

Bab VII

Konsep Database pada Flutter



Flutter

Dosen : Bambang Sugiarto, ST, MT

**Program Studi S1 Teknik Informatika
Fakultas Teknik
Universitas Sangga Buana YPKP Bandung**

*Dirangkum dari berbagai sumber referensi
(hanya untuk penggunaan internal/tidak untuk dipublikasikan)*

Pendahuluan

- Flutter menyediakan banyak paket lanjutan untuk bekerja dengan database.
- Paket terpenting tersebut adalah :

- ***sqflite***

Digunakan untuk mengakses dan memanipulasi database SQLite,

- ***firebase_database***

Digunakan untuk mengakses dan memanipulasi database NoSQL yang dihosting di cloud dari Google.

Pada bab ini akan dijelaskan mengenai database SQLite dengan menggunakan paket sqflite.



Database SQLite

- **Database SQLite** merupakan database yang bersifat de-facto dan mesin database tertanam yang berbasis SQL standar dan merupakan database dengan mesin database kecil.
- Paket sqflite menyediakan banyak fungsi untuk bekerja secara efisien dengan database SQLite dan menyediakan metode standar untuk memanipulasi mesin database SQLite.
- Fungsi-fungsi inti yang disediakan oleh paket sqflite adalah sebagai berikut :
 - Create / Open (openDatabase method) SQLite Database



- Execute SQL statement (execute method) terhadap SQLite Database
- Advanced query methods (query method) untuk mengurangi kode yang diperlukan dalam melakukan query dan mendapatkan informasi dari database SQLite.



Proses CRUD pada SQLite

- Sqflite memungkinkan kita melakukan operasi create, read, update, delete (CRUD) dasar.
- Berikut ringkasan operasi yang akan digunakan :

➤ *Create*

```
Future<int> insert(String table, Map<String, dynamic> values,  
    {String nullColumnHack, ConflictAlgorithm conflictAlgorithm})
```

✓ *Tables*

Di sinilah kita akan memasukkan entri

✓ *Values*

Berisi daftar nama kolom dan nilainya masing-masing.

✓ *nullColumnHack*

Opsional field yang digunakan saat kolom kosong

✓ *conflictAlgorithm*

- Merupakan algoritma penanganan konflik.
- Kita akan melihat daftar opsi dalam kode algoritma tersebut di bawah ini.
- Hal Ini mengembalikan ID record yang dimasukkan terakhir.

```
enum ConflictAlgorithm {  
    rollback,  
    abort,  
    fail,  
    ignore,  
    replace,  
}
```

- 
- conflict algorithm options ini didokumentasikan dengan baik dan berikut uraian singkatnya :

- ❑ *rollback*

Mengembalikan ke transaksi state inisialisasi pada section selanjutnya jika bernilai satu. Jika tidak, maka akan dibatalkan.

- ❑ *abort*

Cukup menghentikan operasi.

- ❑ *fail*

Mirip dengan abort, tetapi berfungsi untuk mengembalikan kegagalan (failure).

- ❑ *ignore*

Tidak menghentikan transaksi, tetapi menghentikan operasi saat ini.

- ❑ *replace*

Mengganti entri yang konflik



➤ Read

```
Future<List<Map<String, dynamic>>> query(String table,  
    {bool distinct,  
    List<String> columns,  
    String where,  
    List<dynamic> whereArgs,  
    String groupBy,  
    String having,  
    String orderBy,  
    int limit,  
    int offset});
```

- ✓ Fungsi ini juga didokumentasikan dengan sangat baik, sehingga kita dapat memeriksa secara spesifik untuk setiap parameter.
- ✓ Dua parameter, *where* dan *whereArgs*, digunakan bersama untuk menggantikan nilai di klausa *where* sambil mencegah injeksi SQL.
- ✓ Hal ini akan mengembalikan entri yang cocok sebagai daftar.

➤ Update

```
Future<int> update(String table, Map<String, dynamic> values,  
    {String where,  
    List<dynamic> whereArgs,  
    ConflictAlgorithm conflictAlgorithm});
```

- ✓ Fungsi *update* memiliki struktur yang mirip dengan fungsi *insert*
- ✓ Tetapi memungkinkan kita menentukan klausa *where*, seperti metode *query* di atas.
- ✓ Parameter terakhir, *conflictAlgorithm*, sama seperti di *insert* yang sudah dibahas sebelumnya.
- ✓ Hal ini mengembalikan jumlah perubahan yang dibuat.



➤ Delete

```
Future<int> delete(String table, {String where, List<dynamic> whereArgs});
```

- ✓ Kita dapat menggunakan *where* dan *whereArgs* untuk memberikan klausa *where* yang aman.
- ✓ Hal ini mengembalikan jumlah entri yang dihapus.



References

- SQLite in Flutter, raywenderlich.com,
<https://www.raywenderlich.com/9376629-sqlite-in-flutter>
- Flutter - Database Concepts, Tutorialspoint,
https://www.tutorialspoint.com/flutter/flutter_database_concepts.htm

