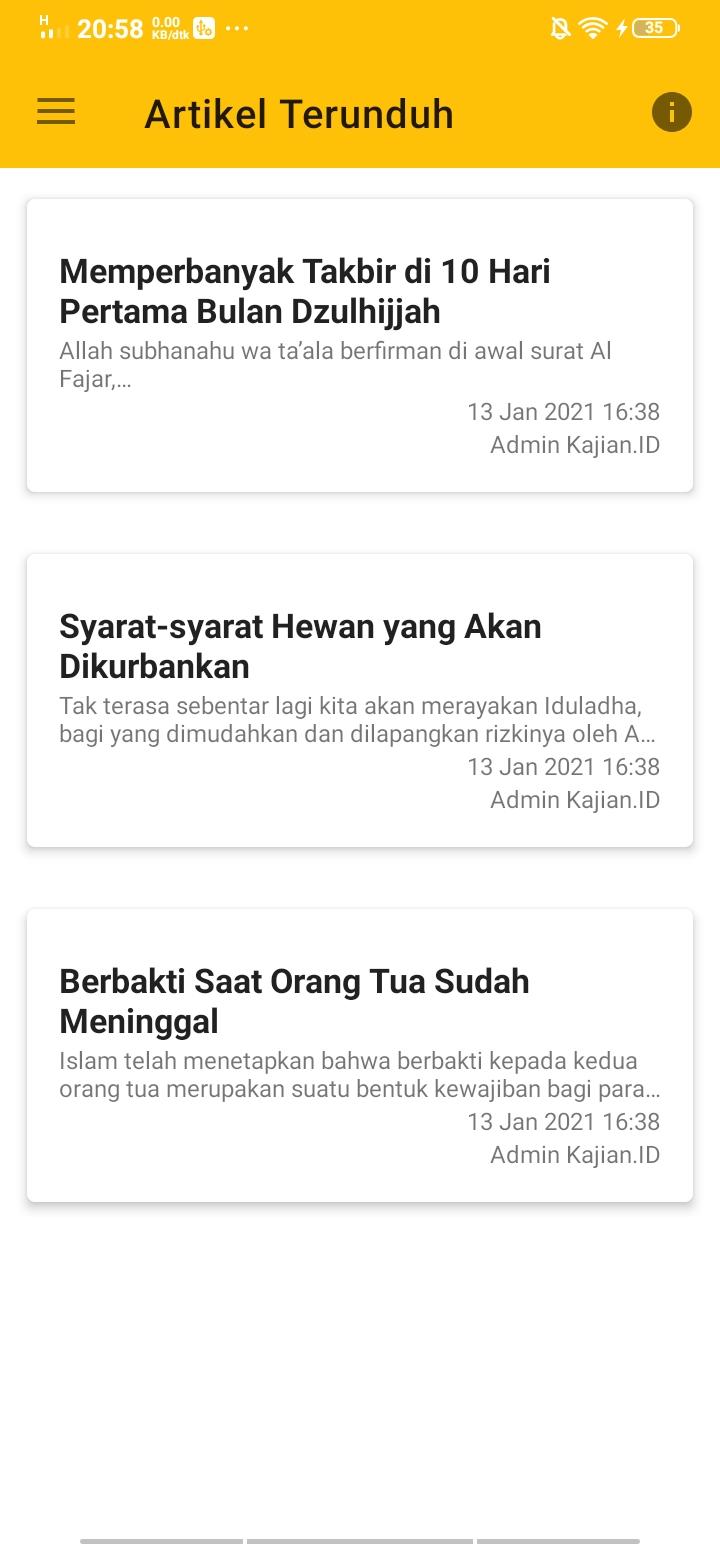
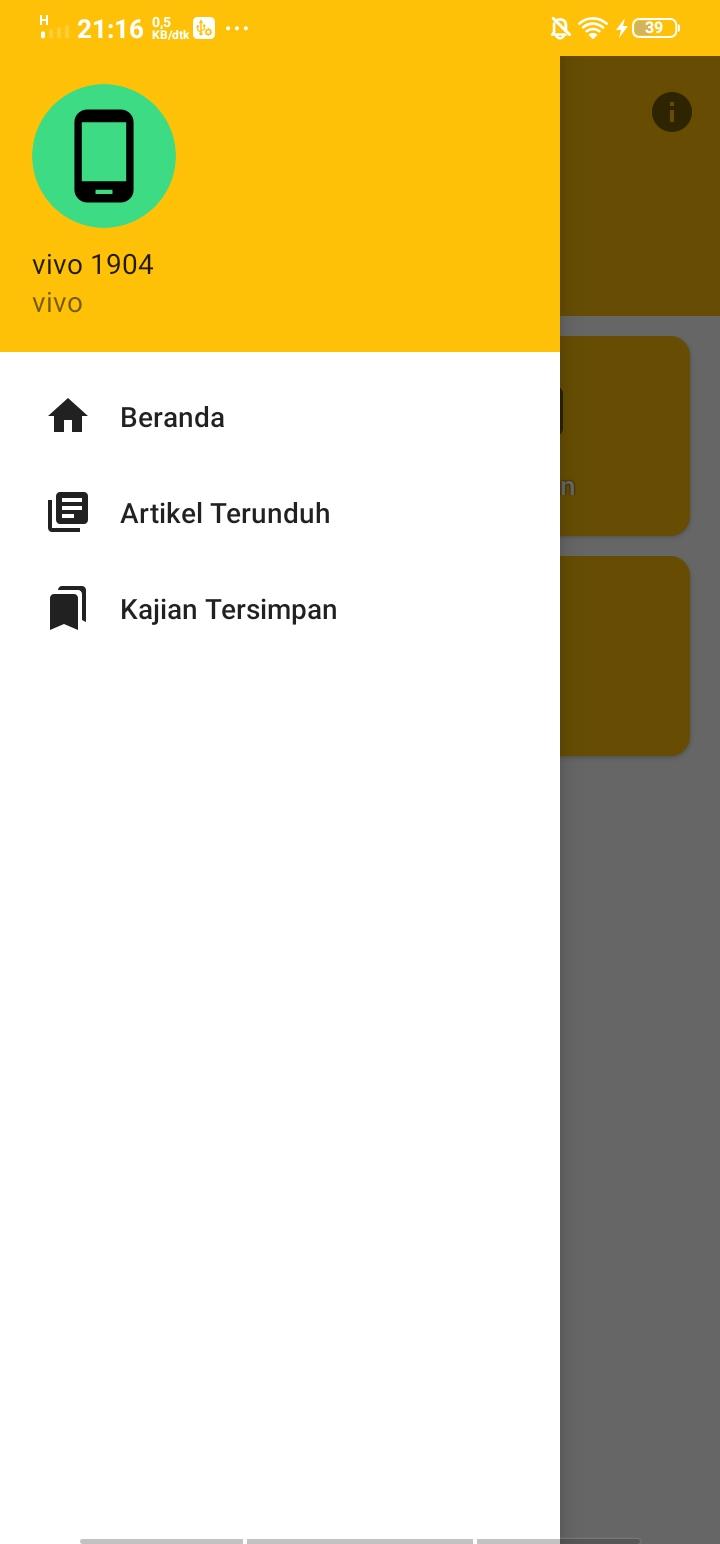
Untuk pengingat kajian yang sudah dipilih dan tekan tombol “Lonceng Post” di pojok kanan bawah layar. dan untuk membagikan atau menyebarluaskan artikel yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca artikel dan pilih ke aplikasi mana artikel Anda akan dibagikan.



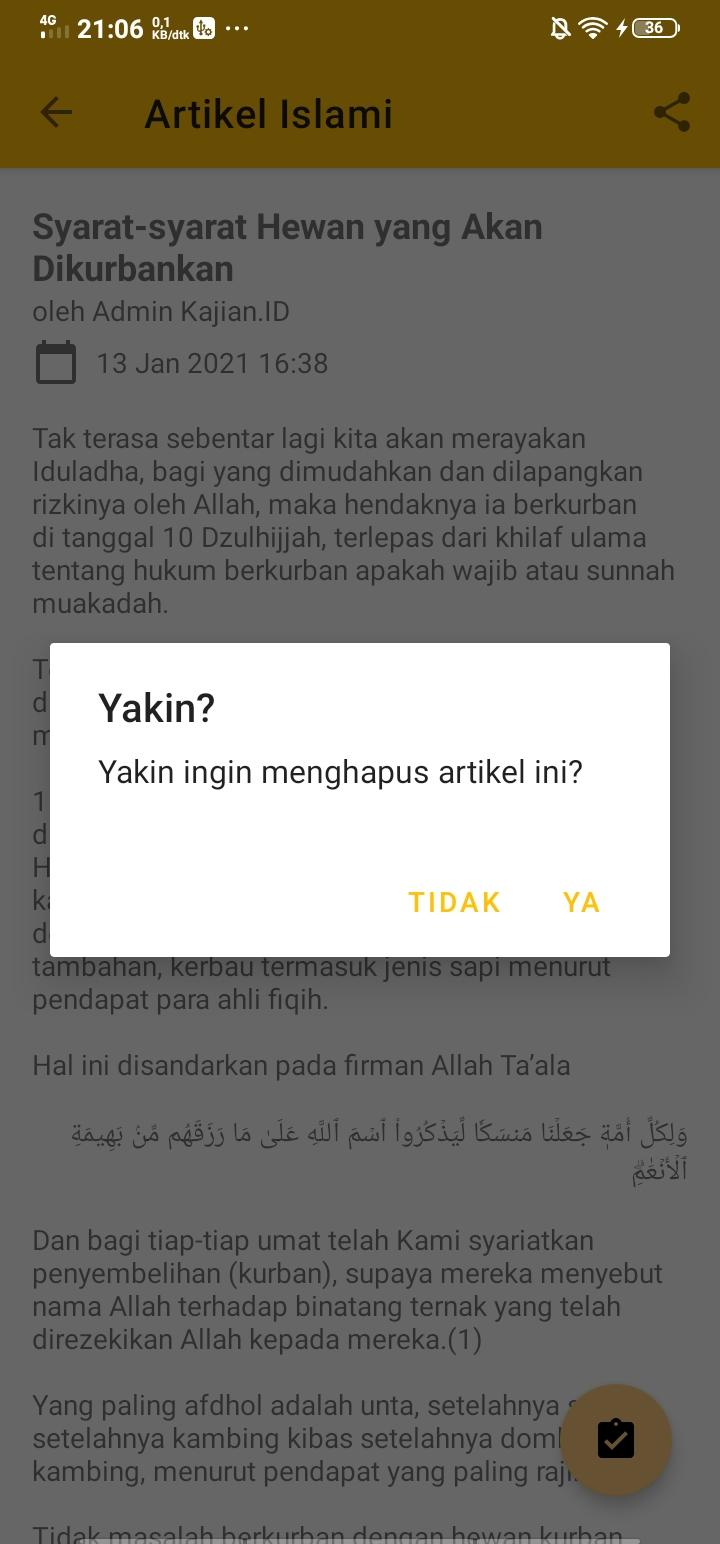
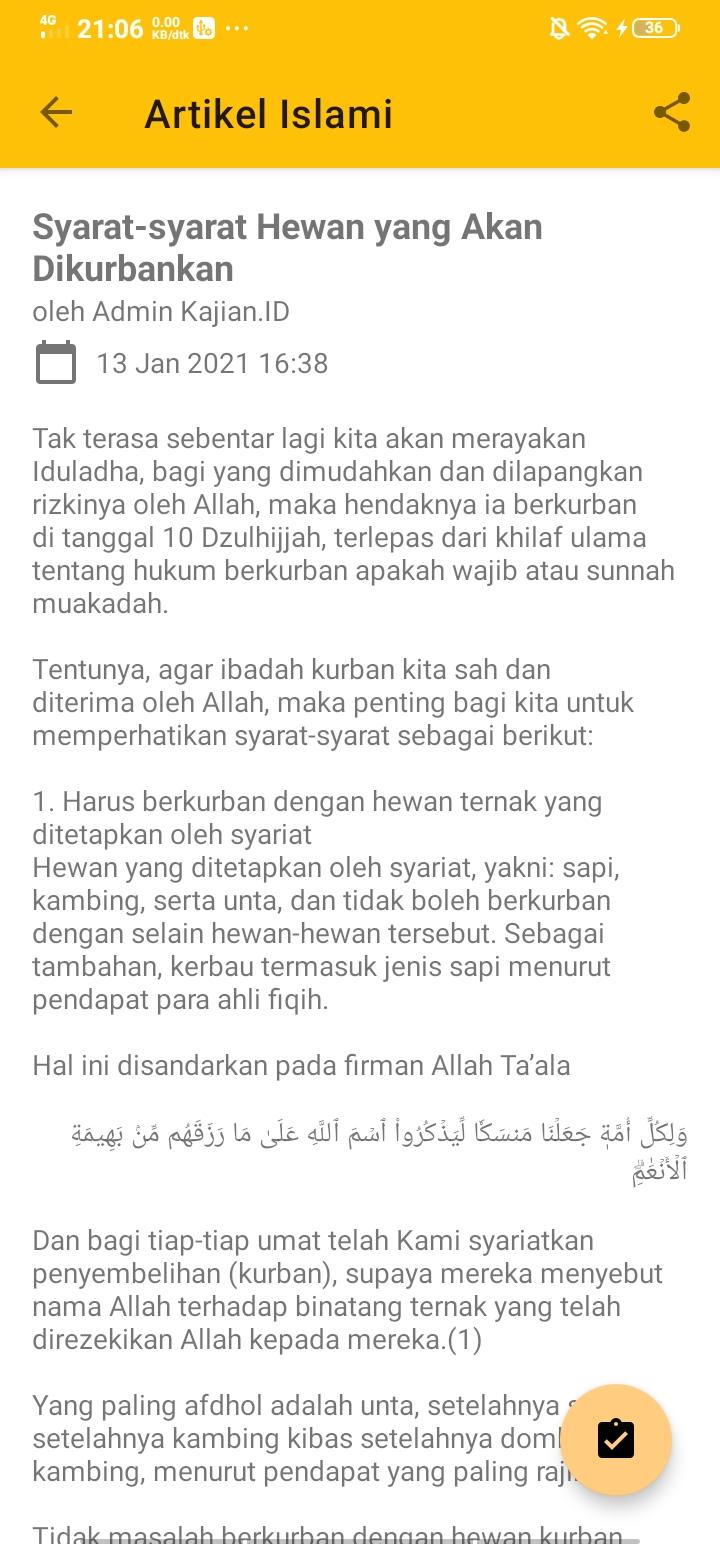
Untuk melihat masjid di sekitar kita, Di layar utama, pilih tab “Lokasi Masjid” lalu pilih Lokasi.



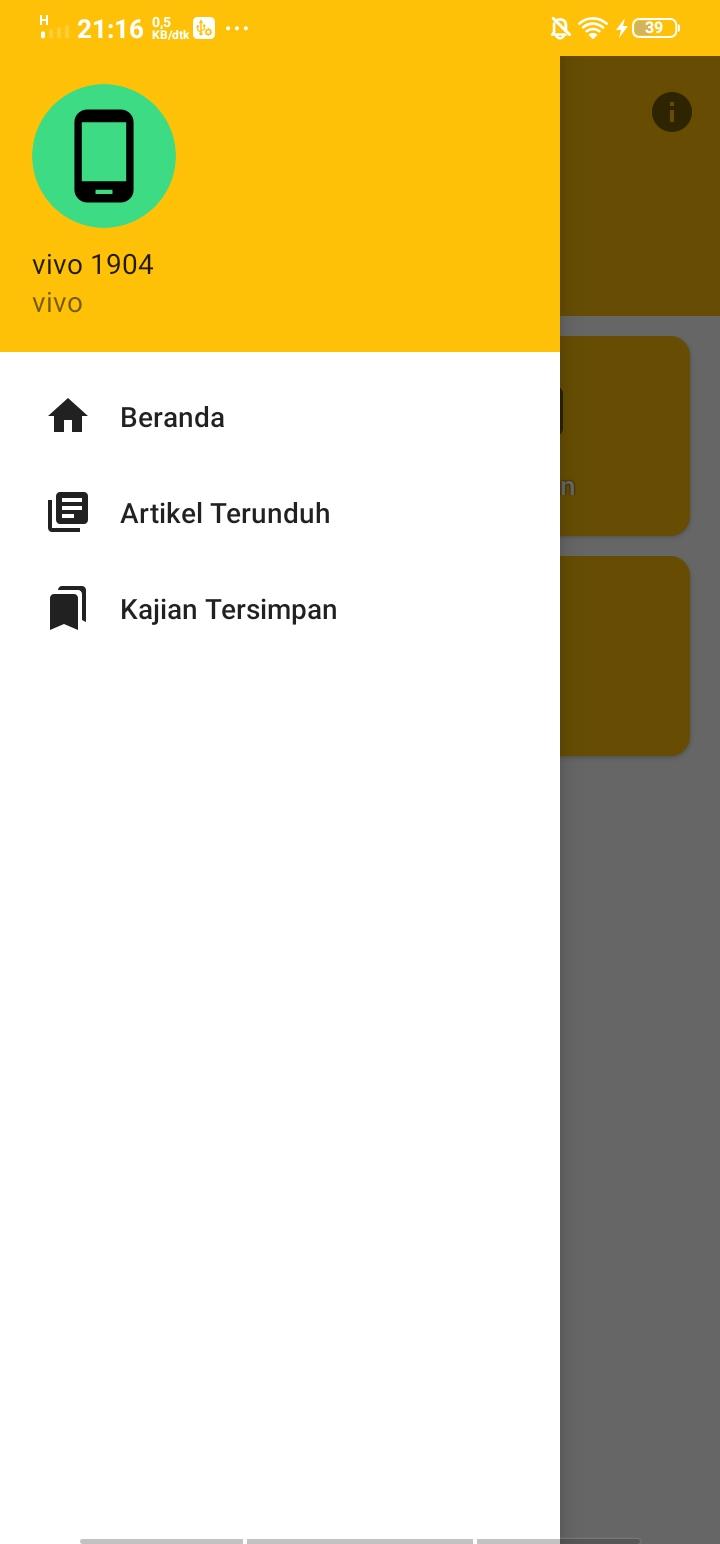
1. Jika kalian ingin melihat Artikel yang sudah tersimpan, silahkan tekan tombol pada layar utama pojok kiri atas, lalu pilih “Artikel Terunduh”



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



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



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

## 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.

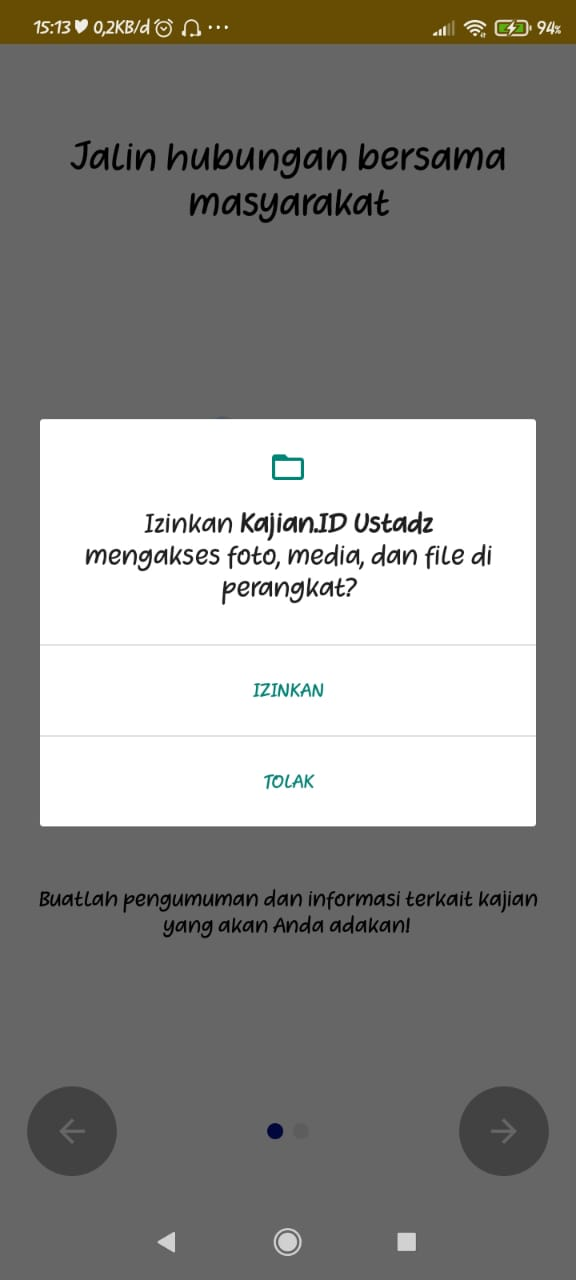
Berikut adalah tata cara penggunaan aplikasi ini.

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 perkenalan sudah selesai.



* 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.



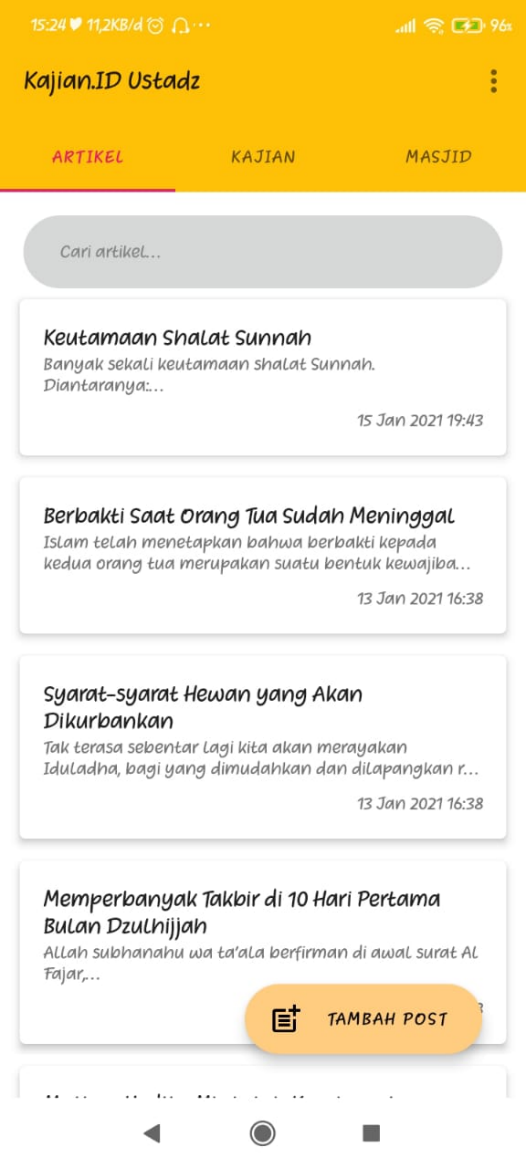
* Setelah layar perkenalan selesai, anda akan diarahkan menuju layar Masuk sebagai Ustadz. Silakan masukkan Username dan Kata Sandi (*password*) dalam kolom yang sudah disediakan. Apabila Anda belum memiliki akun, Anda bisa menekan tombol “Daftar” untuk menuju aplikasi WhatsApp dan kirim pesan serta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



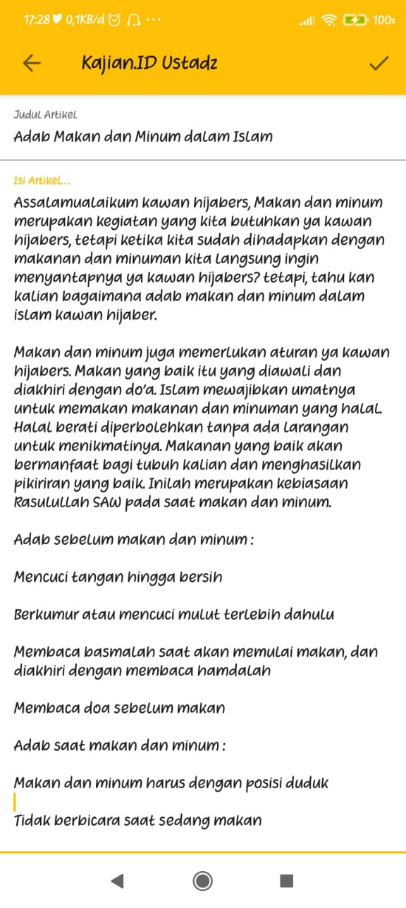
* Aplikasi akan menyimpan status login terakhir Anda dan akan selalu diingat sampai Anda melakukan *Logout* dari aplikasi.

1. Cara mengunggah artikel baru, membaca artikel, dan menyebarluaskan artikel Anda

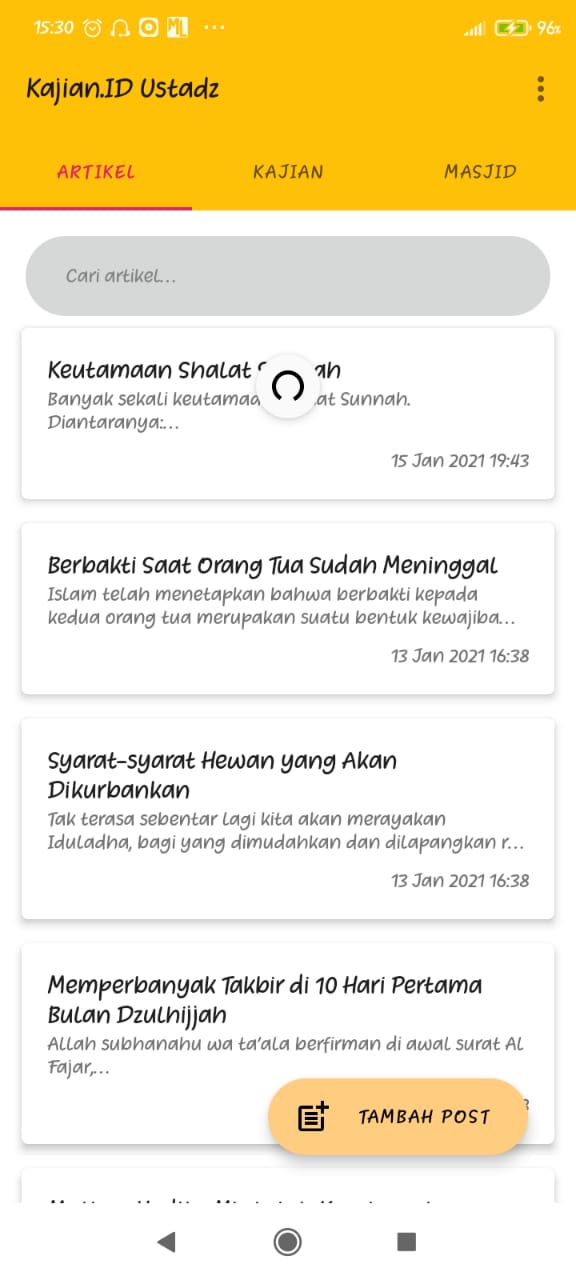
* Di layar utama, pilih tab “Artikel” dan tekan tombol “Tambah Post” di pojok kanan bawah layar.

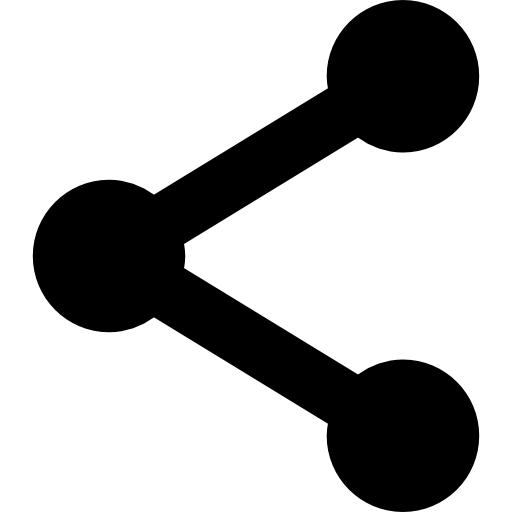


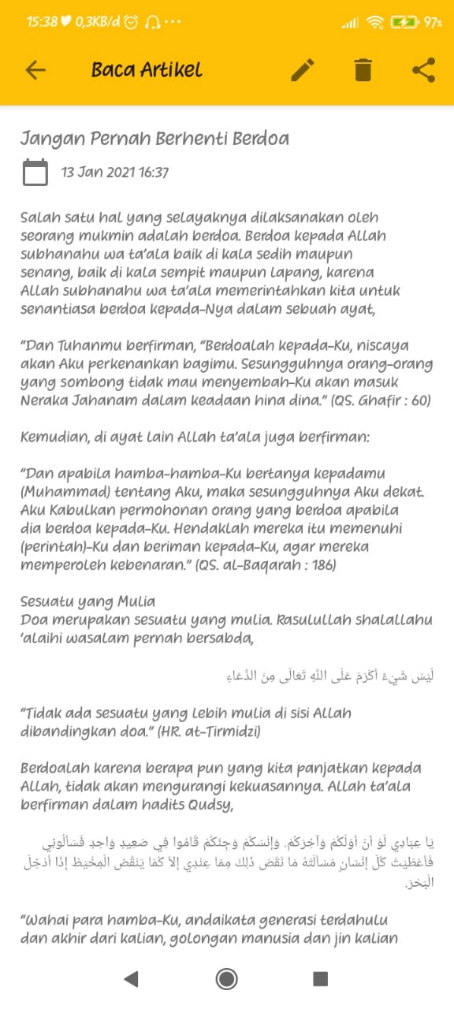
* Silahkan isi judul dan isi artikel pada halaman yang sudah disediakan. Setelah selesai mengisi, tekan tombol **✓** di pojok kanan atas layar.



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

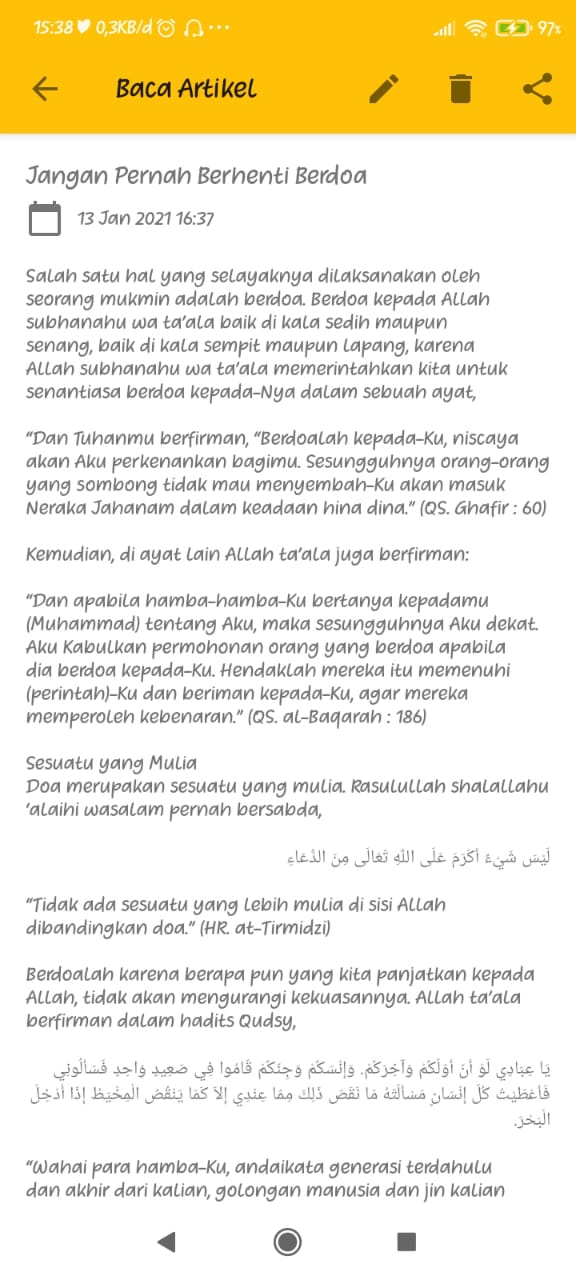


* Untuk melihat isi artikel yang sudah atau pernah Anda buat, cukup pilih artikel mana yang akan Anda baca.
* Untuk membagikan atau menyebarluaskan artikel yang Anda buat, cukup tekan tombol  di pojok kanan atas layar saat Anda membaca artikel dan pilih ke aplikasi mana artikel Anda akan dibagikan.



1. 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.

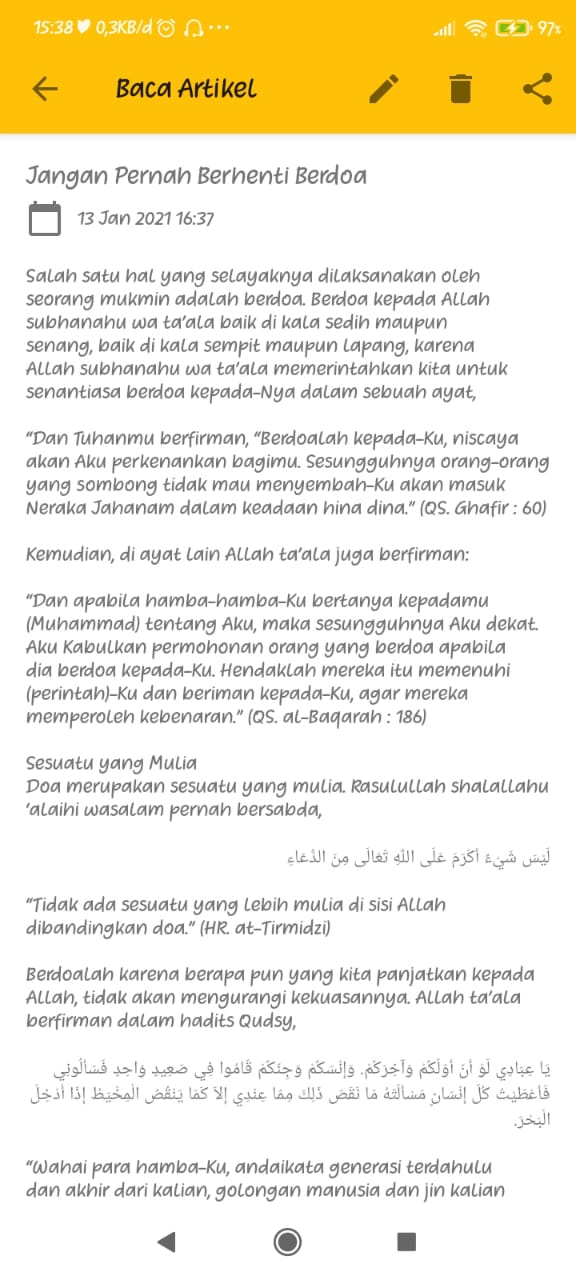


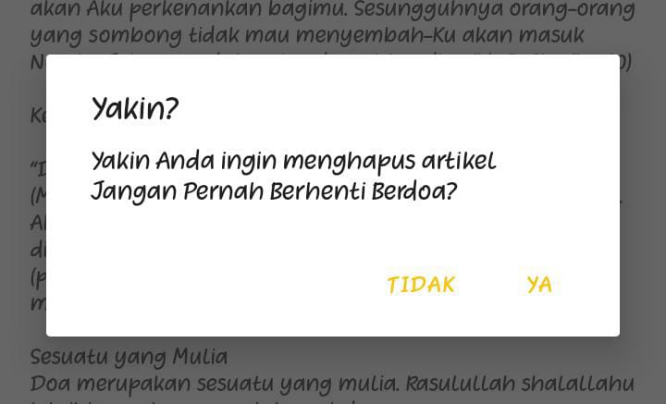
* Silahkan ubah judul atau isi artikel tersebut. Tekan tombol **✓** di pojok kanan atas layar apabila sudah selesai mengubah artikel.



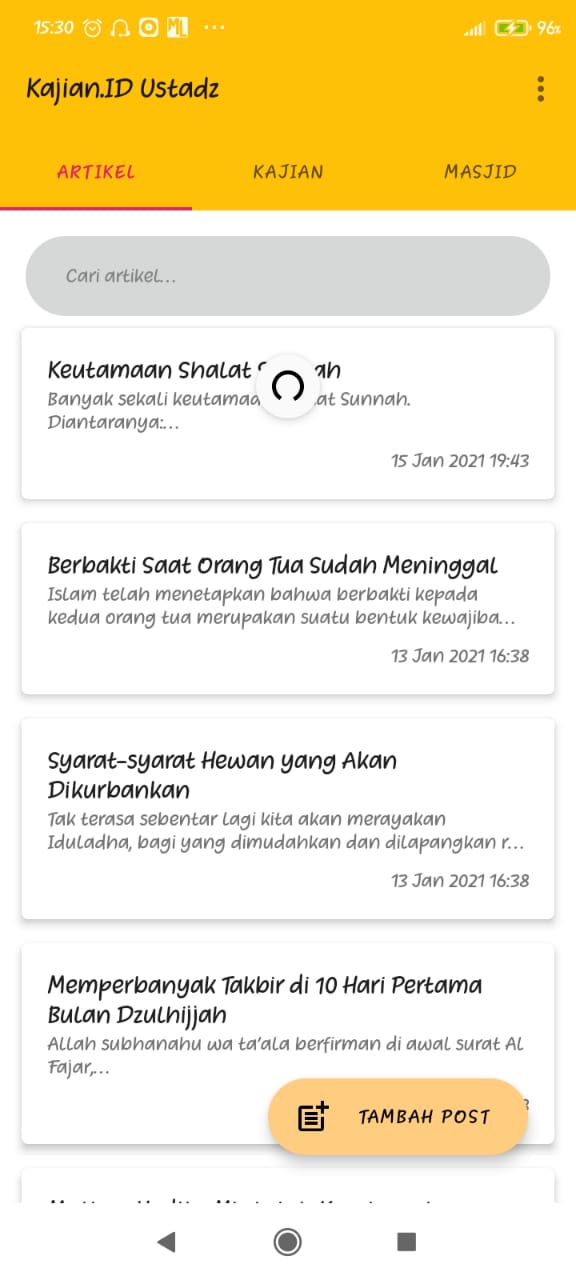
1. Cara menghapus artikel

* 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.



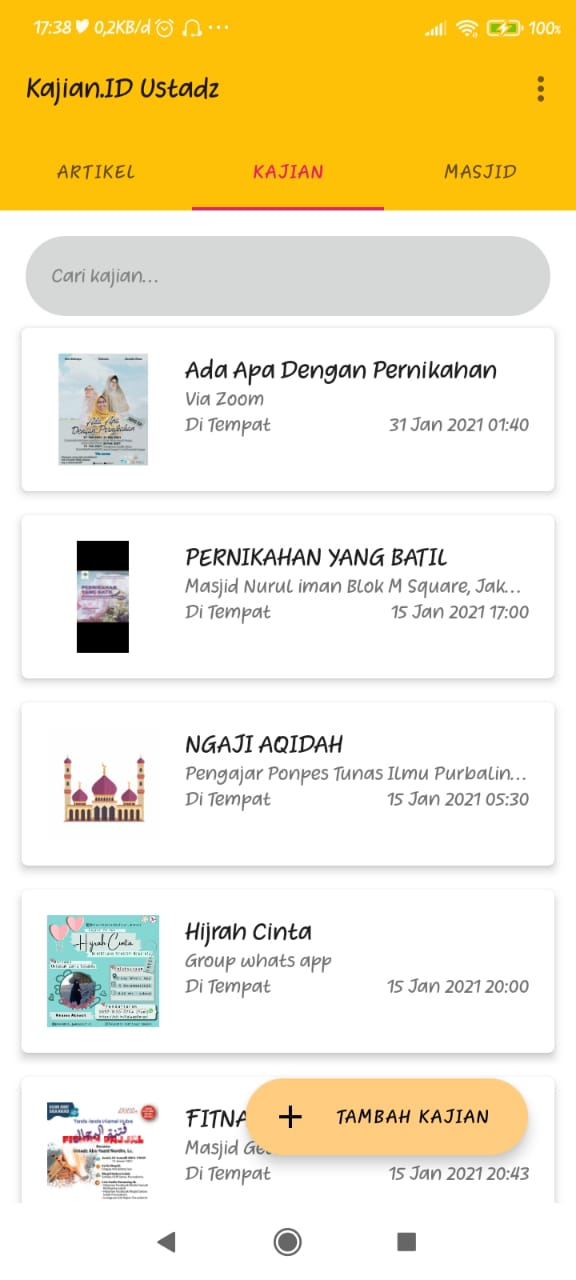


* Artikel berhasil dihapus. Untuk memperbarui daftar artikel, tarik daftar artikel dari atas ke bawah layar sampai muncul animasi indikator loading.

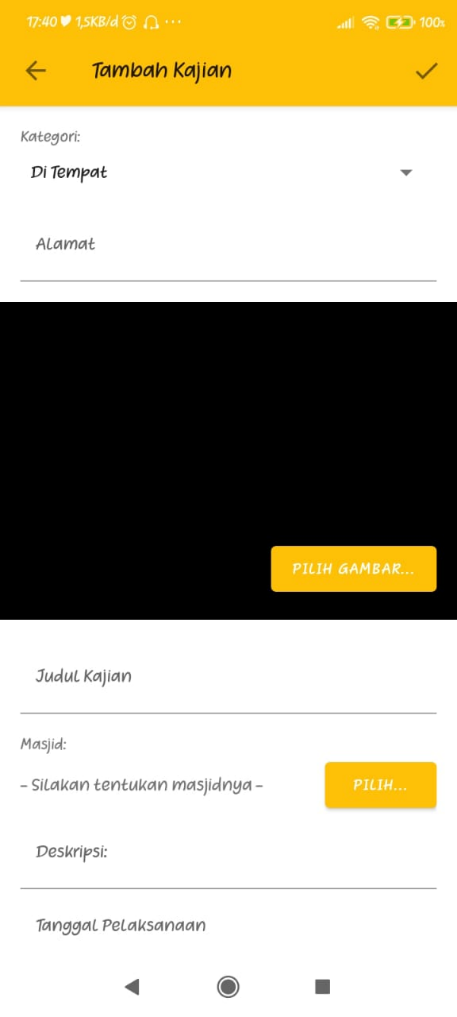


1. Cara mengunggah dan menyebarluaskan informasi kajian

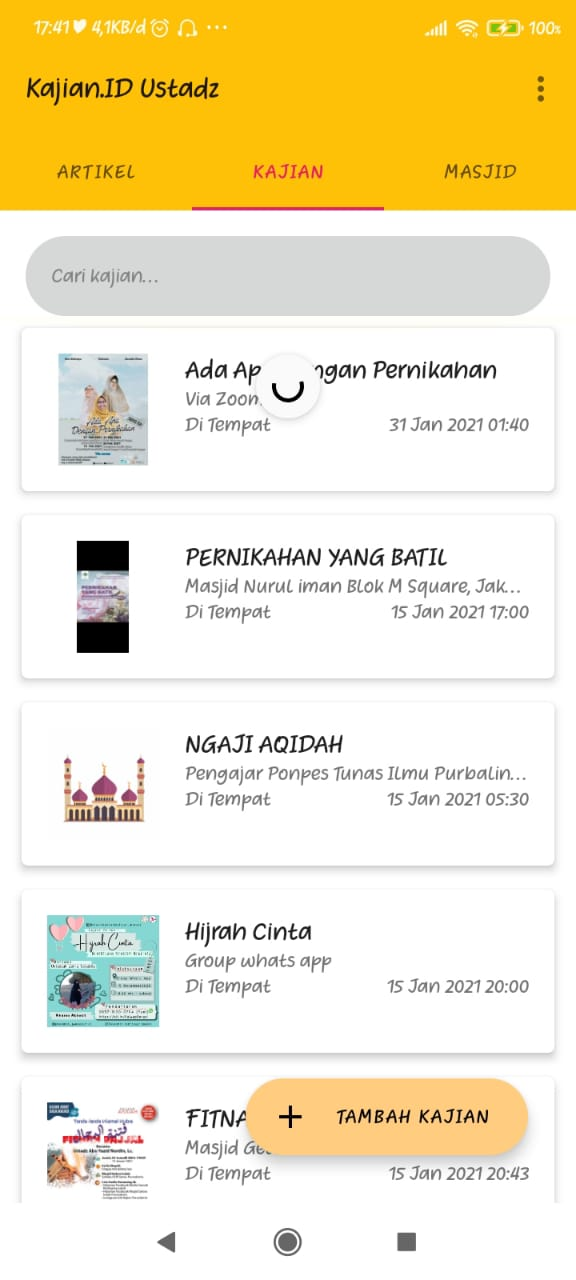
* Di layar utama, pilih tab “Kajian” dan tekan tombol “Tambah Kajian” di pojok kanan bawah layar.

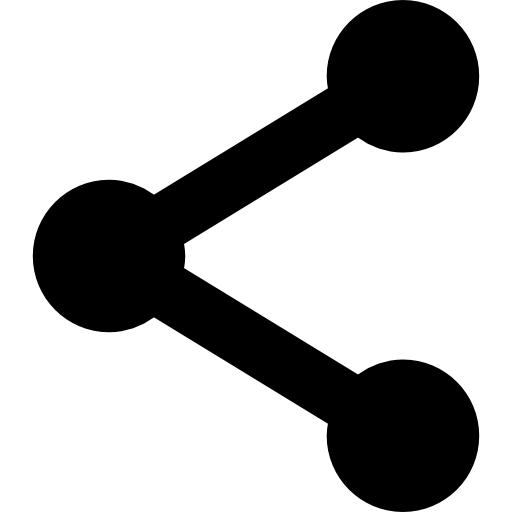


* Silahkan isi informasi mengenai kajian pada halaman yang sudah disediakan. Setelah selesai mengisi, tekan tombol **✓** di pojok kanan atas layar.



* 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.



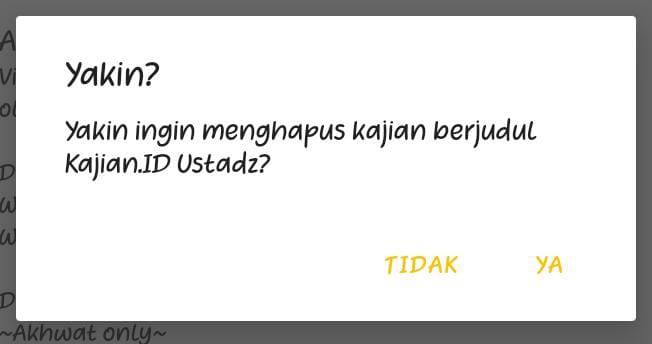
1. Cara mengubah atau memperbarui informasi kajian

* 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.



1. 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.



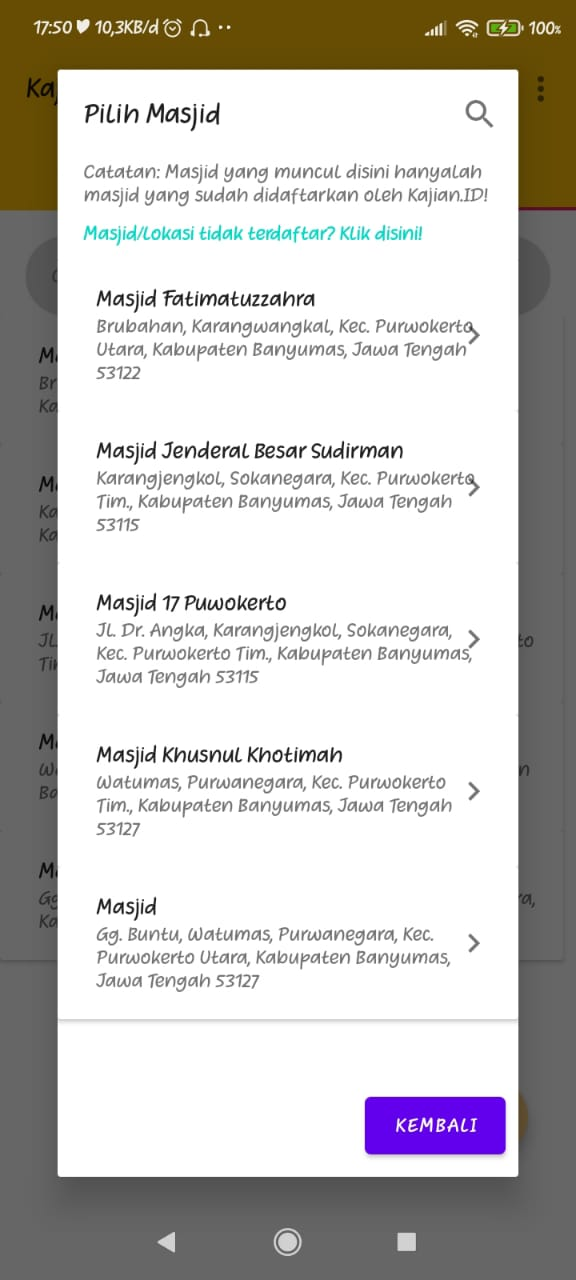
* Kajian berhasil dihapus. Untuk memperbarui daftar kajian, tarik daftar kajian dari atas ke bawah layar sampai muncul animasi indikator *loading*.

1. Cara menambah masjid atau lokasi untuk dikelola dalam kajian

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

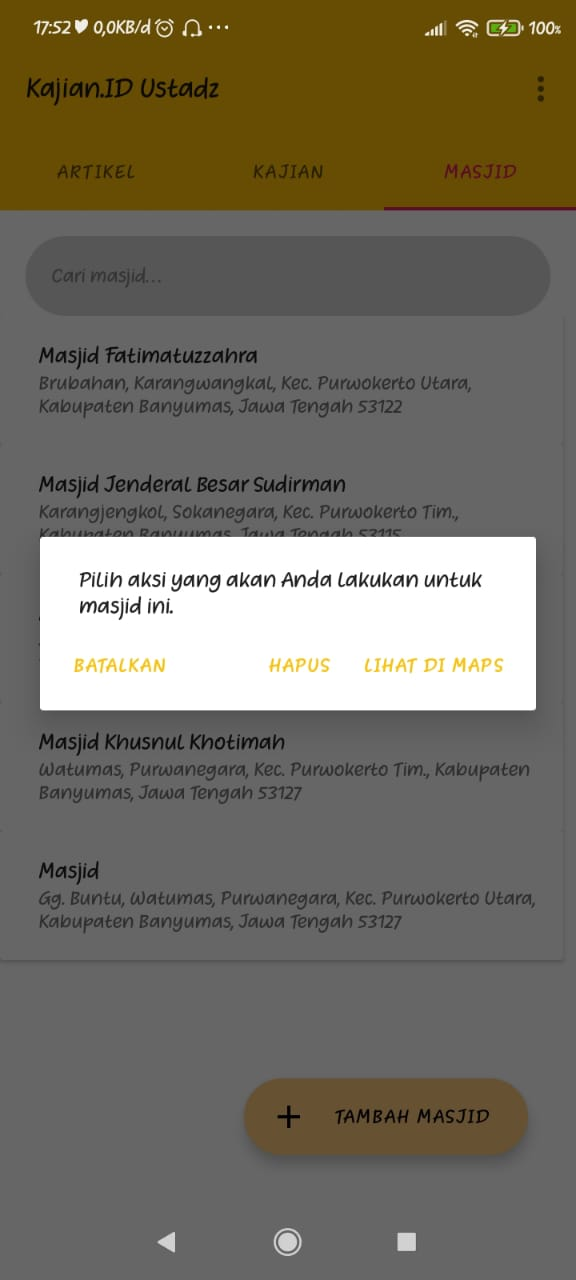


* Silahkan pilih masjid mana yang ingin Anda kelola. Masjid yang ada pada daftar ini hanya masjid yang sudah didaftarkan di server Kajian.ID. Untuk mendaftarkan lokasi baru, ikuti instruksi yang terdapat pada layar.

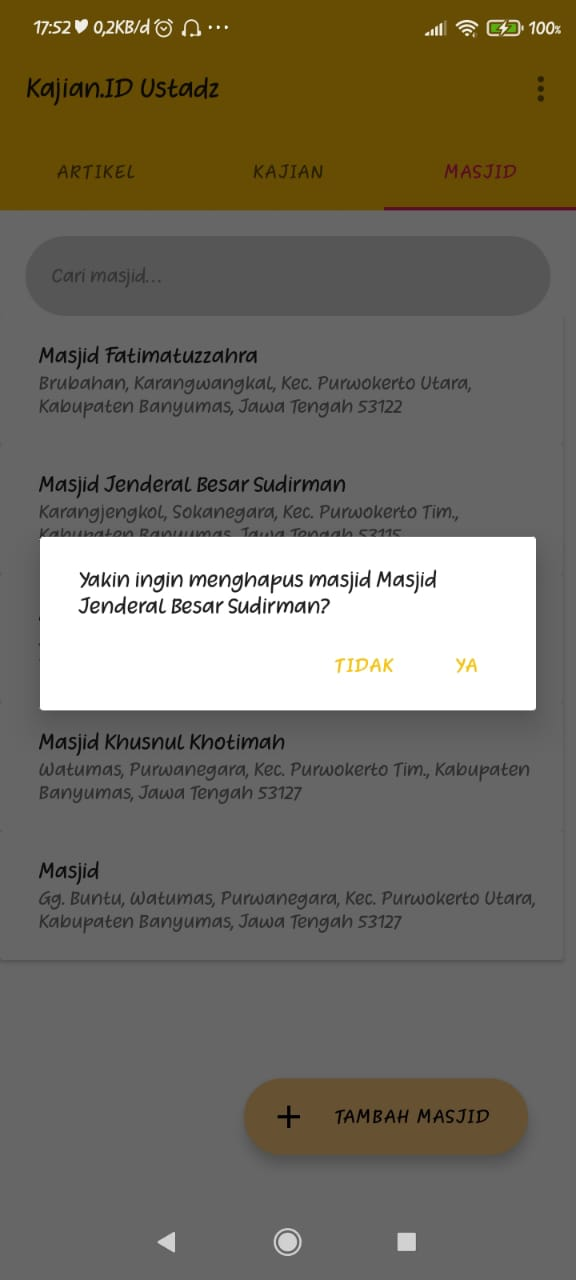


1. Cara menghapus masjid yang dikelola

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



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

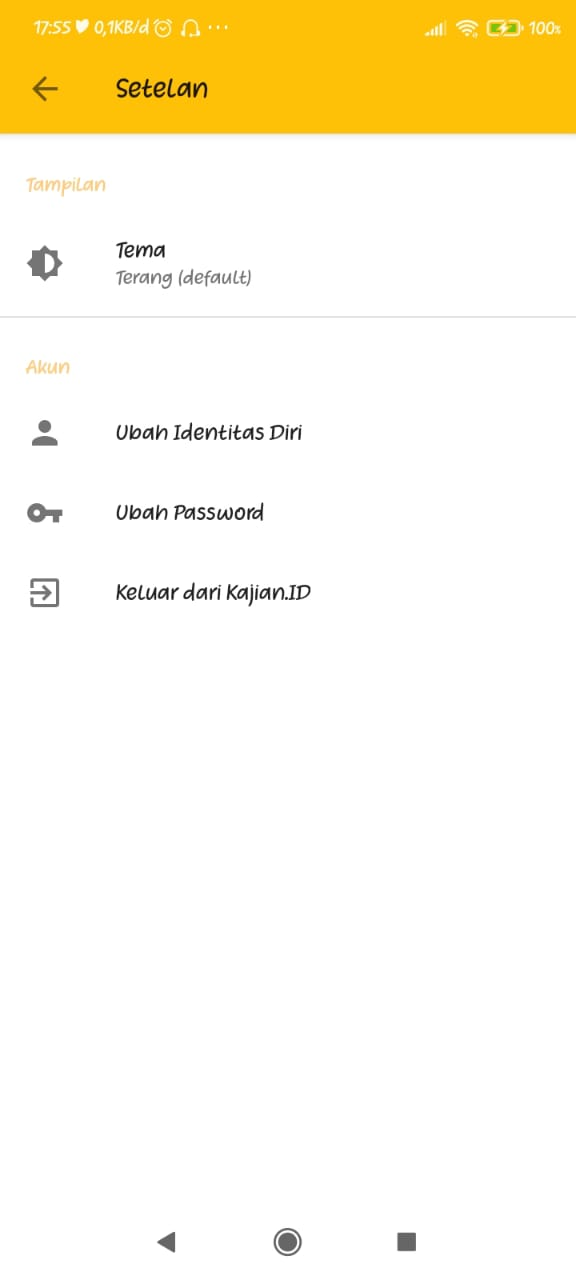


* Masjid telah dihapus. Perlu diingat, kajian yang pernah diadakan dan disimpan di masjid ini juga ikut terhapus. Namun, masjid tersebut tetap ada dalam server Kajian.ID.

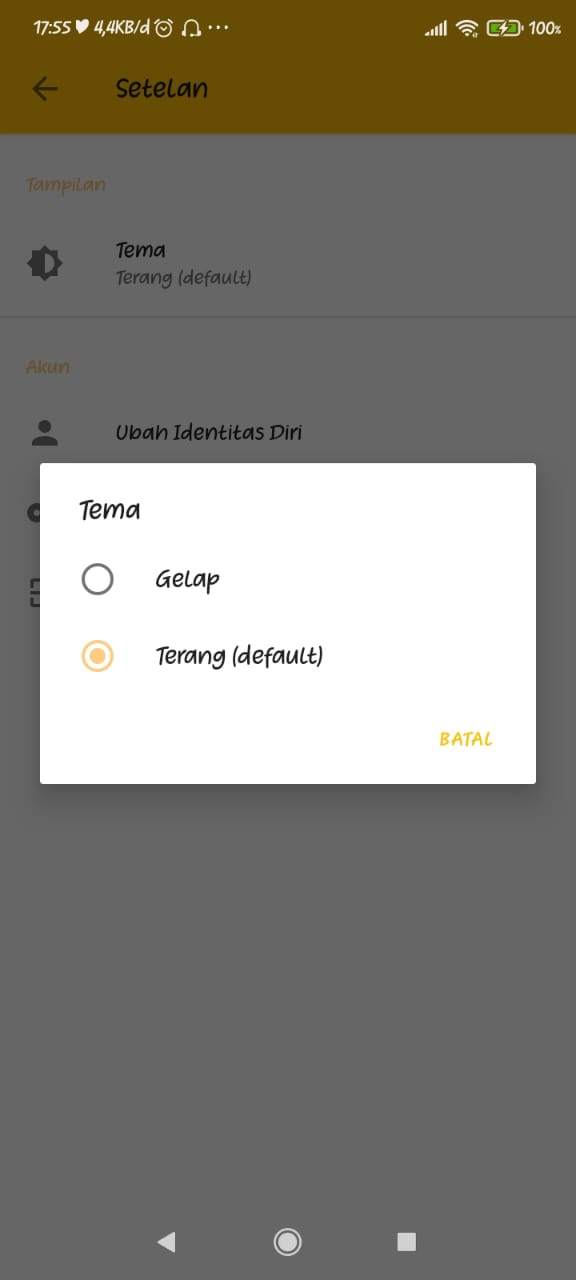
1. Cara mengubah tema

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”

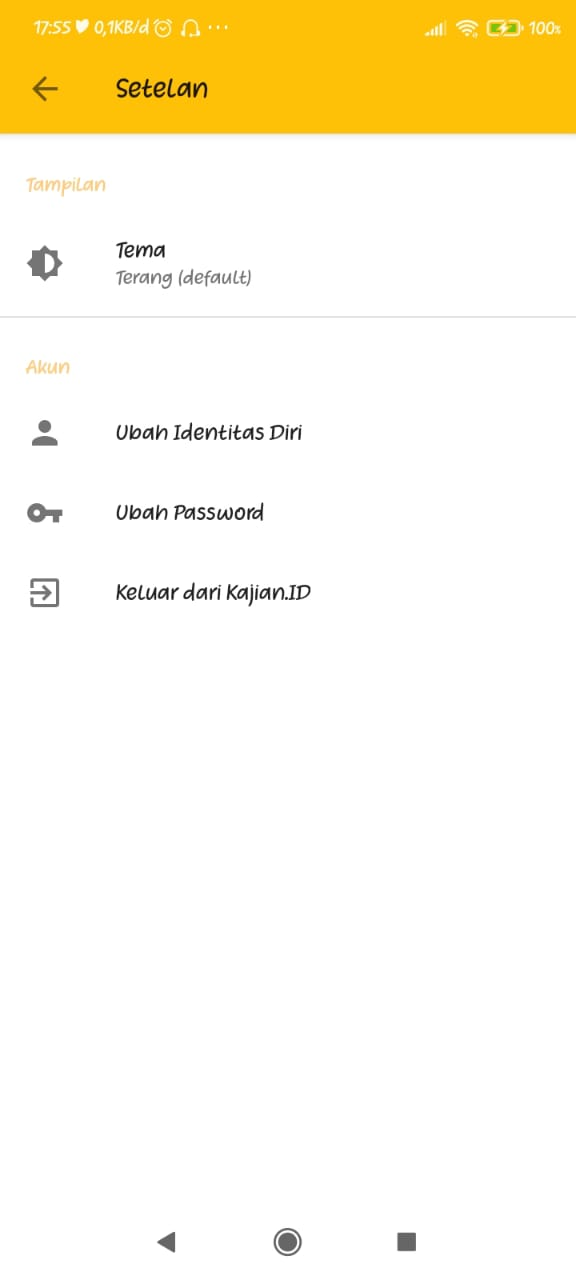


* Pilih “Tema” dan pilih tema mana yang ingin Anda terapkan di aplikasi Kajian.ID Ustadz.



1. Cara mengubah identitas diri

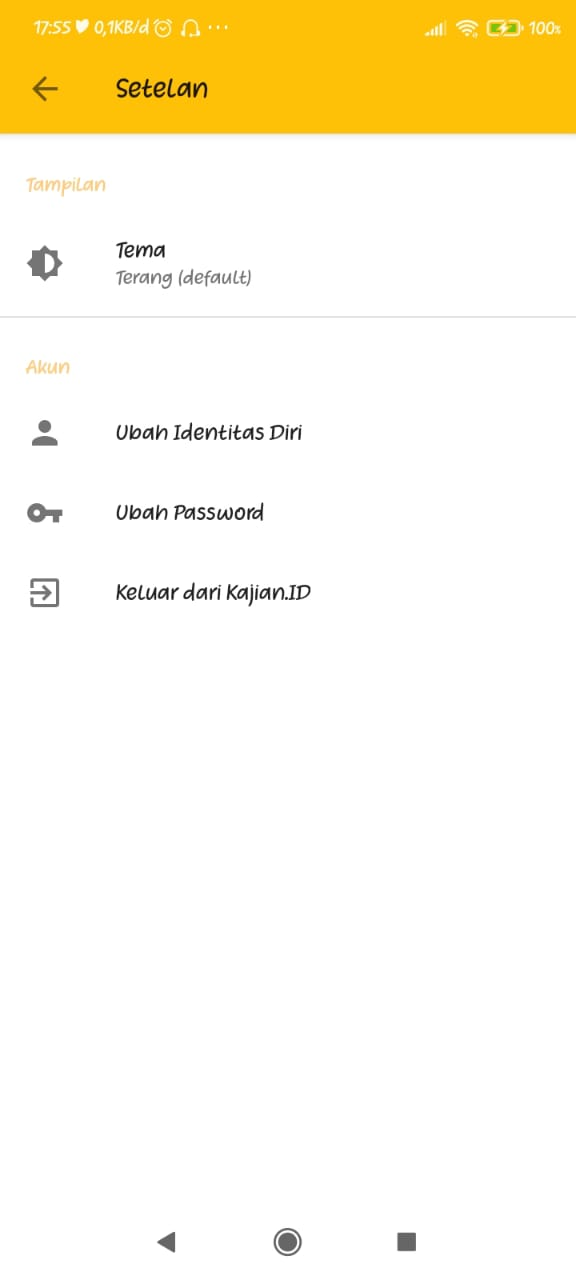
* Di layar utama, tekan tombol tiga titik ( ⋮ ) di pojok kanan atas layar dan pilih “Setelan”



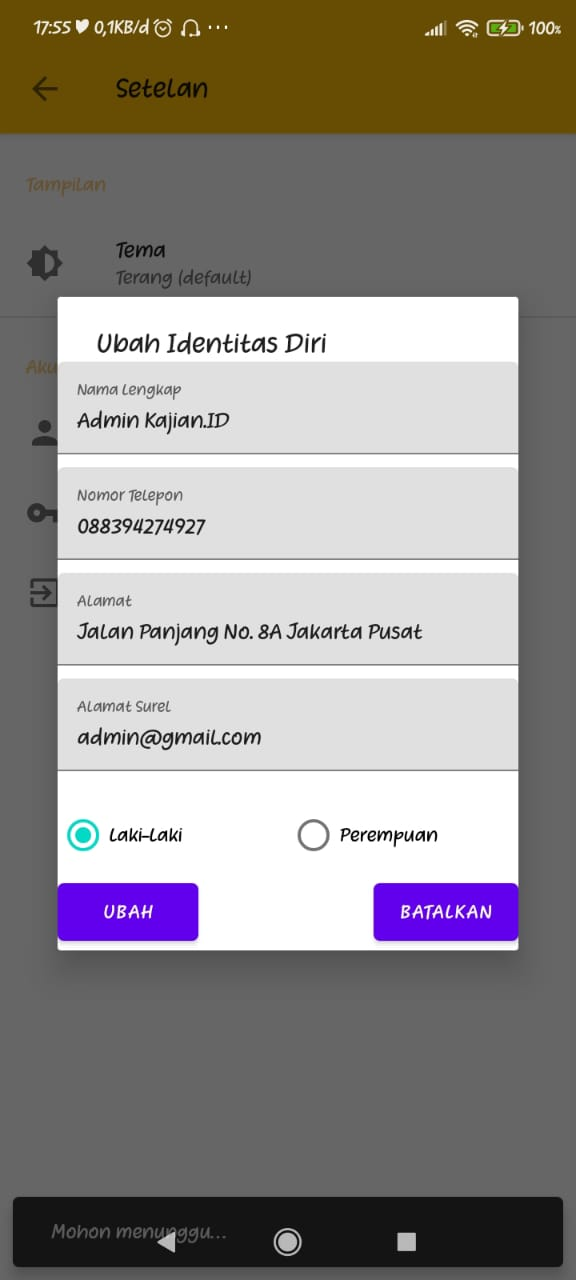
* Pilih “Ubah Identitas Diri” dan silahkan Anda ubah data diri Anda. Jika sudah, klik tombol “Ubah”.

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

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

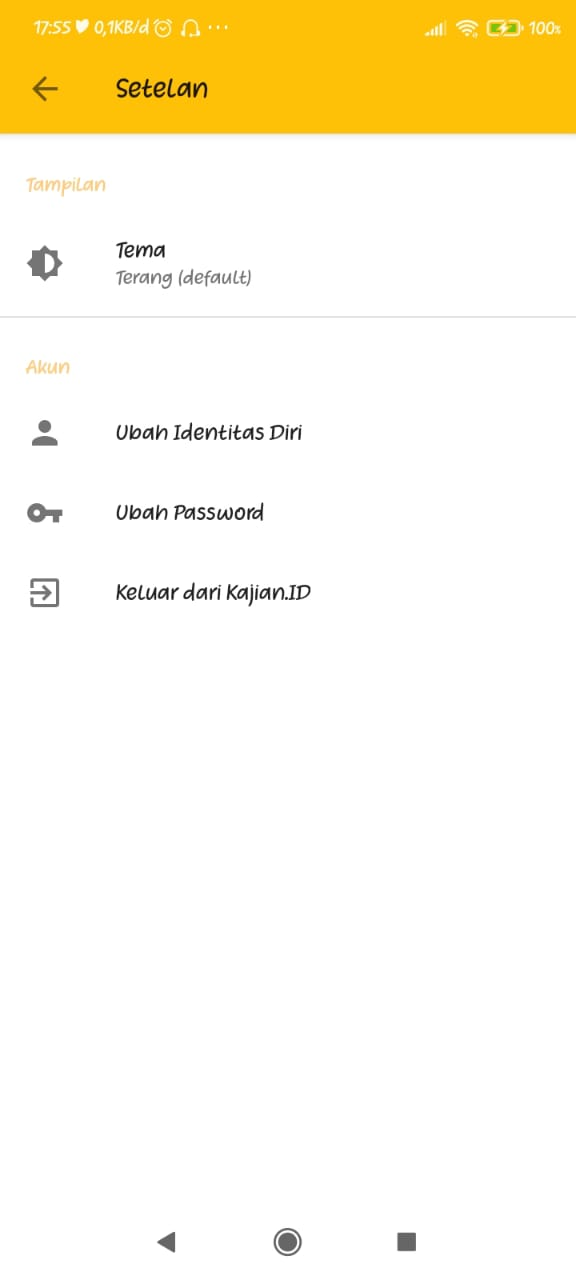


* Pilih “Ubah Password” dan silahkan ubah kata sandi Anda. Jika sudah, klik tombol “Ubah”.

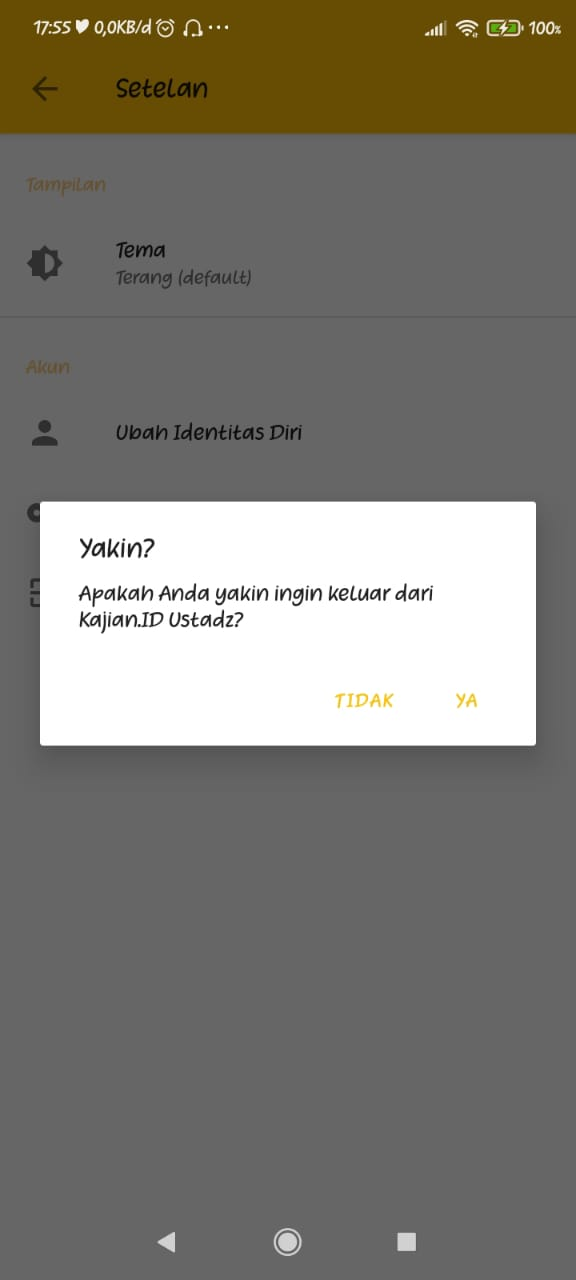


1. Cara mengeluarkan akun (*logout*) dari Kajian.ID Ustadz

* 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.



# BAB III: *SOURCE CODE*

## ALAMAT *REPOSITORY*

* Alamat repository untuk source code Kajian.ID User dan Kajian.ID Ustadz ada di <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>.

## KAJIAN.ID USER

* 1. Splash Screen

1. 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>

1. 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)  
 }  
}

* 1. Tampilan Utama

1. 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.DownloadedArticleFragment"  
 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>

1. 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;  
 }  
}

* 1. Menu Beranda

1. 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>

1. 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);  
 });  
 }  
}

* 1. Menu Artikel

1. 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.FloatingActionButton  
 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>

1. 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\_article))  
  
 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()));  
 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\_ARTICLES*, 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.ArticleColums.Companion.*ID*import com.kajianid.android.databases.DatabaseContract.ArticleColums.Companion.*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)  
 }  
  
 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.ArticleColums.ID))  
 val title = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.TITLE))  
 val post\_date = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.POST\_DATE))  
 val content = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.CONTENT))  
 val has\_img = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.HAS\_IMG))  
 val ustadz\_name = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.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.ArticleColums.ID))  
 val title = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.TITLE))  
 val post\_date = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.POST\_DATE))  
 val content = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.CONTENT))  
 val has\_img = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.HAS\_IMG))  
 val ustadz\_name = getString(getColumnIndexOrThrow(DatabaseContract.ArticleColums.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 ArticleColums : 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.ArticleColums.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.DESCRIPTION} 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.ArticleColums.ID} varchar(5) not null primary key,  
 ${DatabaseContract.ArticleColums.TITLE} text,  
 ${DatabaseContract.ArticleColums.POST\_DATE} Text,  
 ${DatabaseContract.ArticleColums.CONTENT} Text,  
 ${DatabaseContract.ArticleColums.HAS\_IMG} Int,  
 ${DatabaseContract.ArticleColums.USTADZ\_NAME} varchar(15),  
 ${DatabaseContract.ArticleColums.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 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.*server*) + "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.*server*) + "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"+responseBody+"\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\_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.ArticleColums.ID, article.id)  
 values.put(DatabaseContract.ArticleColums.TITLE, article.title)  
 values.put(DatabaseContract.ArticleColums.POST\_DATE, article.postDate)  
 values.put(DatabaseContract.ArticleColums.CONTENT, article.content)  
 values.put(DatabaseContract.ArticleColums.HAS\_IMG, article.hasImg)  
 values.put(DatabaseContract.ArticleColums.USTADZ\_NAME, article.ustadzName)  
 values.put(DatabaseContract.ArticleColums.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 disebarkan 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\_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\_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.ArticleColums.ID, article.*id*)  
 values.put(DatabaseContract.ArticleColums.TITLE, article.*title*)  
 values.put(DatabaseContract.ArticleColums.POST\_DATE, article.*postDate*)  
 values.put(DatabaseContract.ArticleColums.CONTENT, article.*content*)  
 values.put(DatabaseContract.ArticleColums.HAS\_IMG, article.*hasImg*)  
 values.put(DatabaseContract.ArticleColums.USTADZ\_NAME, article.*ustadzName*)  
 values.put(DatabaseContract.ArticleColums.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 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.FragmentDownloadedArticleBinding  
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()  
 }  
  
  
}

* 1. Menu Kajian

1. 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" />  
  
 </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.FloatingActionButton  
 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>

1. 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.getAddress(), "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.Companion.*ID*import com.kajianid.android.databases.DatabaseContract.KajianColumns.Companion.*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 ArticleColums : 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.ArticleColums.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.DESCRIPTION} 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.ArticleColums.ID} varchar(5) not null primary key,  
 ${DatabaseContract.ArticleColums.TITLE} text,  
 ${DatabaseContract.ArticleColums.POST\_DATE} Text,  
 ${DatabaseContract.ArticleColums.CONTENT} Text,  
 ${DatabaseContract.ArticleColums.HAS\_IMG} Int,  
 ${DatabaseContract.ArticleColums.USTADZ\_NAME} varchar(15),  
 ${DatabaseContract.ArticleColums.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 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("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"+responseBody+"\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 remindered = 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.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)  
 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){  
 remindered = false  
 binding.fabreminder.setImageResource(R.drawable.ic\_baseline\_notifications\_none\_24)  
 }else{  
 remindered = true  
 binding.fabreminder.setImageResource(R.drawable.ic\_baseline\_notifications\_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\_confirm))  
 alert.setPositiveButton(resources.getString(R.string.yes)) **{** \_, \_ **->** dbKajianHelper.deleteById(id)  
 remindered = 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.DESCRIPTION, 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)  
 remindered = true  
 binding.fabreminder.setImageResource(R.drawable.ic\_baseline\_notifications\_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 disebarkan 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 remindered = 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, makan 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\_notifications\_none\_24)  
 }else{  
 remindered = true  
 binding.fabreminder.setImageResource(R.drawable.ic\_baseline\_notifications\_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\_confirm))  
 alert.setPositiveButton(resources.getString(R.string.yes)) **{** \_, \_ **->** kajian.id?.let **{** it1 **->** dbKajianHelper.deleteById(it1) **}** remindered = false  
 binding.fabreminder.setImageResource(R.drawable.ic\_baseline\_notifications\_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.DESCRIPTION, 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)  
 remindered = true  
 binding.fabreminder.setImageResource(R.drawable.ic\_baseline\_notifications\_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 disebarkan 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 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*());  
 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 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;  
 }  
}

* 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 onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(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, "Pengingat 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  
 }  
 }  
}

* 1. Menu About

1. 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  
 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>

1. 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;  
 }  
  
 @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;  
 }  
}

* 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",  
 "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",  
 };  
  
}

* 1. 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>

* 1. 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>

* 1. 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>

* 1. 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>

* 1. 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 adispicing 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="l">?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="l">?attr/colorPrimary</item>  
 <!-- Customize your theme here. -->  
 </style>  
</resources>

* 1. 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.MapsResultActivity" />  
 <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>

* 1. 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.kotlinx: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()  
}

## KAJIAN.ID USTADZ

# BAB IV: KESIMPULAN DAN SARAN

## KESIMPULAN

## SARAN