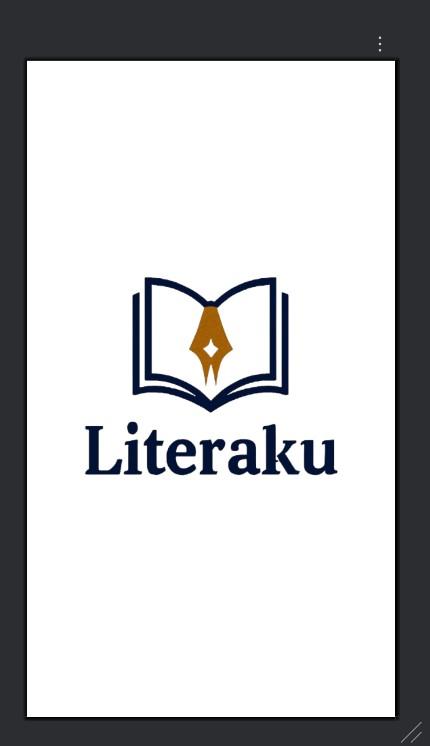
Nama: Muhammad Rizki Pratama

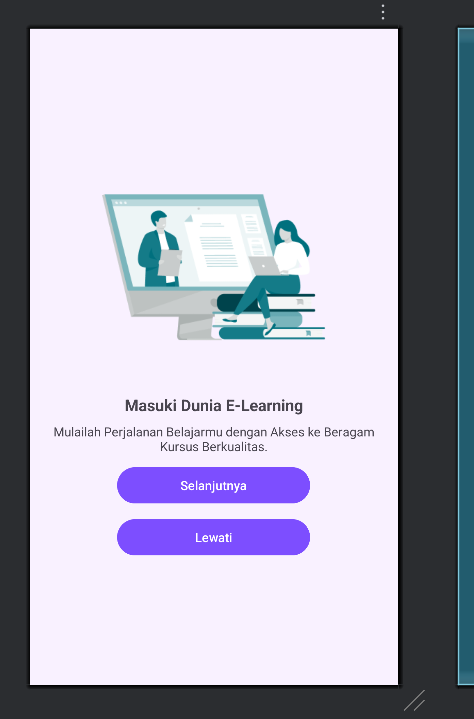
Npm : 23312191

Kelas: IF 23 FX

DOKUMENTASI

1. Tampilan awal ketika aplikasi pertama kali dijalankan. Lalu implementasi aktivitas splash screen yang akan tampil selama beberapa detik lalu berpindah ke halaman berikutnya
2. **SplashActivity** tampil 3 detik.
   1. Setelah itu menampilkan **FragmentIntro1**.
   2. Di FragmentIntro1:

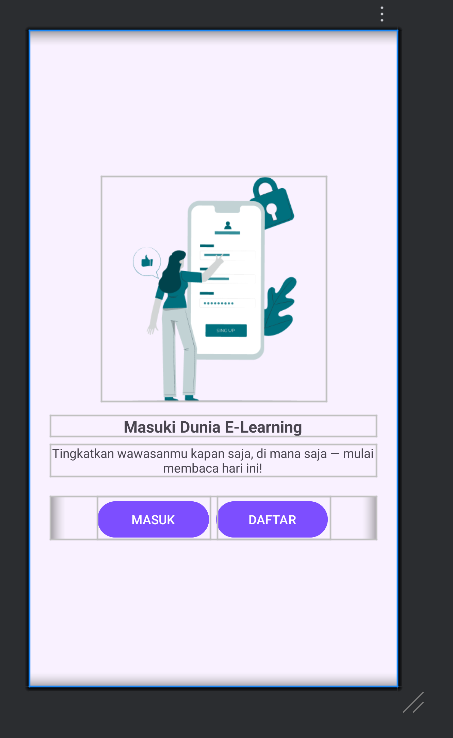
Tombol "Selanjutnya" → ke FragmentIntro2. Tombol "Lewati" → langsung ke FragmentUtama.



1. Tampilan ini adalah **halaman ketiga** dari rangkaian onboarding atau pengenalan aplikasi.

Dua Tombol Aksi:

* **MASUK** → Arahkan ke halaman login.
* **DAFTAR** → Arahkan ke halaman pendaftaran.



4. Saya membuat database untuk menampung id,name,email,password untuk bisa membuat fungsi pada tampilan signin dan signup.

package com.example.yourapp.database

import android.content.ContentValues

import android.content.Context

import android.database.Cursor

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper

data class User(

val id: Int,

val fullName: String,

val email: String,

val password: String

)

class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE\_NAME, null, DATABASE\_VERSION) {

companion object {

private const val DATABASE\_NAME = "userDB"

private const val DATABASE\_VERSION = 1

private const val TABLE\_USER = "user"

private const val COLUMN\_ID = "id"

private const val COLUMN\_NAME = "full\_name"

private const val COLUMN\_EMAIL = "email"

private const val COLUMN\_PASSWORD = "password"

}

override fun onCreate(db: SQLiteDatabase?) {

// Create the user table

val CREATE\_USER\_TABLE = "CREATE TABLE $TABLE\_USER (" +

"$COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT," +

"$COLUMN\_NAME TEXT," +

"$COLUMN\_EMAIL TEXT UNIQUE," +

"$COLUMN\_PASSWORD TEXT)"

db?.execSQL(CREATE\_USER\_TABLE)

}

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

// Drop older table if exists

db?.execSQL("DROP TABLE IF EXISTS $TABLE\_USER")

// Create new table

onCreate(db)

}

// Method to insert user

fun insertUser(fullName: String, email: String, password: String): Long {

val db = *writableDatabase*

val values = ContentValues().*apply* **{**

put(COLUMN\_NAME, fullName)

put(COLUMN\_EMAIL, email)

put(COLUMN\_PASSWORD, password)

**}**

// Insert the data into the database and return the row ID

return db.insert(TABLE\_USER, null, values).*also* **{**

db.close() // Ensure the database is closed after the insert

**}**

}

// Method to check if the user exists and validate credentials

fun isValidUser(email: String, password: String): Boolean {

val db = *readableDatabase*

val query = "SELECT \* FROM $TABLE\_USER WHERE $COLUMN\_EMAIL = ? AND $COLUMN\_PASSWORD = ?"

val cursor: Cursor = db.rawQuery(query, *arrayOf*(email, password))

val isValid = cursor.*count* > 0

cursor.close() // Always close the cursor

db.close() // Always close the database

return isValid

}

// Method to get user data by email and password

fun getUserData(email: String, password: String): User? {

val db = *readableDatabase*

val query = "SELECT \* FROM $TABLE\_USER WHERE $COLUMN\_EMAIL = ? AND $COLUMN\_PASSWORD = ?"

val cursor: Cursor = db.rawQuery(query, *arrayOf*(email, password))

var user: User? = null

if (cursor.moveToFirst()) {

user = User(

id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_ID)),

fullName = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME)),

email = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_PASSWORD))

)

}

cursor.close()

db.close()

return user

}

// Method to get user data by ID

fun getUserById(userId: Int): User? {

val db = *readableDatabase*

val query = "SELECT \* FROM $TABLE\_USER WHERE $COLUMN\_ID = ?"

val cursor: Cursor = db.rawQuery(query, *arrayOf*(userId.toString()))

var user: User? = null

if (cursor.moveToFirst()) {

user = User(

id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_ID)),

fullName = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME)),

email = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_PASSWORD))

)

}

cursor.close()

db.close()

return user

}

// Method to update user data

fun updateUser(userId: Int, fullName: String, email: String, password: String): Boolean {

val db = *writableDatabase*

val values = ContentValues().*apply* **{**

put(COLUMN\_NAME, fullName)

put(COLUMN\_EMAIL, email)

put(COLUMN\_PASSWORD, password)

**}**

val result = db.update(TABLE\_USER, values, "$COLUMN\_ID = ?", *arrayOf*(userId.toString()))

db.close()

return result > 0

}

// Method to check if email already exists (excluding current user)

fun isEmailExistsForOtherUser(email: String, currentUserId: Int): Boolean {

val db = *readableDatabase*

val query = "SELECT \* FROM $TABLE\_USER WHERE $COLUMN\_EMAIL = ? AND $COLUMN\_ID != ?"

val cursor: Cursor = db.rawQuery(query, *arrayOf*(email, currentUserId.toString()))

val exists = cursor.*count* > 0

cursor.close()

db.close()

return exists

}

// Method to check if email already exists

fun isEmailExists(email: String): Boolean {

val db = *readableDatabase*

val query = "SELECT \* FROM $TABLE\_USER WHERE $COLUMN\_EMAIL = ?"

val cursor: Cursor = db.rawQuery(query, *arrayOf*(email))

val exists = cursor.*count* > 0

cursor.close()

db.close()

return exists

}

// Method to verify current password

fun verifyCurrentPassword(userId: Int, currentPassword: String): Boolean {

val db = *readableDatabase*

val query = "SELECT \* FROM $TABLE\_USER WHERE $COLUMN\_ID = ? AND $COLUMN\_PASSWORD = ?"

val cursor: Cursor = db.rawQuery(query, *arrayOf*(userId.toString(), currentPassword))

val isValid = cursor.*count* > 0

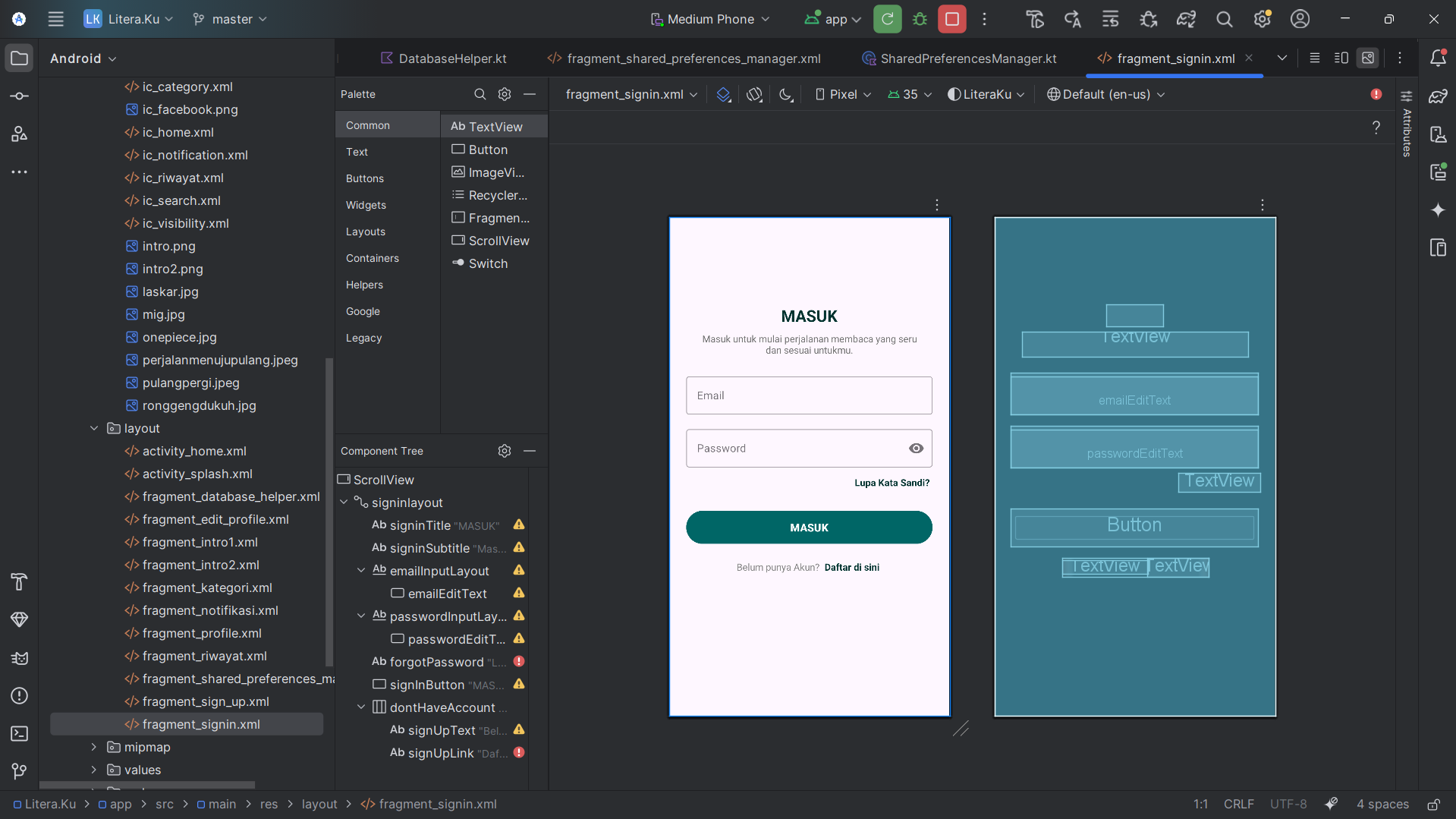
cursor.close()

db.close()

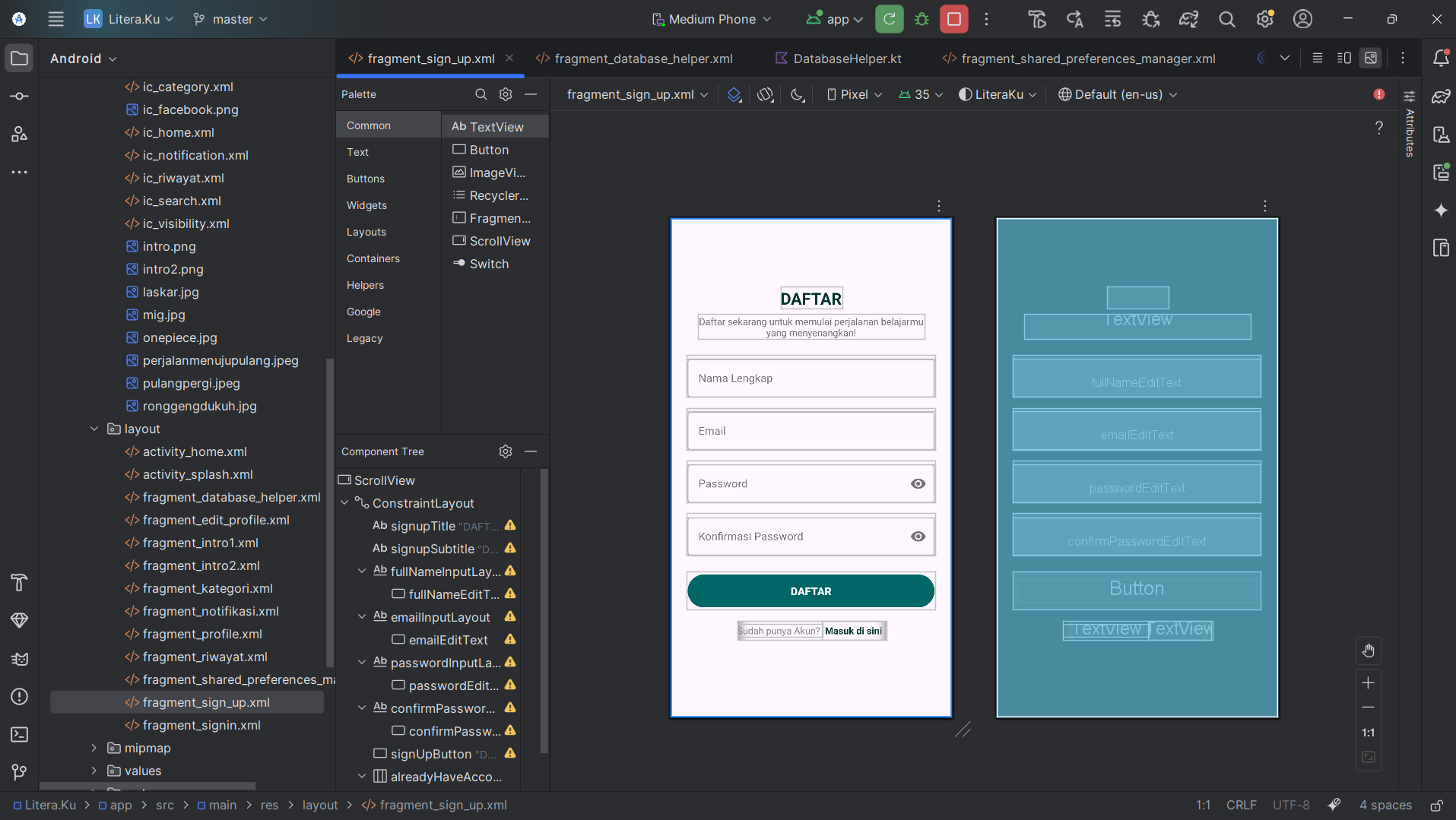
return isValid

}

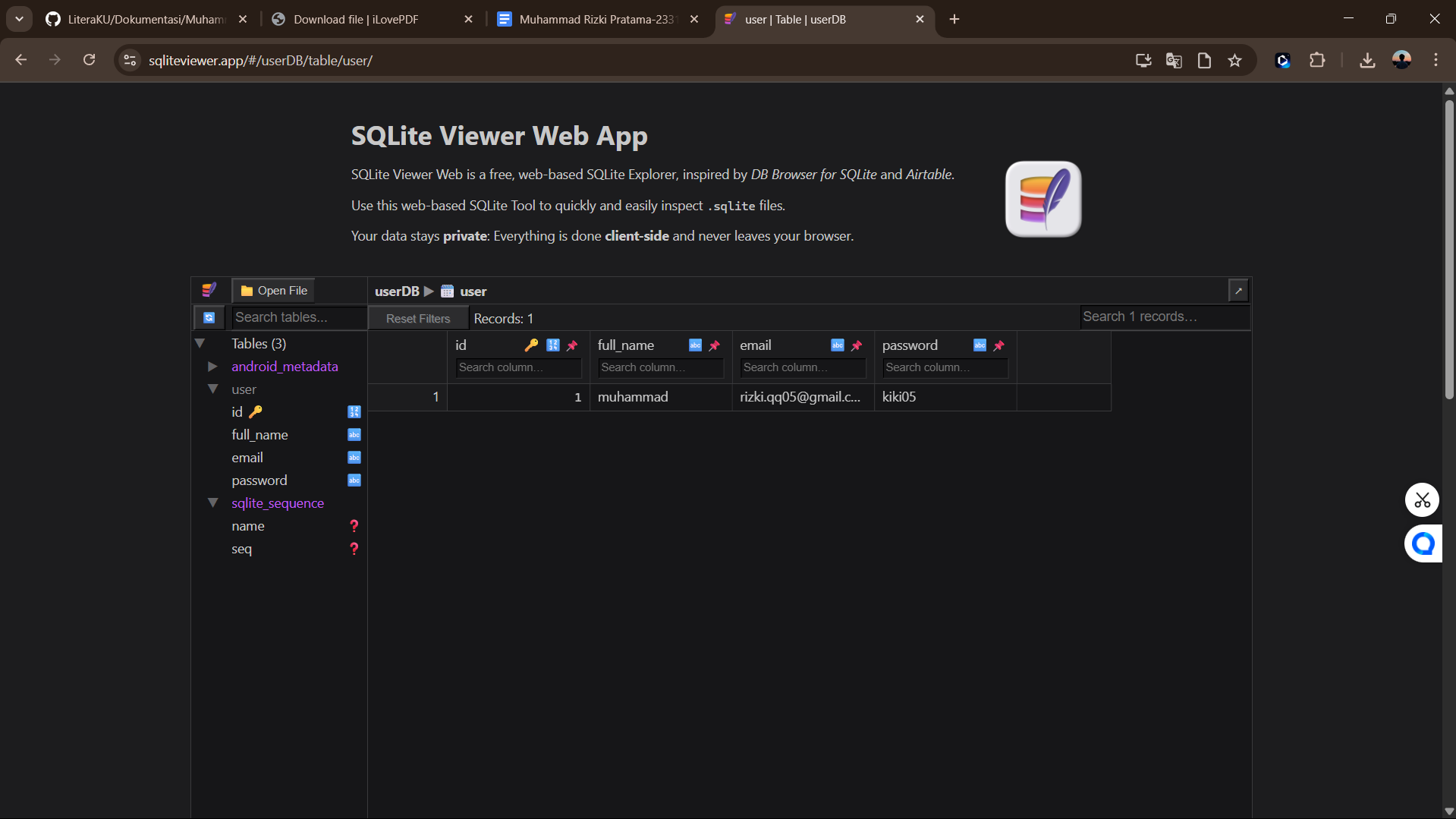
}

5. seperti ini tampilan halaman jika kita memasukan email,dan password maka akan masuk ketampilan utama

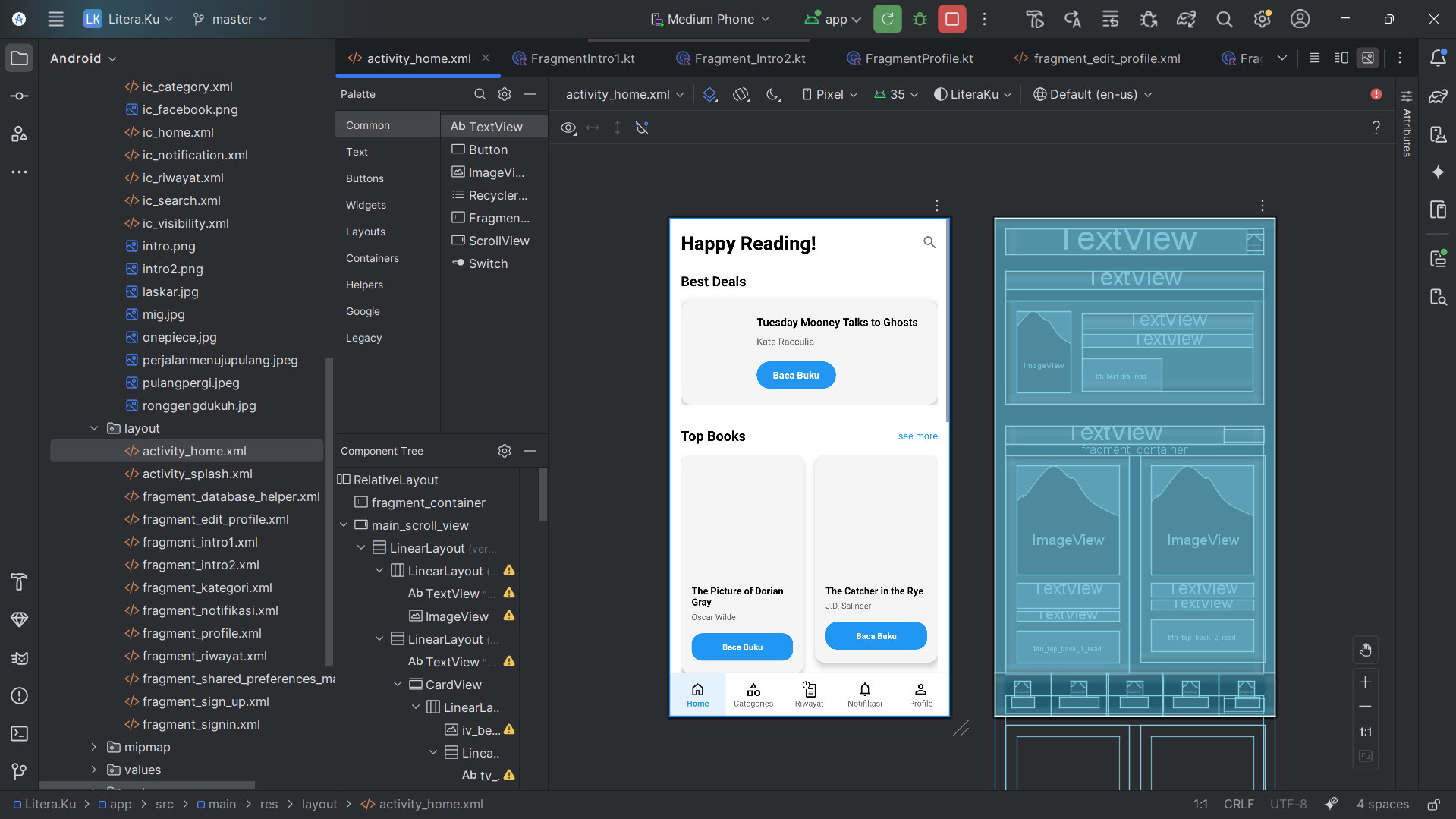
6.ini tampilan menu daftar/signup jika kita memasukan data nama,email,password maka akan data tersebut akan masuk ke data sqlite kita



7.ini hasil tampilan isi database yang menampung data login akun kita



8. ini tampilan awal saat kita sudah login masuk pertama kali dan ini masih polos dikarenakan blum dimasukan data buku agar kita bisa membacany dan ini kalo kita geser kebawah itu sudah bisa



9. saya membuat layout untuk bagian edit profile serta membuat fungsi agar kita bisa mengupdate data pribadi kita

