

PRATIK APLIKASI MOBILE

“Sqflite Upload File”
Modul Praktikum 11



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**

Laporan Proyek: Aplikasi "*Sqflite Upload File*" dengan Flutter

I. Pendahuluan

1.1 Latar Belakang

Pada praktikum ini, kami mengimplementasikan fitur upload file pada aplikasi mobile menggunakan SQFLITE sebagai basis data lokal. Upload file menjadi salah satu fungsi penting dalam pengembangan aplikasi mobile untuk menyimpan dan mengelola berbagai jenis data.

1.2 Tujuan

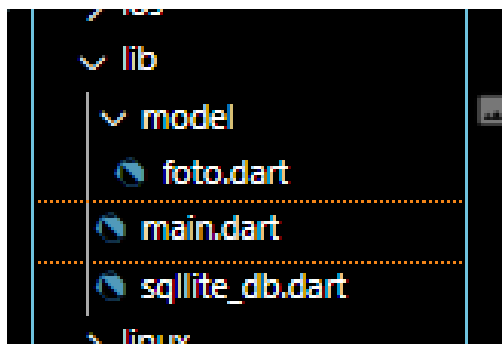
Tujuan dari praktikum ini adalah mengimplementasikan dan memahami proses upload file menggunakan SQFLITE serta melibatkan interaksi dengan basis data lokal pada aplikasi mobile.

II. Implementasi Dasar

2.1 Struktur Folder

Struktur folder proyek diatur sebagai berikut:

```
lib/  
|-- main.dart  
|-- sqllite_db.dart  
|-- model/  
|   |-- foto.dart  
--
```



→ Implementasi

2.2 Deskripsi File Utama

2.2.1 `main.dart`

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_application_11/sqlite_db.dart';
import 'dart:async';
import 'package:image_picker/image_picker.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget { //mendefinisikan class MyApp turunan dari StatelessWidget
  const MyApp({Key? key}) : super(key: key); //konstruktor untuk kelas myApp

  @override
  Widget build(BuildContext context) { // Mendefinisikan fungsi build untuk mengembalikan widget
    return MaterialApp(
      title: 'List Gambar Diah Munica N', // judul aplikasi
      home: const MyHomePage(title: 'List Gambar Diah Munica N'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);
  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  TextEditingController judulController = TextEditingController(); // Mendeklarasikan controller untuk judul
  TextEditingController deskripsiController = TextEditingController(); // Mendeklarasikan controller untuk deskripsi
  TextEditingController gambarController = TextEditingController(); // Mendeklarasikan controller untuk gambar

  List<Map<String, dynamic>> catatan = []; // Mendeklarasikan list kosong untuk catatan
  Map<String, dynamic>? catatanDihapus; // Mendeklarasikan variabel untuk menyimpan catatan yang dihapus

  void refreshCatatan() async { // Fungsi untuk memperbarui daftar catatan dari database
    final data = await SQLHelper.getCatatan(); // Mendapatkan data catatan dari database
    setState(() { // Memperbarui state dari widget
      catatan = data; // Menyimpan data yang diperoleh ke dalam list catatan
    });
  }

  @override
  void initState() { // Metode yang dipanggil saat state pertama kali dibuat
    refreshCatatan(); // Memanggil fungsi untuk memperbarui catatan
    super.initState();
  }

  Future<void> _ambilGambar() async { // Fungsi untuk mengambil gambar dari galeri
    final picker = ImagePicker(); // Objek ImagePicker
```

```
final pickedFile = await picker.pickImage(source: ImageSource.gallery); // Mengambil gambar dari galeri
```

```
if (pickedFile != null) {  
  setState(() {  
    gambarController.text = pickedFile.path; // Mengatur path gambar yang dipilih ke dalam controller gambar  
  });  
}
```

// Fungsi-fungsi untuk menambah, menghapus, mengubah, dan menampilkan catatan serta tampilan aplikasi (ine 62-153)

```
Future<void> tambahCatatan(  
  String judul, String deskripsi, String gambar) async {  
  await SQLHelper.tambahCatatan(judul, deskripsi, gambar);  
  refreshCatatan();  
}
```

```
Future<void> hapusCatatan(int id) async {  
  catatanDihapus = catatan.firstWhere((item) => item['id'] == id);  
  await SQLHelper.hapusCatatan(id);  
  
  refreshCatatan();  
}
```

```
Future<void> ubahCatatan(  
  int id, String judul, String deskripsi, String gambar) async {  
  await SQLHelper.ubahCatatan(id, judul, deskripsi, gambar);  
  refreshCatatan();  
}
```

```
void modalForm(int? id) async {  
  if (id != null) {  
    final dataCatatan = catatan.firstWhere((item) => item['id'] == id);  
  
    judulController.text = dataCatatan['judul'];  
    deskripsiController.text = dataCatatan['deskripsi'];  
    gambarController.text = dataCatatan['gambar'];  
  }  
  showModalBottomSheet(  
    context: context,  
    builder: (context) => Container(  
      padding: const EdgeInsets.all(15),  
      width: double.infinity,  
      height: 800,  
      child: SingleChildScrollView(  
        child: Column(  
          children: [  
            TextField(  
              controller: judulController,  
              decoration: const InputDecoration(hintText: "masukkan judul"),  
            ),  
            const SizedBox(  
              height: 10,  
            ),  
            TextField(  
              controller: deskripsiController,
```

```

        decoration:
          const InputDecoration(hintText: "masukkan deskripsi"),
        ),
        const SizedBox(
          height: 10,
        ),
        ElevatedButton.icon(
          onPressed: _ambilGambar,
          icon: Icon(
            Icons.image,
            color: Colors.white, // Warna ikon
          ),
          label: Text(
            "Pilih Gambar",
            style: TextStyle(
              color: Colors.white, // Warna teks
            ),
          ),
          style: ElevatedButton.styleFrom(
            primary: Colors.blue, // Warna latar belakang
          ),
        ),
        ElevatedButton(
          onPressed: () async {
            if (id == null) {
              await tambahCatatan(judulController.text,
                deskripsiController.text, gambarController.text);
              print("tambah");
            } else {
              print("update");
              await ubahCatatan(id, judulController.text,
                deskripsiController.text, gambarController.text);
            }
            judulController.text = "";
            deskripsiController.text = "";
            gambarController.text = "";
            Navigator.pop(context);
            refreshCatatan();
          },
          child: Text(id == null ? 'Tambah' : 'Ubah'),
        ),
      ],
    ),
  ),
);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: ListView.builder( // Membuat daftar menggunakan ListView.builder
      itemCount: catatan.length, // Jumlah item dalam daftar adalah panjang dari list catatan
    ),
  );
}

```

```
itemBuilder: (context, index) => Card( // Membuat kartu untuk setiap item dalam daftar
child: ListTile(
title: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Image.file(
File(catatan[index]['gambar']),
width: double.infinity,
height: 200,
fit: BoxFit.cover,
),
const SizedBox(height: 8),
Row(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Expanded(
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
catatan[index]['judul'],
style: TextStyle(
fontSize: 18, fontWeight: FontWeight.bold),
),
const SizedBox(height: 8),
Text(catatan[index]['deskripsi']),
const SizedBox(height: 8),
],
),
),
Padding(
padding: const EdgeInsets.only(
bottom: 8.0), // Atur padding ke bawah ikon
child: Row(
mainAxisAlignment: MainAxisAlignment.end,
children: [
IconButton(
onPressed: () => modalForm(catatan[index]['id']),
icon: const Icon(Icons.edit),
),
const SizedBox(width: 8),
IconButton(
onPressed: () {
hapusCatatan(catatan[index]['id']);
},
icon: const Icon(Icons.delete),
),
],
),
),
),
),
),
),
),
),
),
),
```

```

    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        modalForm(null);
      },
      tooltip: 'Tambah Catatan',
      child: const Icon(Icons.add),
    ),
  );
}
}

```

2.3 Implementasi Fungsionalitas Dasar

2.3.1 Model - foto.dart

```

class Foto {
  late int id; // Variabel untuk menyimpan ID dari foto
  late String judul; // Variabel untuk menyimpan judul dari foto
  late String deskripsi; // Variabel untuk menyimpan deskripsi dari foto
  late String gambar; // Variabel untuk menyimpan URL gambar dari foto

  Foto({
    required this.id, // Parameter wajib untuk menginisialisasi variabel ID
    required this.judul, // Parameter wajib untuk menginisialisasi variabel judul
    required this.deskripsi, // Parameter wajib untuk menginisialisasi variabel deskripsi
    required this.gambar, // Parameter wajib untuk menginisialisasi variabel gambar
  });
}

```

2.3.2 Database – sqllite_db.dart

```

import 'package:sqflite/sqflite.dart' as sql;

class SQLHelper {
  static Future<sql.Database> db() async { // Fungsi untuk mendapatkan koneksi database
    return sql.openDatabase("catatan.db", version: 1, // Buka atau buat database "catatan.db" versi 1
      onCreate: (sql.Database database, int version) async {
        await createTable(database);
      }
    );
  }

  static Future<void> createTable(sql.Database database) async { //ungsi untuk membuat tabel
    await database.execute("""
    CREATE TABLE catatan(
      id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
      judul TEXT,
      deskripsi TEXT,
      gambar TEXT
    );
    """);
  }
}

```

```

    )
    """); // Perintah untuk membuat table database
}

static Future<int> tambahCatatan(String judul, String deskripsi, String gambar) async { //Menambah catatan
baru ke dalam database (CREATE)
    final db = await SQLHelper.db(); // mendapatkan koneksi ke databse
    final data = {'judul': judul, 'deskripsi': deskripsi, 'gambar': gambar}; //D ata yang akan dimasukkan
    return await db.insert("catatan", data); // Masukkan data ke dalam tabel 'catatan'
}

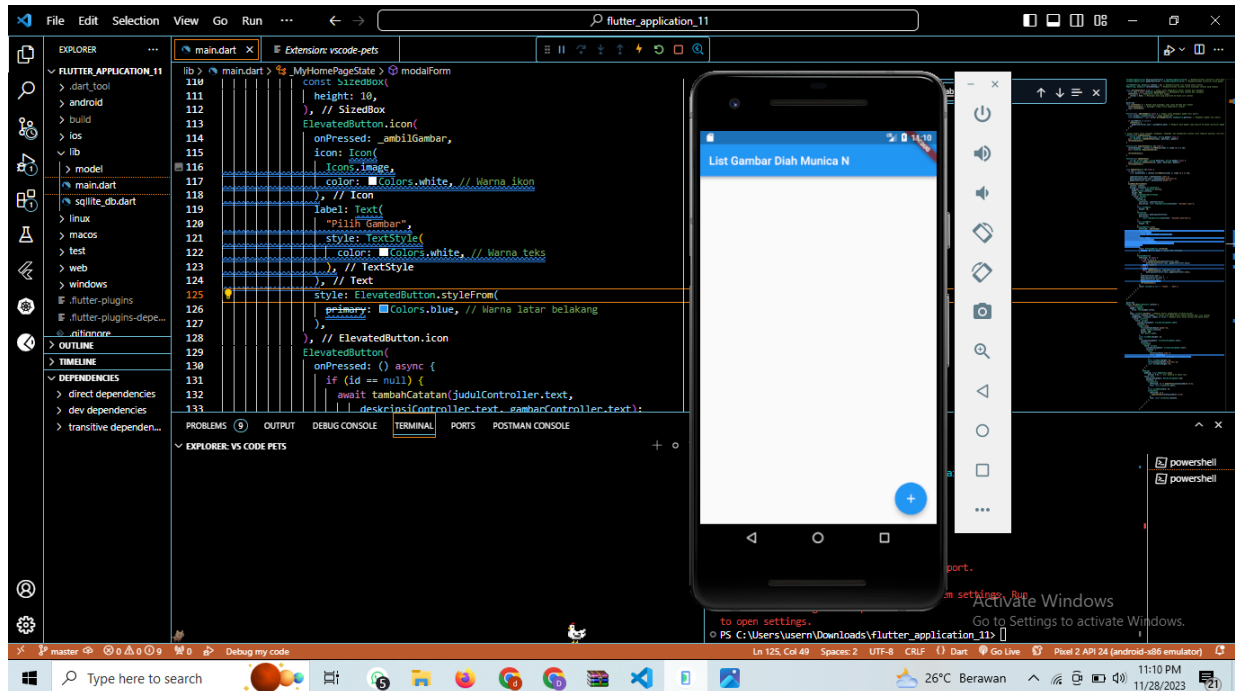
static Future<void> hapusCatatan(int id) async { // Hapus catatan berdasarkan ID (DELETE)
    final db = await SQLHelper.db(); // mendapatkan koneksi ke databse
    await db.delete("catatan", where: "id=$id"); // hapus data berdasarkan id
}

static Future<List<Map<String, dynamic>>> getCatatan() async { // mendapatkan daftar catatan dari database
(READ)
    final db = await SQLHelper.db(); // mendapatkan koneksi ke databse
    return db.query("catatan"); // mengambil semua data dari tabel
}

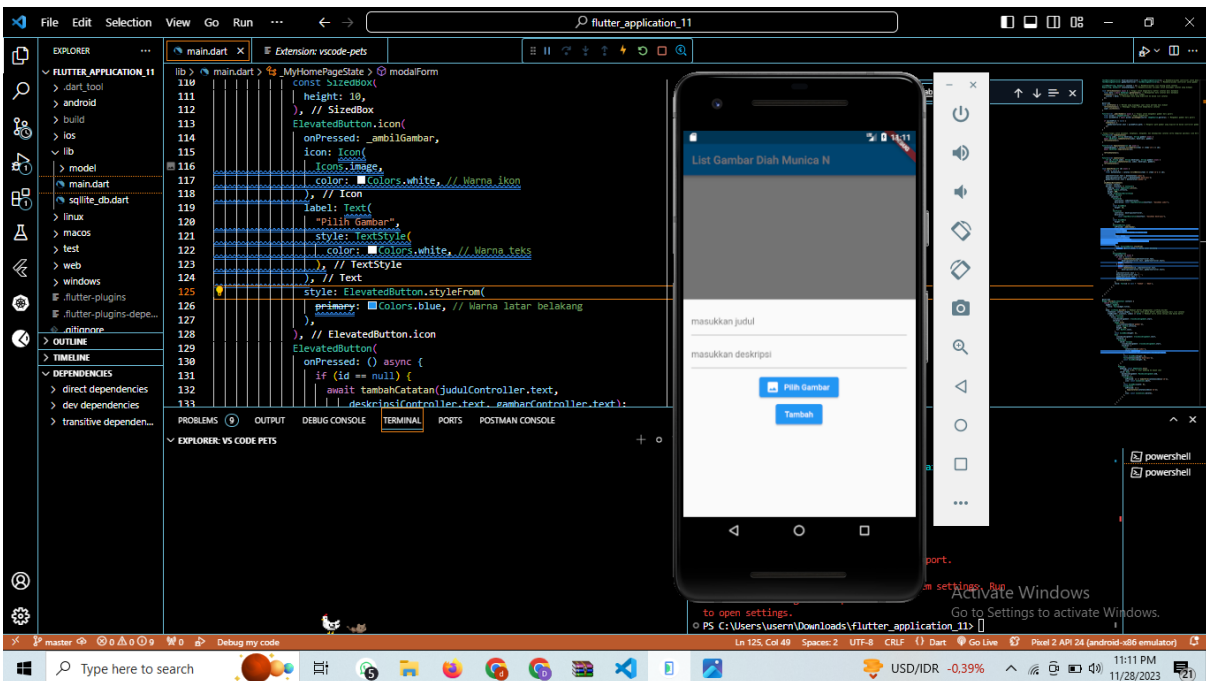
static Future<int> ubahCatatan(int id, String judul, String deskripsi, String gambar) async { // Ubah catatan
berdasarkan ID (UPDATE)
    final db = await SQLHelper.db(); // mendapatkan koneksi ke databse
    final data = {'judul': judul, 'deskripsi': deskripsi, 'gambar': gambar}; // data yang akan di ubah
    return await db.update('catatan', data, where: "id = $id"); // mengubah data dalam tabel berdasrkan id
}
}

```

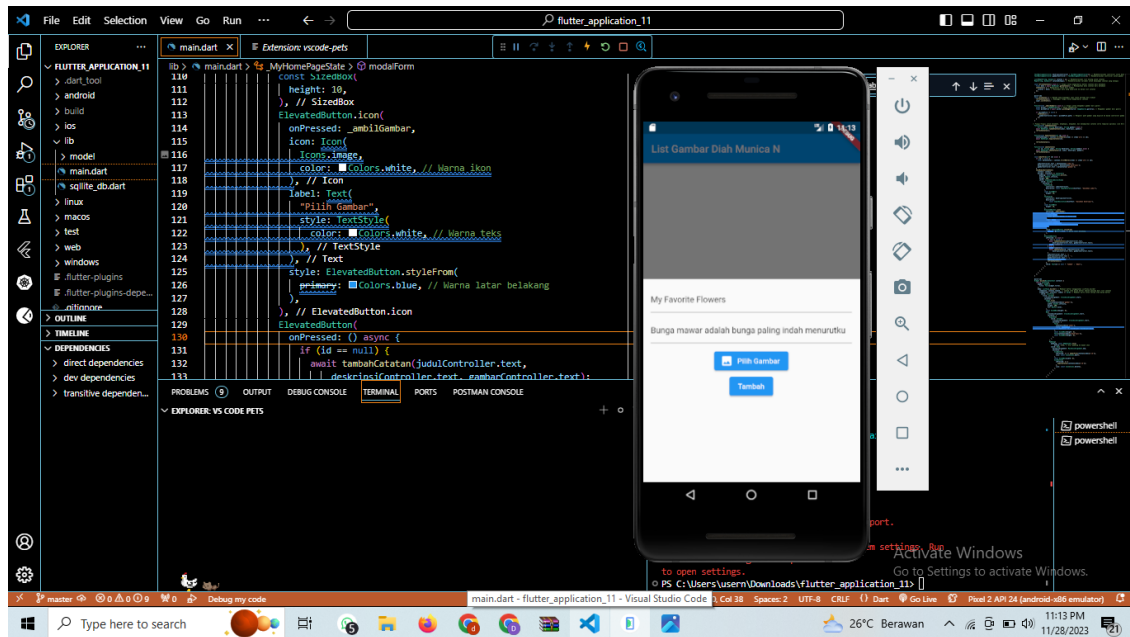

Hasil Tampilan Utama adalah seperti dibawah, tanda (+) dibawah adalah button jika kita ingin mengupload file gambar.



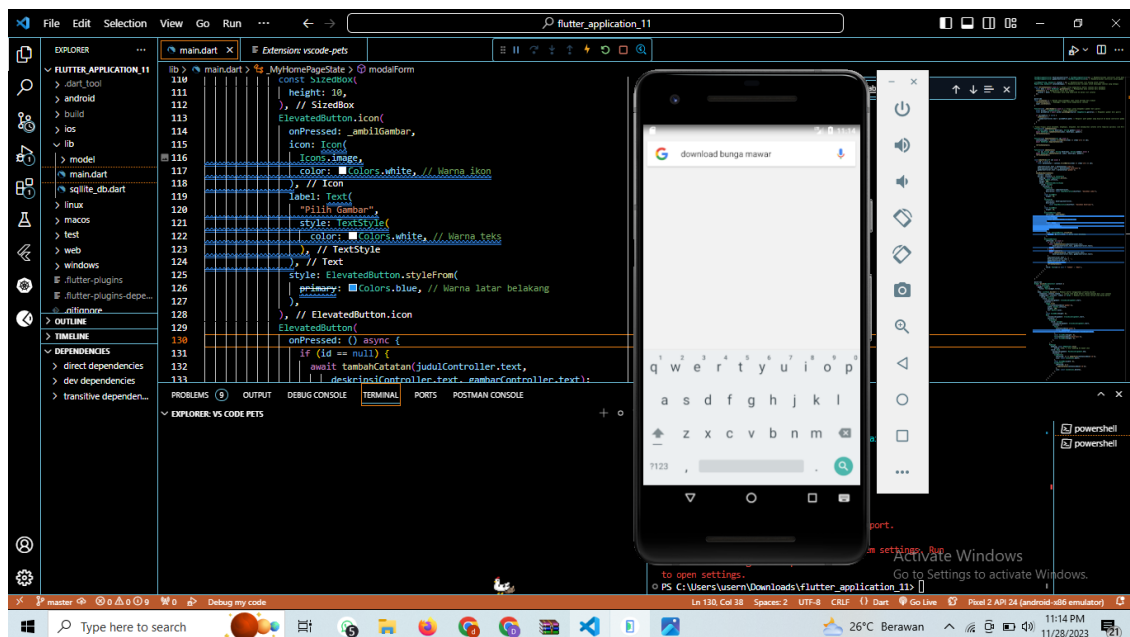
Kemudian kita bisa menuliskan Judul dan Deskripsi sesuai yang kita inginkan.



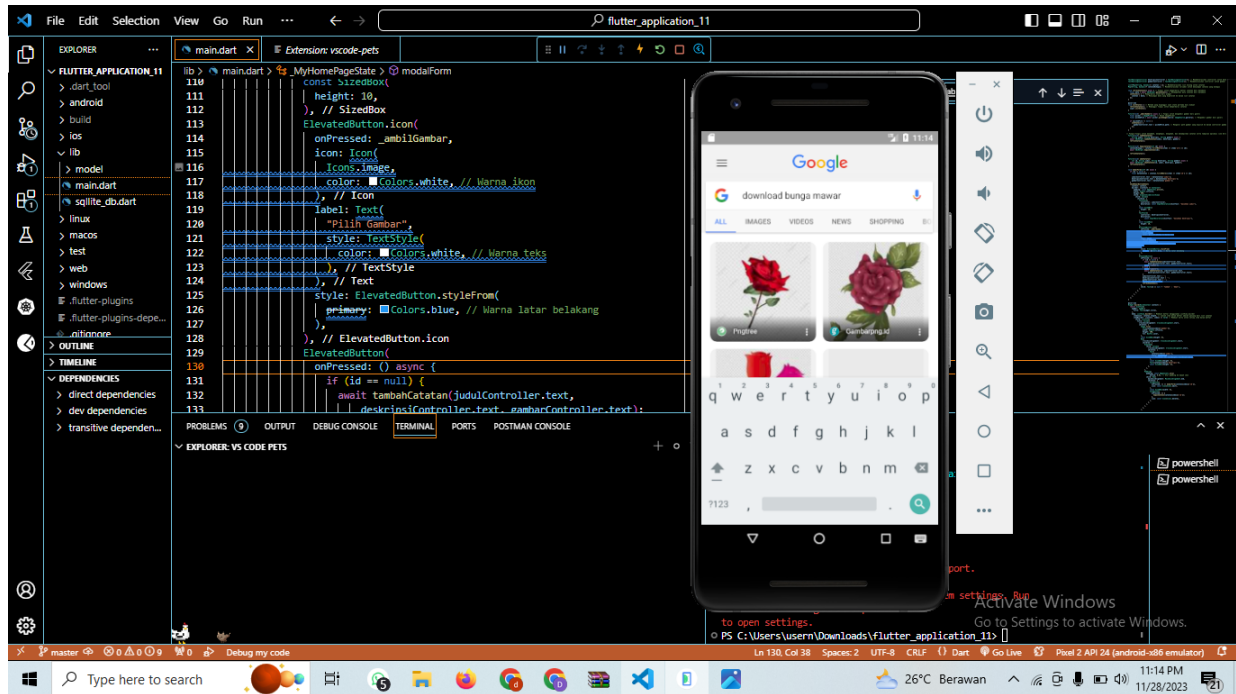
Disini saya menuliskan Judul “My Favorite Flowers” dan deskripsi “ Bunga mawar adalah bunga yang paling indah menurutku”



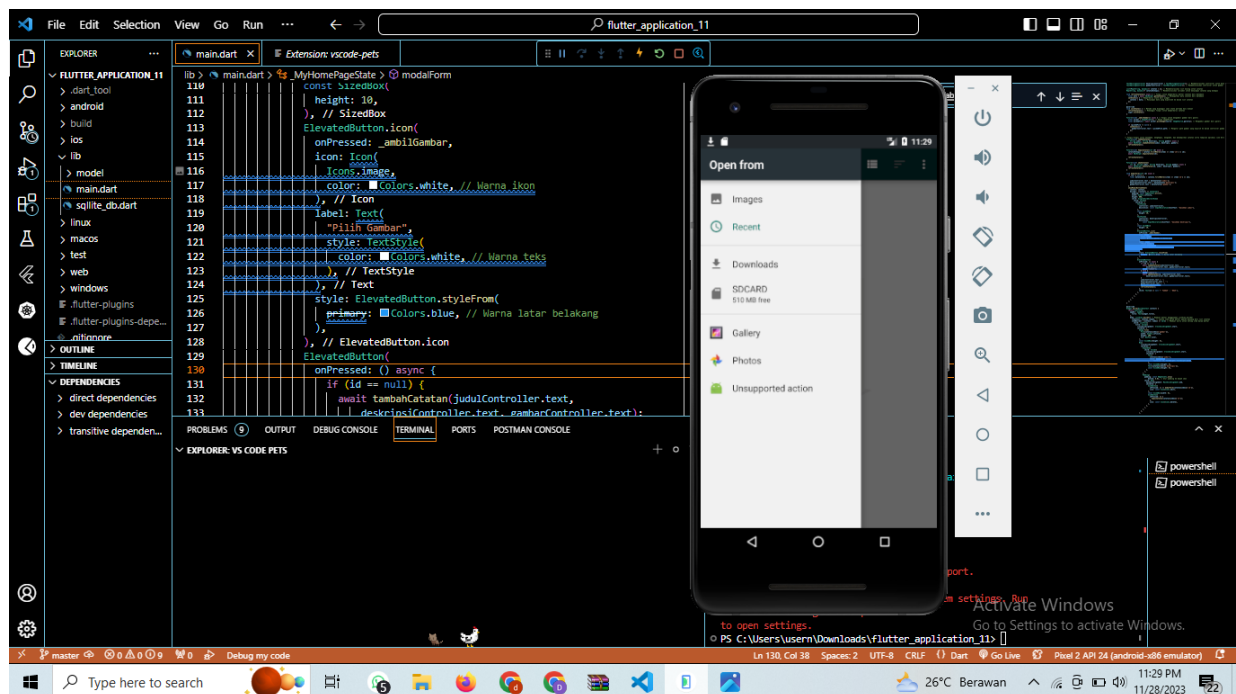
Dan Sebelum, mengupload foto kita bisa download foto terlebih dahulu di Google karena di galeri tidak ada foto.



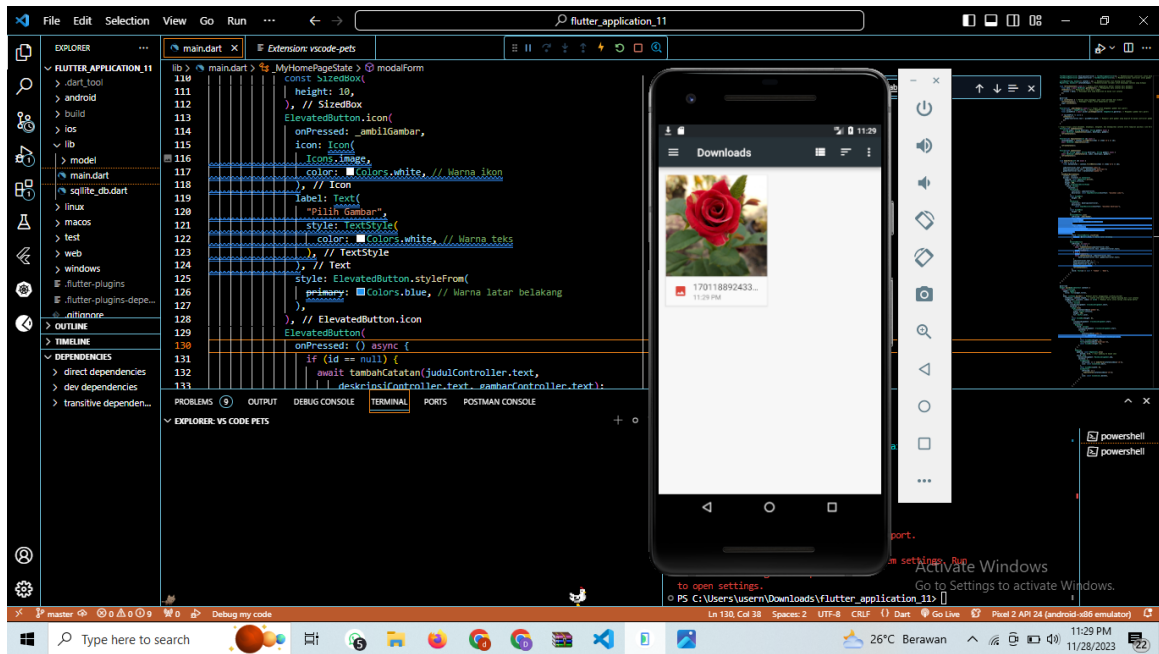
Pilih Foto yang kita inginkan, jika sudah menemukan foto yang kita inginkan download gambar tersebut.



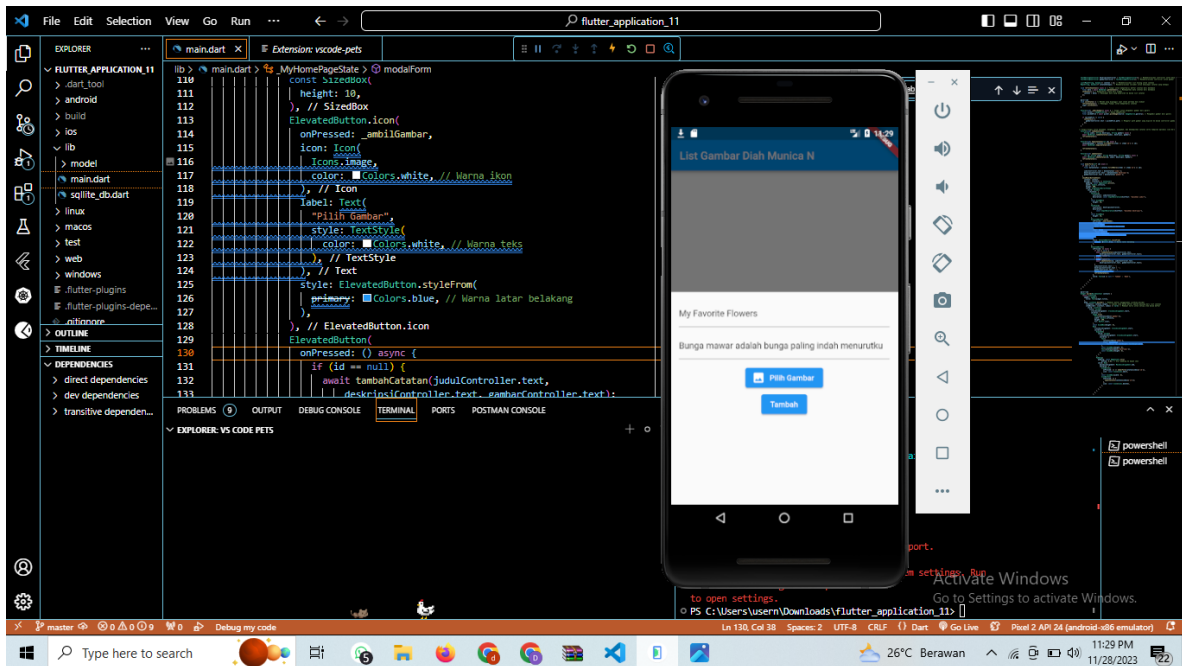
Gambar yang sudah kita download tadi akan masuk dalam folder download.



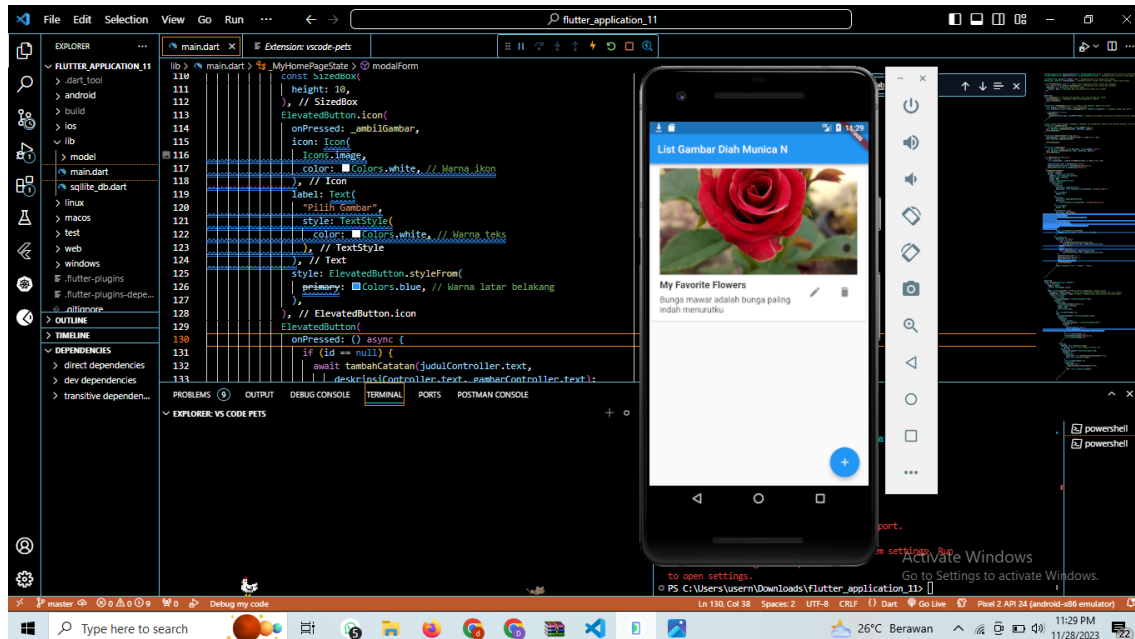
Selanjutny akita klik gambarnya untuk kita upload.



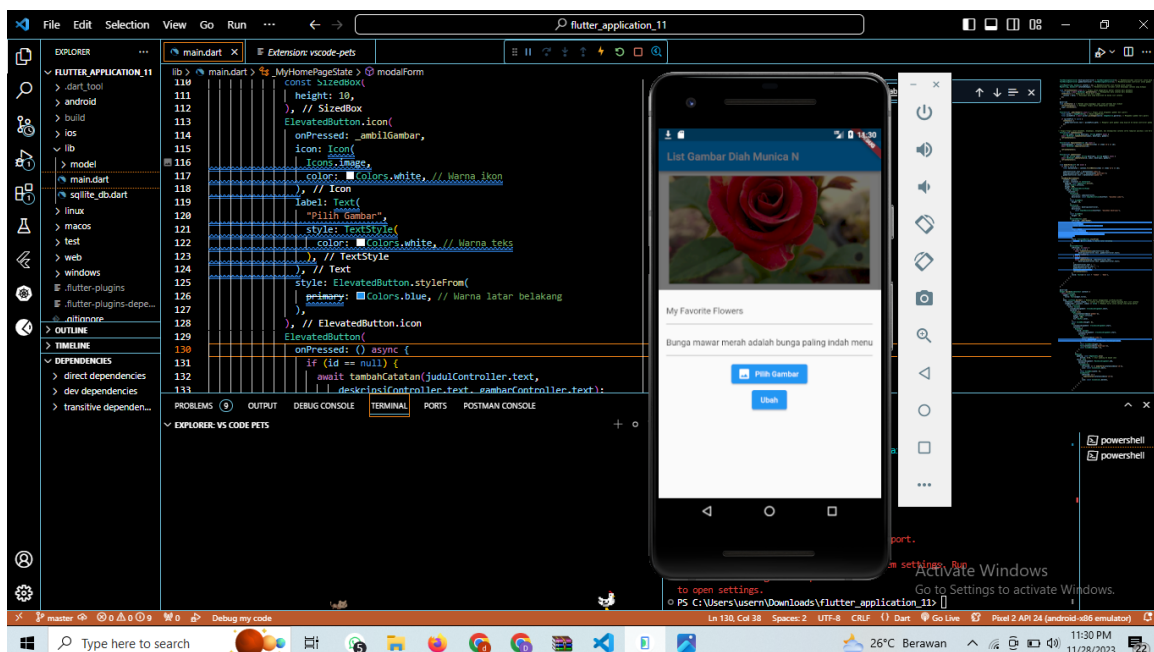
Kita Klik button tambah untuk mengupload gambar.



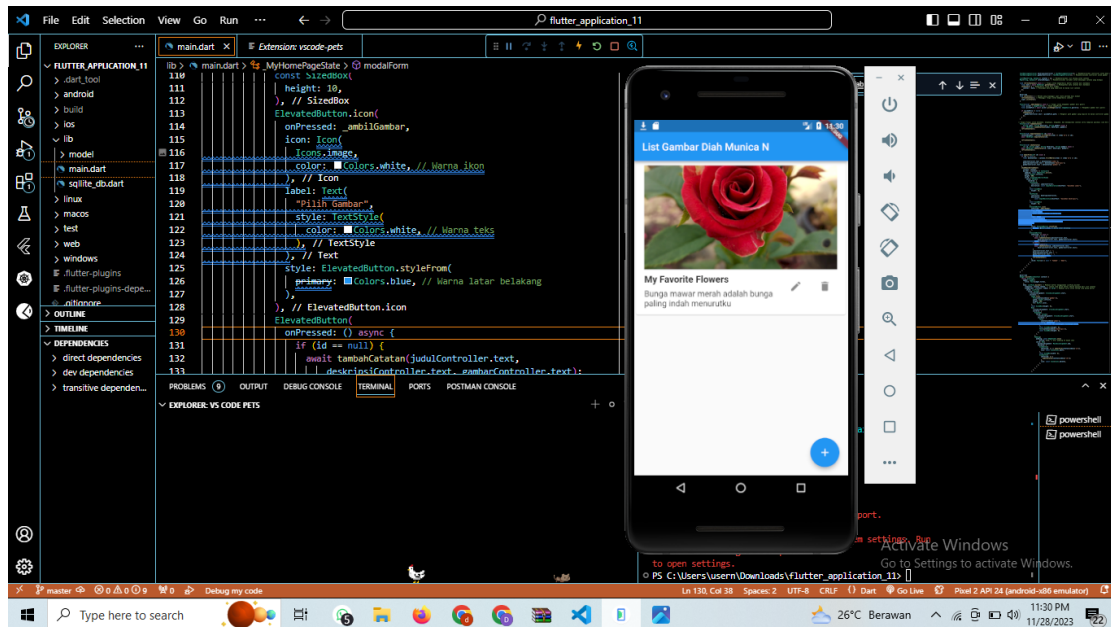
Maka gambar pun akan terupload lengkap dengan judul dan deskripsinya yang sudah kita tulis diawal.



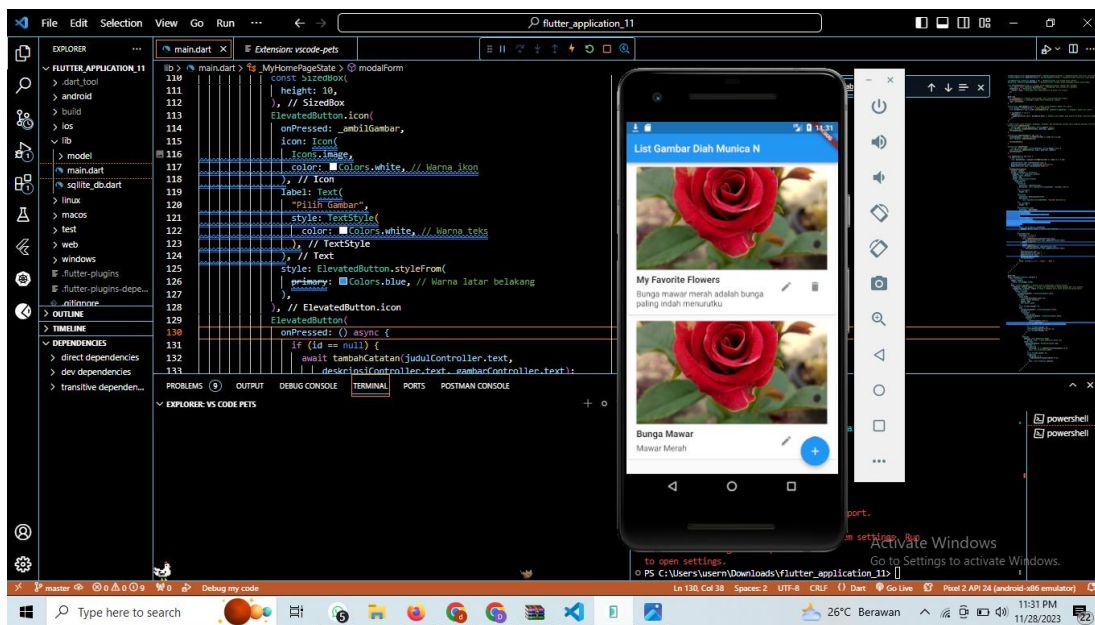
Judul beserta deskripsinya pun juga bisa diedit dengan mengklik icon “pen” disini saya mencoba mengedit deskripsinya menjadi “Bunga mawar **merah** adalah bunga yang paling indah menurutku”



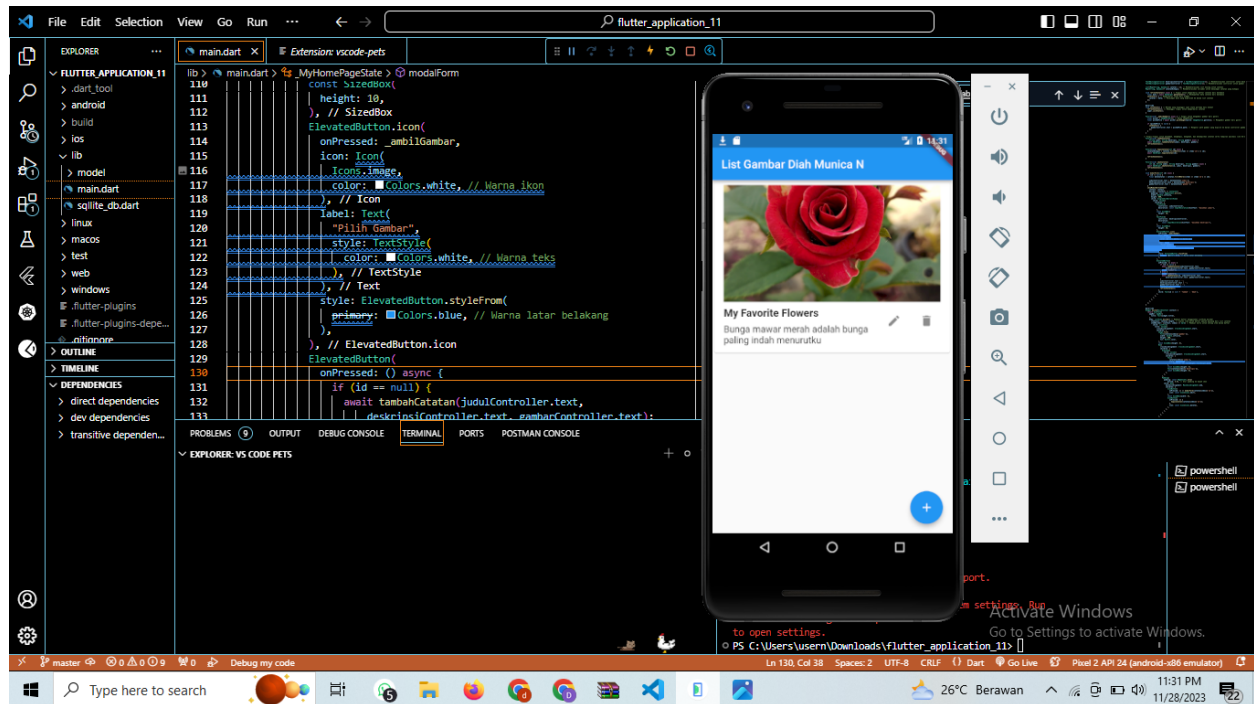
Maka Deskripsi pun berhasil di edit



Disini saya mencoba mengupload 1 konten lagi dan berhasil.



Terakhir saya mencoba menghapus konten yang kedua dengan mengklik icon “tempat sampah” maka konten pun berhasil dihapus.



IV. Kesimpulan

Praktikum ini memberikan pemahaman dasar dalam mengembangkan aplikasi mobile dengan Flutter, dengan fokus khusus pada fungsionalitas pengunggahan file menggunakan SQFLITE. Berikut