ACARA 2 : SQFLITE FLUTTER

Pokok Bahasan	**: SQFLITE FLUTTER**
Acara Praktikum/Pertemuan	**: Minggu 9**
Tempat	**: Daring / Luring Politeknik Negeri Jember**
Alokasi Waktu	**: 3x50 menit**

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep dasar penggunaan SQFLITE FLUTTER.

2. Mahasiswa mampu menggunakan dan menerapkan SQFLITE FLUTTER.

3. Mahasiswa mampu menggunakan dan menerapkan SQFLITE FLUTTER.

b. Indikator

Kemampuan mahasiswa dalam memahami dan menerapkan konsep serta penggunaan SQFLITE FLUTTER untuk CRUD .

c. Dasar Teori

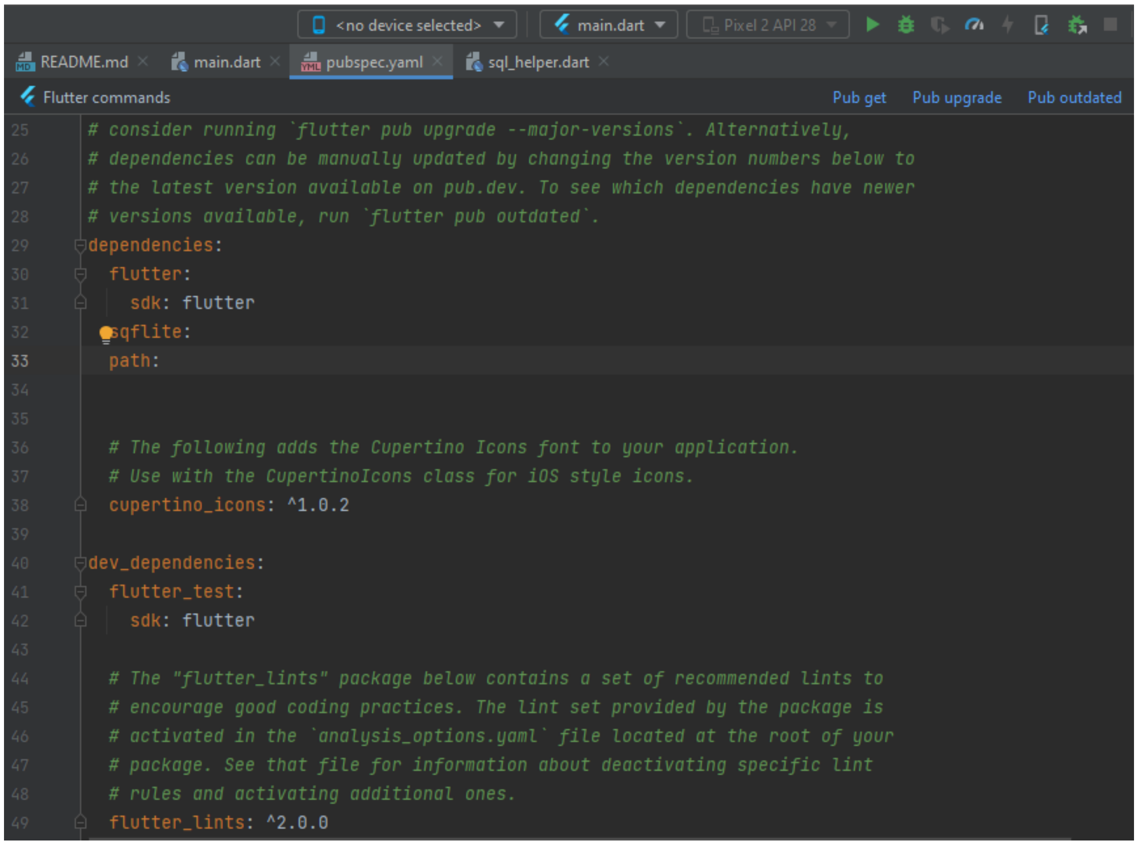
-

d. Alat dan Bahan

1. Notebook
2. Aplikasi Visual Studio Code / Andorid Studio
3. Hive
4. HP Android
5. Kabel Data

e. Prosedur Kerja

1. Buat project baru dengan nama sqflite_flutter.
2. Tambahkan dependencies sqflite dan path pada file pubspec.yaml dan jalankan perintah flutter pub get pada terminal.



```yaml
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
  sqflite:
  path:


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the `analysis_options.yaml` file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^2.0.0
```

3. Ubah kode pada kelas MyApp menjadi seperti berikut.

```dart
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // Remove the debug banner
        debugShowCheckedModeBanner: false,
        title: 'Kindacode.com',
        theme: ThemeData(
          primarySwatch: Colors.orange,
        ),
        home: const HomePage());
  }
}
```

4. Buat stateful widget dengan nama HomePage dan isikan kodenya menjadi seperti berikut.

```dart
class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  _HomePageState createState() => _HomePageState();
}
```

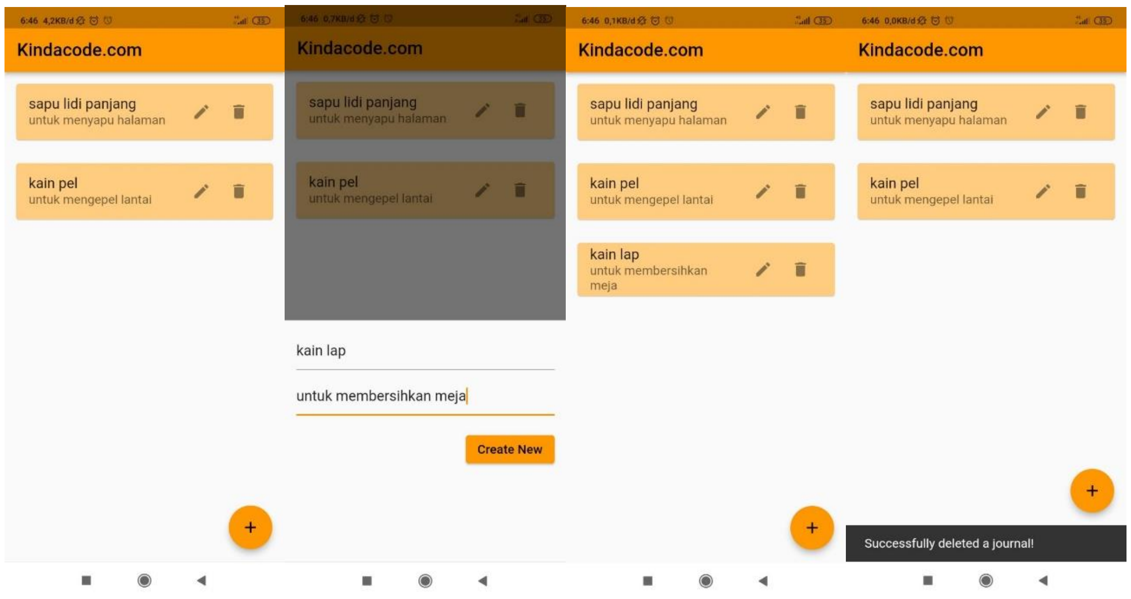
```
 class _HomePageState extends State<HomePage> {
  // All journals
  List<Map<String, dynamic>> _journals = [];

  bool _isLoading = true;
  // This function is used to fetch all data from the database
  void _refreshJournals() async {
    final data = await SQLHelper.getItems();
    setState(() {
      _journals = data;
      _isLoading = false;
    });
  }

  @override
  void initState() {
    super.initState();
    _refreshJournals(); // Loading the diary when the app starts
  }

  final TextEditingController _titleController =
TextEditingController();
  final TextEditingController _descriptionController =
TextEditingController();

  // This function will be triggered when the floating button is pressed
  // It will also be triggered when you want to update an item
  void _showForm(int? id) async {
    if (id != null) {
      // id == null -> create new item
      // id != null -> update an existing item
      final existingJournal =
      _journals.firstWhere((element) => element['id'] == id);
      _titleController.text = existingJournal['title'];
      _descriptionController.text = existingJournal['description'];
    }

    showModalBottomSheet(
        context: context,
        elevation: 5,
        isScrollControlled: true,
        builder: (_) => Container(
          padding: EdgeInsets.only(
            top: 15,
            left: 15,
            right: 15,
            // this will prevent the soft keyboard from covering the
text fields
            bottom: MediaQuery.of(context).viewInsets.bottom + 120,
          ),
          child: Column(
            mainAxisSize: MainAxisSize.min,
            crossAxisAlignment: CrossAxisAlignment.end,
            children: [
              TextField(
                controller: _titleController,
                decoration: const InputDecoration(hintText: 'Title'),
```

```dart
              ),
              const SizedBox(
                height: 10,
              ),
              TextField(
                controller: _descriptionController,
                decoration: const InputDecoration(hintText:
'Description'),
              ),
              const SizedBox(
                height: 20,
              ),
              ElevatedButton(
                onPressed: () async {
                  // Save new journal
                  if (id == null) {
                    await _addItem();
                  }

                  if (id != null) {
                    await _updateItem(id);
                  }

                  // Clear the text fields
                  _titleController.text = '';
                  _descriptionController.text = '';

                  // Close the bottom sheet
                  Navigator.of(context).pop();
                },
                child: Text(id == null ? 'Create New' : 'Update'),
              )
            ],
          ),
        ));
  }

// Insert a new journal to the database
  Future<void> _addItem() async {
    await SQLHelper.createItem(
        _titleController.text, _descriptionController.text);
    _refreshJournals();
  }

  // Update an existing journal
  Future<void> _updateItem(int id) async {
    await SQLHelper.updateItem(
        id, _titleController.text, _descriptionController.text);
    _refreshJournals();
  }

  // Delete an item
  void _deleteItem(int id) async {
    await SQLHelper.deleteItem(id);
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text('Successfully deleted a journal!'),
    ));
    _refreshJournals();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Kindacode.com'),
      ),
      body: _isLoading
          ? const Center(
        child: CircularProgressIndicator(),
      )
          : ListView.builder(
        itemCount: _journals.length,
        itemBuilder: (context, index) => Card(
          color: Colors.orange[200],
          margin: const EdgeInsets.all(15),
          child: ListTile(
              title: Text(_journals[index]['title']),
              subtitle: Text(_journals[index]['description']),
              trailing: SizedBox(
                width: 100,
                child: Row(
                  children: [
                    IconButton(
                      icon: const Icon(Icons.edit),
                      onPressed: () =>
_showForm(_journals[index]['id']),
                    ),
                    IconButton(
                      icon: const Icon(Icons.delete),
                      onPressed: () =>
                          _deleteItem(_journals[index]['id']),
                    ),
                  ],
                ),
              )),
        ),
      ),
      floatingActionButton: FloatingActionButton(
        child: const Icon(Icons.add),
        onPressed: () => _showForm(null),
      ),
    );
  }
}
```

5. Buat file sql_helper.dart di dalam folder lib dan import ke ke dalam main.dart.
6. Isikan kode sql_helper.dart seperti berikut.

```
import 'package:flutter/foundation.dart';
import 'package:sqflite/sqflite.dart' as sql;

class SQLHelper {
  static Future<void> createTables(sql.Database database) async {
    await database.execute("""CREATE TABLE items(
        id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
        title TEXT,
        description TEXT,
        createdAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
      )
      """);
  }
// id: the id of a item
// title, description: name and description of your activity
// created_at: the time that the item was created. It will be
automatically handled by SQLite

  static Future<sql.Database> db() async {
    return sql.openDatabase(
      'kindacode.db',
      version: 1,
      onCreate: (sql.Database database, int version) async {
        await createTables(database);
      },
    );
  }

  // Create new item (journal)
  static Future<int> createItem(String title, String? descrption) async
{
    final db = await SQLHelper.db();

    final data = {'title': title, 'description': descrption};
    final id = await db.insert('items', data,
        conflictAlgorithm: sql.ConflictAlgorithm.replace);
    return id;
  }

  // Read all items (journals)
  static Future<List<Map<String, dynamic>>> getItems() async {
    final db = await SQLHelper.db();
    return db.query('items', orderBy: "id");
  }

  // Read a single item by id
  // The app doesn't use this method but I put here in case you want to
see it
  static Future<List<Map<String, dynamic>>> getItem(int id) async {
    final db = await SQLHelper.db();
    return db.query('items', where: "id = ?", whereArgs: [id], limit:
1);
  }

  // Update an item by id
  static Future<int> updateItem(
      int id, String title, String? descrption) async {
    final db = await SQLHelper.db();

    final data = {
      'title': title,
      'description': descrption,
      'createdAt': DateTime.now().toString()
    };

    final result =
    await db.update('items', data, where: "id = ?", whereArgs: [id]);
    return result;
  }

  // Delete
  static Future<void> deleteItem(int id) async {
    final db = await SQLHelper.db();
    try {
      await db.delete("items", where: "id = ?", whereArgs: [id]);
    } catch (err) {
      debugPrint("Something went wrong when deleting an item: $err");
    }
  }
}
```

7. Jalankan aplikasi pada emulator atau real device. Sqflite ini tidak support dijalankan pada web browser.
8. Jika tidak ada error akan tampil seperti berikut.


Kindacode.com
sapu lidi panjang
untuk menyapu halaman
kain pel
untuk mengepel lantai
kain lap
untuk membersihkan meja
Create New
Successfully deleted a journal!