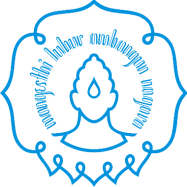
**PRATIK APLIKASI MOBILE**

“Sync dan Async Process & Sqflite”

Modul Praktikum 9

****

**Disusun oleh :**

Diah Munica Nawang

V3922015 / TI D

**Dosen :**

Trisna Ari Roshinta, S.S.T., M.T

**PS D-III TEKNIK INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**2023**

**Langkah Praktikum**

**1. Implementasi Sync Process**

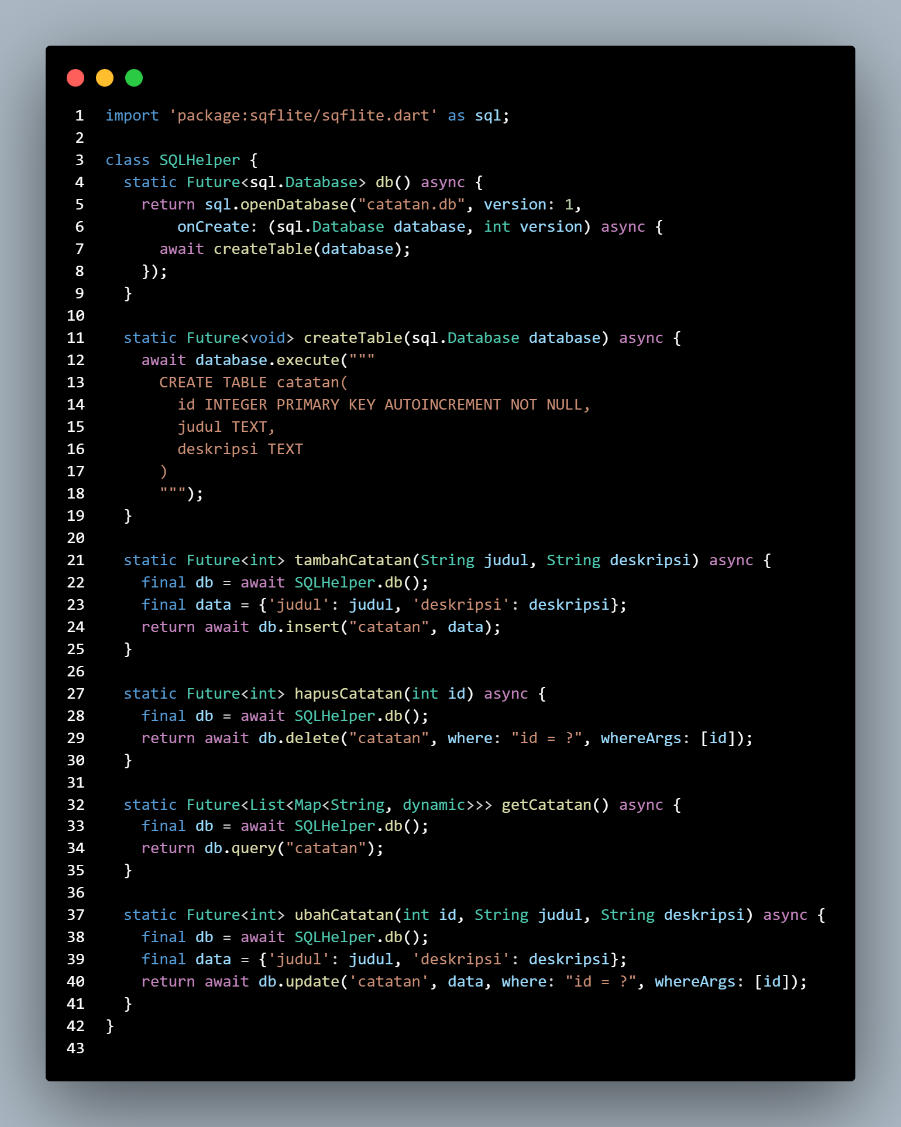
**Deskripsi**

Pada langkah ini, saya akan membuat fungsi untuk mengakses database SQLite secara sinkron dan melakukan operasi CRUD sederhana.

**Langkah-langkah**

1. Inisialisasi database SQLite.
2. Tambahkan data ke database.
3. Ambil data dari database.
4. Update data dalam database.
5. Hapus data dari database.

**Script :**

****

**Penjelasan Script :**

|  |
| --- |
| **Import Library**:  dart   import 'package:sqflite/sqflite.dart' as sql;  Baris ini mengimpor library sqflite dan memberikannya alias sql untuk digunakan dalam script. Library ini digunakan untuk mengakses dan mengelola database SQLite.   **Class SQLHelper**:  Ini adalah definisi kelas utama yang digunakan untuk mengelola akses ke database SQLite. Kelas ini memiliki beberapa metode yang memfasilitasi operasi CRUD (Create, Read, Update, Delete) pada tabel "catatan" dalam database SQLite.   **Static Method db()**:   * Metode ini digunakan untuk membuka atau membuat database SQLite. * Itu mengembalikan objek Future<sql.Database>, yang akan berisi koneksi ke database. * Jika database "catatan.db" belum ada, maka metode onCreate akan dipanggil untuk membuat tabel "catatan".    **Static Method createTable()**:   * Metode ini digunakan untuk membuat tabel "catatan" jika belum ada. * Ini menggunakan perintah SQL CREATE TABLE untuk mendefinisikan skema tabel.    **Static Method tambahCatatan()**:   * Metode ini digunakan untuk menambahkan catatan baru ke tabel "catatan". * Menggunakan objek db untuk melakukan operasi insert yang mengembalikan ID dari catatan yang ditambahkan.    **Static Method hapusCatatan()**:   * Metode ini digunakan untuk menghapus catatan dari tabel "catatan" berdasarkan ID. * Menggunakan objek db untuk melakukan operasi delete yang mengembalikan jumlah catatan yang dihapus.    **Static Method getCatatan()**:   * Metode ini digunakan untuk mengambil seluruh catatan dari tabel "catatan". * Menggunakan objek db untuk melakukan operasi query yang mengembalikan daftar Map<String, dynamic> yang berisi catatan-catatan.    **Static Method ubahCatatan()**:   * Metode ini digunakan untuk mengubah catatan dalam tabel "catatan" berdasarkan ID. * Menggunakan objek db untuk melakukan operasi update yang mengembalikan jumlah catatan yang diubah. |

**2. Implementasi Async Process**

**Deskripsi**

Pada langkah ini, saya membuat fungsi untuk mengakses database SQLite secara asinkron menggunakan async dan await.

**Langkah-langkah**

1. Inisialisasi database SQLite secara asinkron.
2. Tambahkan data ke database dengan operasi asinkron.
3. Ambil data dari database dengan operasi asinkron.
4. Update data dalam database dengan operasi asinkron.
5. Hapus data dari database dengan operasi asinkron.

**Script :**

****

****

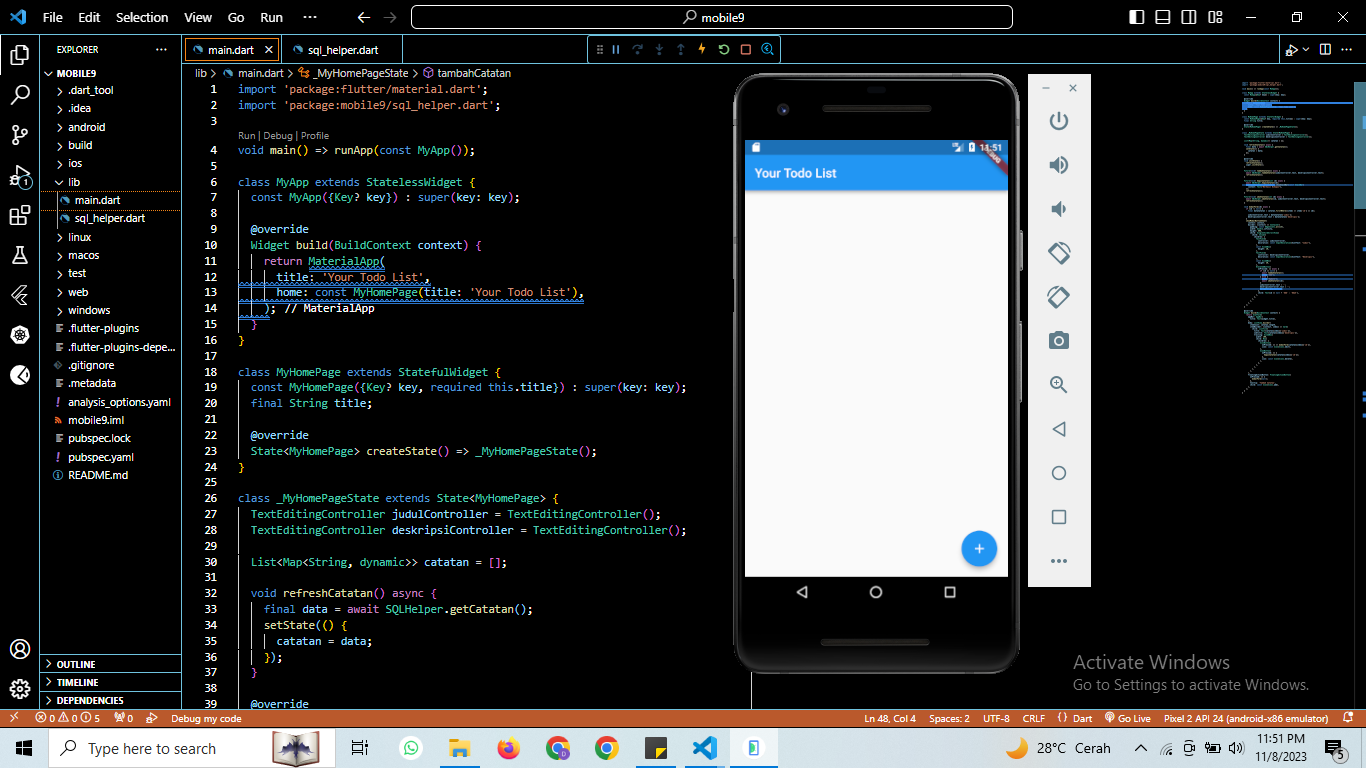
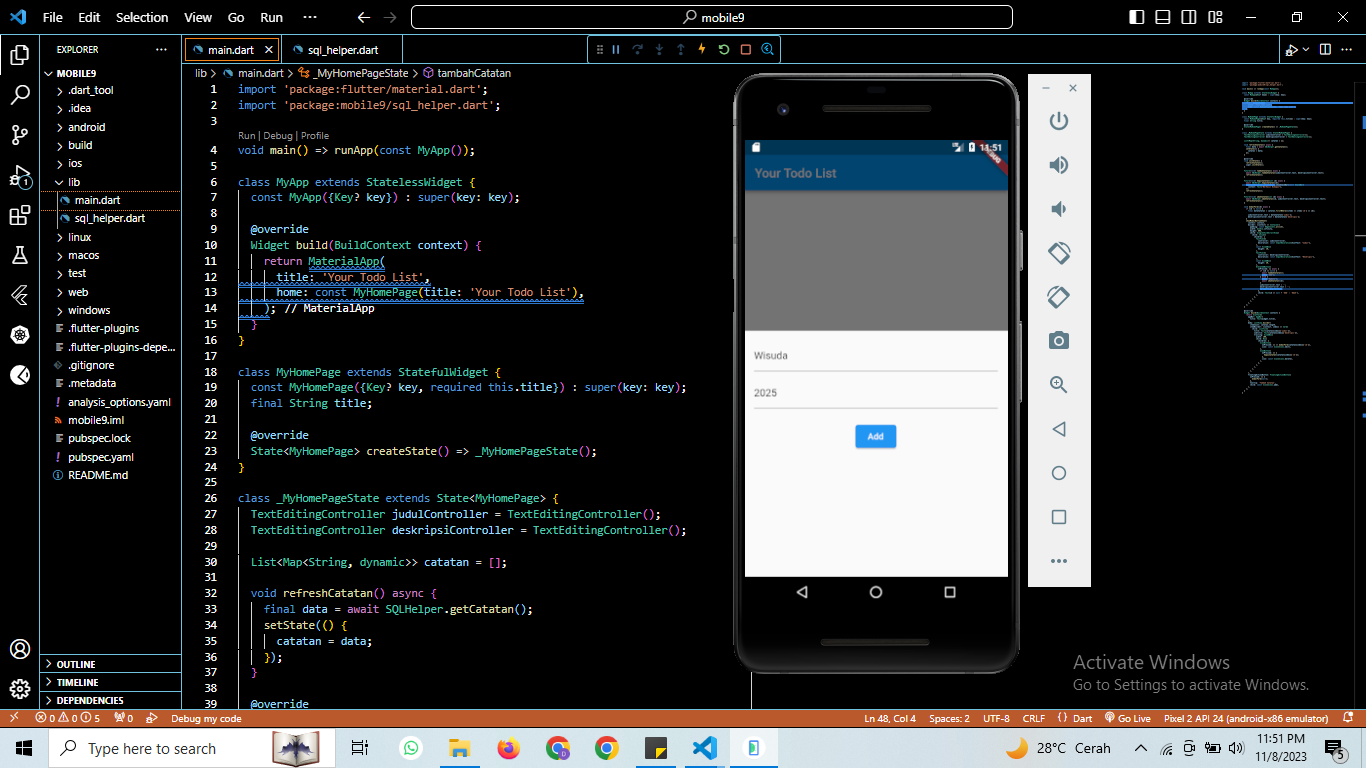
****

**Penjelasan Script :**

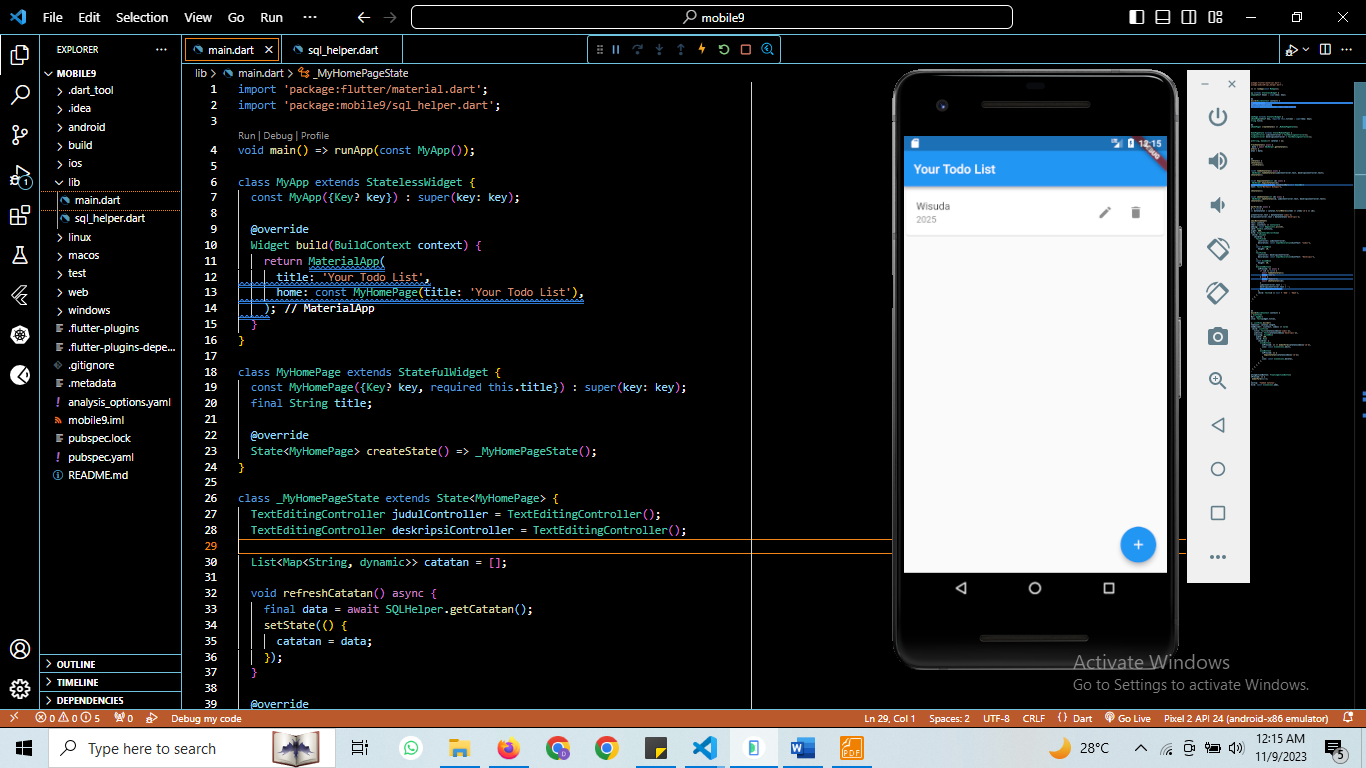
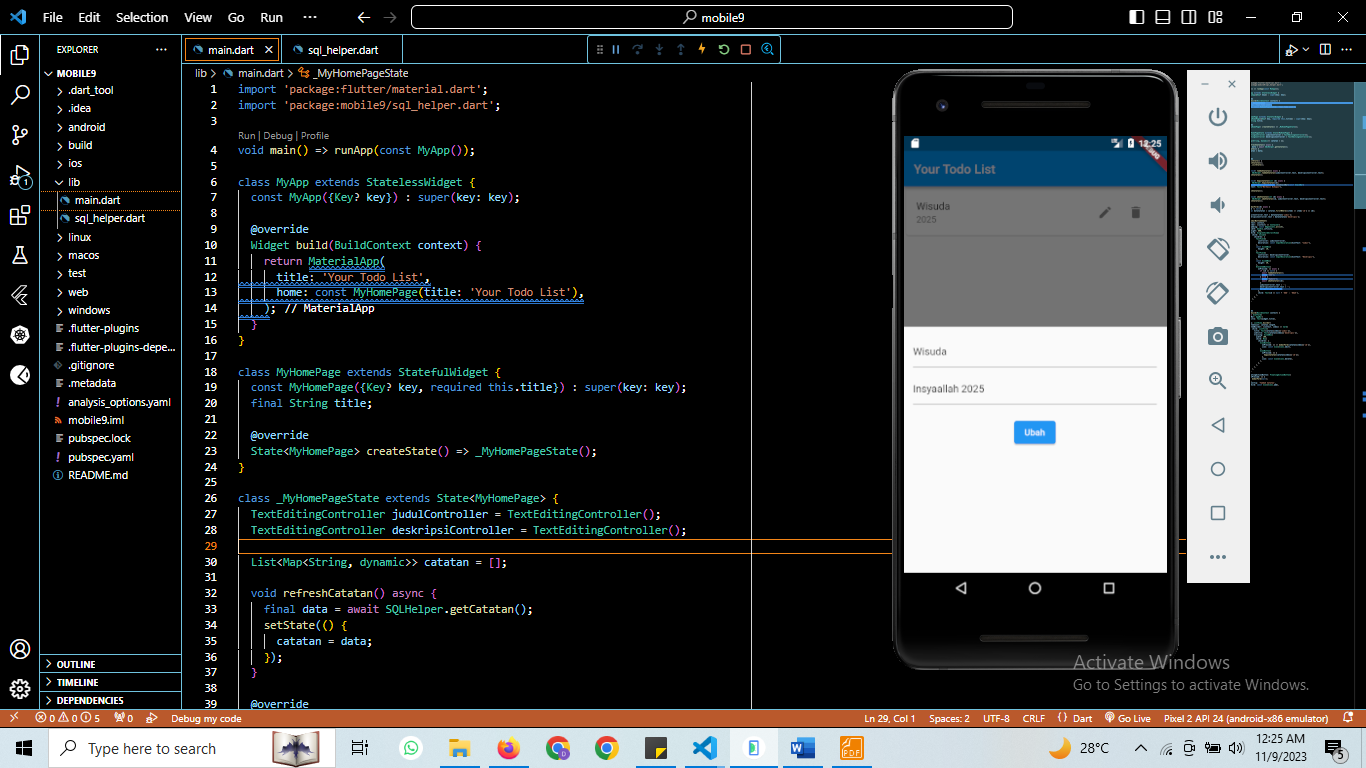
|  |
| --- |
| 1. **Import Libraries**:    * Script ini mengimpor dua library utama, yaitu flutter/material.dart dan mobile9/sql\_helper.dart. Library flutter/material.dart digunakan untuk membuat tampilan antarmuka pengguna (UI), sementara mobile9/sql\_helper.dart adalah library yang berisi kode untuk berinteraksi dengan database SQLite. 2. **Main Function**:    * Fungsi main merupakan fungsi utama dalam aplikasi Flutter.    * Fungsi ini memanggil MyApp, yang merupakan widget utama aplikasi, dan memulai aplikasi. 3. **MyApp Class**:    * Kelas MyApp adalah widget utama yang mewakili seluruh aplikasi.    * Kelas ini mengonfigurasi dan menginisialisasi aplikasi Flutter.    * MyApp menggunakan MaterialApp sebagai root widget untuk menentukan judul aplikasi dan halaman awalnya. 4. **MyHomePage Class**:    * Kelas MyHomePage adalah widget yang mewakili halaman utama aplikasi.    * Kelas ini menerima properti title yang digunakan untuk menampilkan judul halaman. 5. **\_MyHomePageState Class**:    * Kelas \_MyHomePageState adalah state dari MyHomePage.    * Ini mendefinisikan elemen-elemen UI seperti TextField, ListView.builder, dan FloatingActionButton. 6. **Controller Text Fields**:    * judulController dan deskripsiController digunakan untuk mengontrol input teks pada judul dan deskripsi catatan. 7. **List catatan**:    * Variabel catatan adalah daftar catatan yang akan ditampilkan dalam aplikasi.    * Ini diperbarui setiap kali ada perubahan data dalam database SQLite. 8. **refreshCatatan() Method**:    * Metode ini digunakan untuk mengambil data catatan dari database SQLite dan memperbarui tampilan aplikasi.    * Ini dipanggil ketika aplikasi dimulai dan setelah operasi tambah, ubah, atau hapus catatan. 9. **initState() Method**:    * Metode ini digunakan untuk menginisialisasi catatan ketika aplikasi pertama kali dimulai. 10. **tambahCatatan() Method**:     * Metode ini digunakan untuk menambahkan catatan baru ke database SQLite.     * Setelah penambahan, data di-refresh dan field input dikosongkan. 11. **hapusCatatan() Method**:     * Metode ini digunakan untuk menghapus catatan dari database SQLite berdasarkan ID.     * Ini juga menampilkan pesan notifikasi menggunakan ScaffoldMessenger. 12. **ubahCatatan() Method**:     * Metode ini digunakan untuk mengubah catatan yang ada dalam database SQLite berdasarkan ID.     * Setelah perubahan, data di-refresh. 13. **modalForm() Method**:     * Metode ini menampilkan modal bottom sheet yang memungkinkan pengguna untuk menambah atau mengubah catatan.     * Jika id tidak null, itu berarti operasi ubah, dan data catatan diisi ke dalam input fields.     * Modal bottom sheet berisi input fields dan tombol "Add" atau "Ubah". 14. **build() Method**:     * Metode ini digunakan untuk membangun tampilan utama aplikasi.     * Ini berisi AppBar dengan judul, daftar catatan yang ditampilkan dalam ListView.builder, dan FloatingActionButton untuk menambah catatan baru. |

**3. Screenshots**

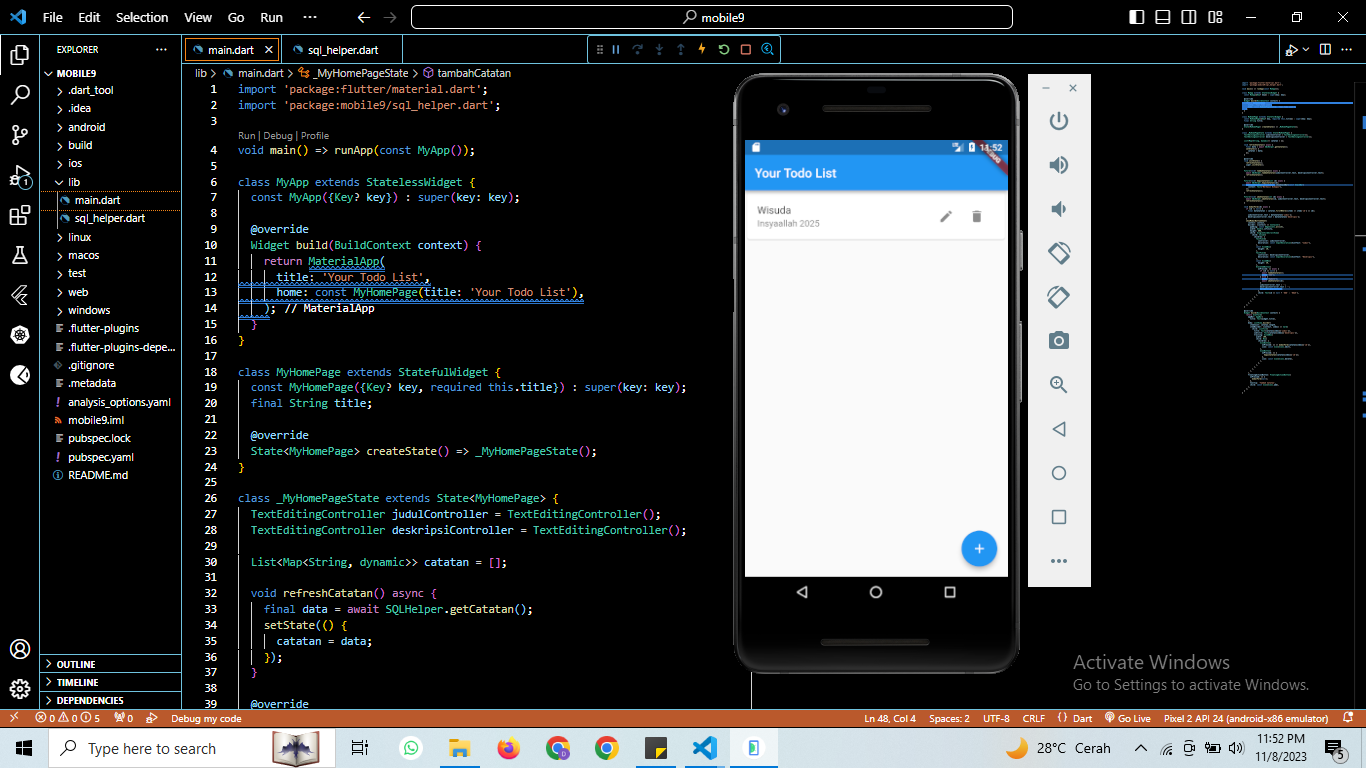
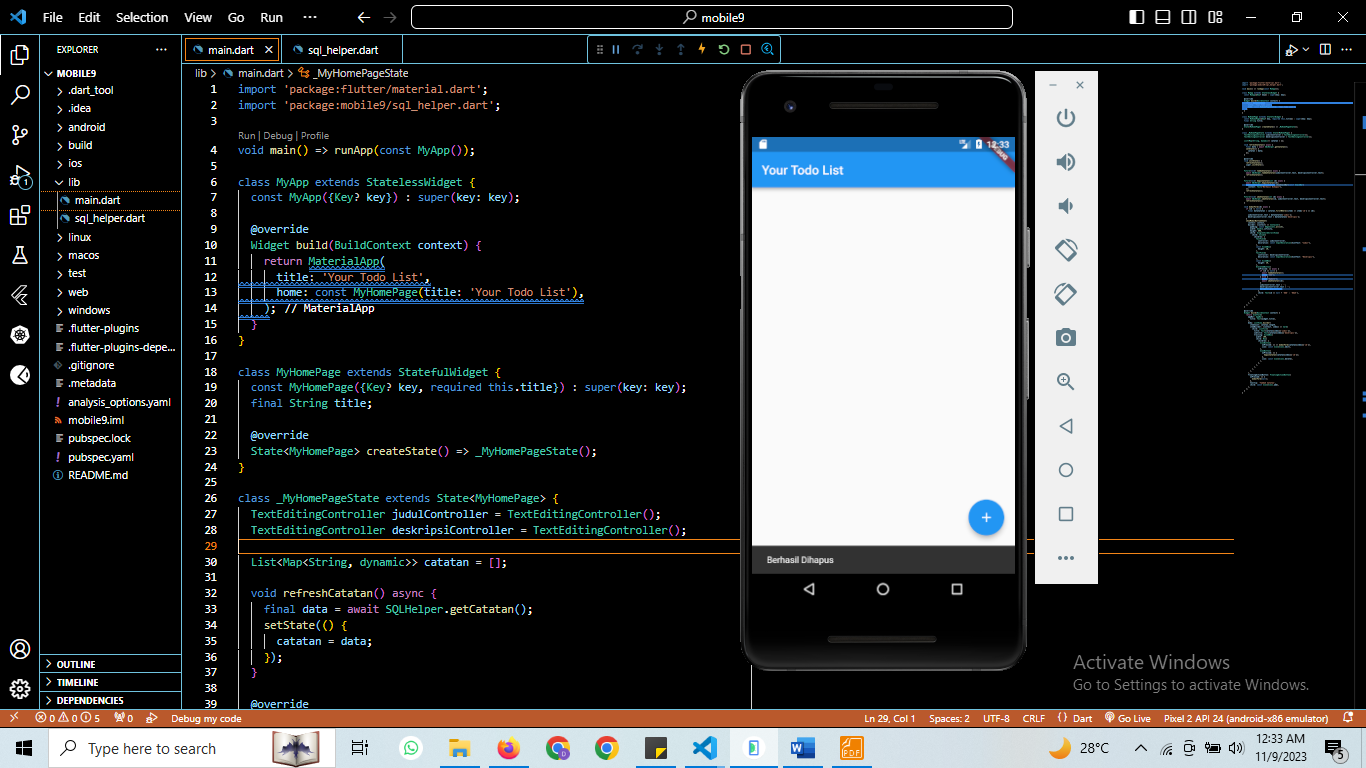
*Tampilan Awal : Create ToDo :*

********

*ToDo Berhasil Dibuat : Edit ToDo :*

*ToDo Berhasil diedit : Hapus ToDo Berhasil :*

**** 

**Hasil dan Kesimpulan**

Dengan demikian, praktikum ini memberikan dasar yang kuat untuk memahami konsep sinkronisasi, asinkronisasi, dan interaksi dengan database dalam pengembangan aplikasi mobile menggunakan Flutter.