Rakamin Academy

mandiri

Project Based Internship

Android

SQLite Database

Daftar Isi

A. Penyimpanan Lokal pada Aplikasi Android

Android menyediakan beberapa opsi untuk penyimpanan data pada sebuah aplikasi. Berikut adalah beberapa opsi yang dapat dipilih ketika kita ingin mendukung penyimpanan data pada aplikasi yang dibangun :

- **Shared Preferences**: menyimpan data dalam bentuk pasangan kunci-nilai (key-value pairs)
- **Penyimpanan Internal**: menyimpan data pada memori smartphone
- **Penyimpanan Eksternal**: menyimpan data publik pada penyimpanan eksternal bersama.
- **Database SQLite**: menyimpan data terstruktur pada basis data private
- **Library Room persistence**: opsi ini merupakan bagian dari library Komponen Arsitektur Android. Room meng-cache basis data SQLite secara lokal, kemudian secara otomatis melakukan sinkronisasi perubahan ke basis data jaringan.

Opsi-opsi penyimpanan data tersebut dapat digunakan bergantung pada kebutuhan penyimpanan data pada aplikasi yang dibangun. Contohnya, apakah data yang ada dapat diakses oleh aplikasi lain atau cenderung hanya untuk internal aplikasi saja. Atau apakah data yang disimpan membutuhkan ruang yang besar atau kecil. Hal-hal tersebut dapat menjadi pertimbangan ketika memilih solusi penyimpanan data.

Berikut ini merupakan beberapa kelebihan dalam menggunakan database lokal pada aplikasi Android yang kita kembangkan :

1. User experience (cepat)
2. Hemat daya tahan baterai device

3. Hemat kuota
4. Mengurangi beban server dan bandwidth jaringan
5. Dapat diakses tanpa koneksi internet

B. SQLite

SQLite merupakan sebuah Relational Database Management System (RDMS) yang tidak memerlukan server untuk beroperasi. SQLite ideal untuk digunakan pada penyimpanan data yang berulang dan terstruktur, seperti data kontak misalnya. Android secara default telah menyediakan metode penyimpanan data seperti SQL. Penggunaan dan kueri yang dipakai pun mirip dengan SQL biasa. Meskipun berjalan seperti database, SQLite umumnya berukuran kecil dan dapat berjalan di perangkat dengan memori terbatas seperti smartphone. SQLite disertakan secara default di semua perangkat Android dan yang paling penting tidak diperlukan proses autentikasi atau penyiapan administratif seperti yang dilakukan pada perangkat lunak database berskala besar.

C. Komponen SQLite

Berikut ini merupakan komponen yang terdapat pada SQLite :

1. **SQLiteOpenHelper**

   SQLiteOpenHelper merupakan salah satu fitur yang digunakan di dalam aplikasi Android Studio untuk membuat tabel yang dapat menyimpan berbagai data seperti String dan Integer. SQLiteOpenHelper memiliki beberapa metode yang dapat digunakan, yaitu:

   - onCreate(), akan dipanggil bila sebelumnya tidak ada database

- onUpgrade(), akan dipanggil bila ditemukan database yang sama namun memiliki versi yang beda, digunakan juga untuk mengubah skema database
- onOpen(), dijalankan pada saat database dalam keadaan terbuka
- getWritableDatabase(), memanggil database agar dapat ditambahkan datanya
- getReadableDatabase(), memanggil database agar dapat dilihat isi database-nya

2. **SQLiteDatabase**

SQLiteDatabase merupakan sebuah class yang berfungsi untuk mengatur database. SQLiteDatabase memiliki beberapa method, yaitu sebagai berikut:

- insert(), method yang digunakan untuk menambahkan data ke dalam database
- update(), method yang digunakan untuk memperbaharui data yang ada di database
- delete(), method yang digunakan untuk menghapus data yang ada di database
- execSQL(), method yang digunakan untuk mengeksekusi sintak pada SQL

D. SQLite Query

1. Membuat Tabel pada Database

Query :

"CREATE TABLE book (id INTEGER PRIMARY KEY, title TEXT, author TEXT, genre TEXT, pages INT)"

Dalam bentuk kode :

```
private val SQL_CREATE_ENTRIES="CREATE TABLE ${BookEntry.TABLE_NAME}"+
        " (${BookEntry._ID} INTEGER PRIMARY KEY," +
        " ${BookEntry.COLUMN_TITLE} TEXT," +
        " ${BookEntry.COLUMN_AUTHOR} TEXT," +
        " ${BookEntry.COLUMN_GENRE} TEXT," +
        " ${BookEntry.COLUMN_PAGES} INT)"
```

2. Delete Tabel

Query :

"DROP TABLE IF EXISTS book"

Dalam bentuk kode :

```
private val SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS
$BookEntry.TABLE_NAME"
```

3. Insert Data Baru ke dalam Tabel

```
// Gets the data repository in write mode
val dbHelper: BooksHelper = BooksHelper(this)
val db: SQLiteDatabase = dbHelper.getWritableDatabase()

// Create a new map of values, where column names are the keys
val values = ContentValues()
values.put(BookEntry.COLUMN_TITLE, "Clean Code")
values.put(BookEntry.COLUMN_AUTHOR, "Robert C. Martin")
values.put(BookEntry.COLUMN_GENRE, "Programming")
values.put(BookEntry.COLUMN_PAGES, 434)

// Insert the new row, returning the primary key value of the new row
val newRowId: Long = db.insert(BookEntry.TABLE_NAME, null, values)
```

4. Read Data

```
val db: SQLiteDatabase = dbHelper.getReadableDatabase()
val cursor: Cursor = db.query(
        BookEntry.TABLE_NAME,   // table
        null,                   // column
        null,                   // selection
        null,                   // selectionArgs
        null,                   // groupBy
        null,                   // having
        null,                   // orderBy
        null                    // limit
)
```

5. Update Data

```
val db = dbHelper.getWritableDatabase()

val values = ContentValues()
values.put(BookEntry.COLUMN_TITLE, "Clean Architecture") // new value

val selection = BookEntry.COLUMN_GENRE + " = ?"
val selectionArgs = { "Programming" }

int count = db.update(
    BookColums.TABLE_NAME,
    values,
    selection,
    selectionArgs
)
```

6. Delete Data

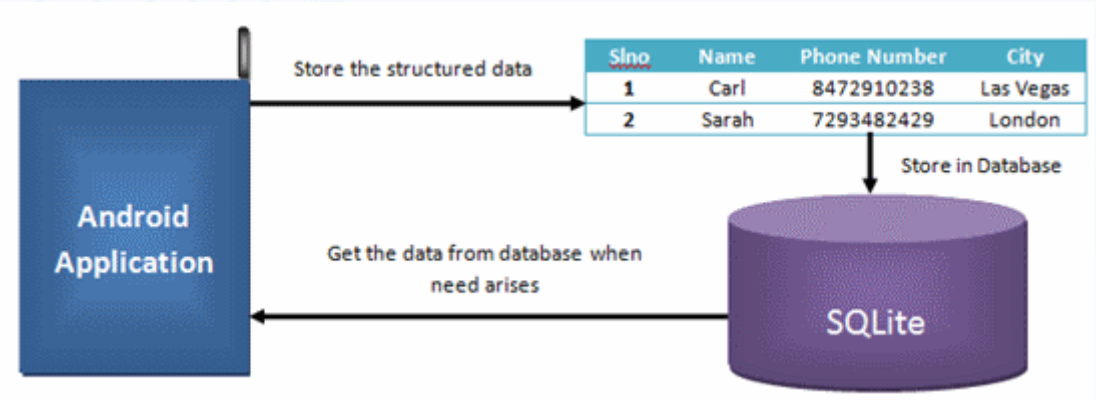
```
// Define 'where' part of query.
val selection = BookColums.COLUMN_AUTHOR + " LIKE ?"

// Specify arguments in placeholder order.
val selectionArgs = { "Hirata" }

// Execute SQL statement
db.delete(
    BookColums.TABLE_NAME,
    selection,
    selectionArgs
)
```

E. Komponen Class Pembuatan CRUD dengan SQLite

Untuk membuat aplikasi CRUD menggunakan SQLite, diperlukan dua komponen berikut, yaitu :

1. **Class Model** (Entity Data) : Representasi dari data yang akan disimpan ke dalam database

2. **Database Handler Class** : Class yang berfungsi untuk meng-handle operasi CRUD (Create, Read, Update, and Delete) dan meng-extends class SQLiteOpenHelper



References

https://ishlah.id/pertemuan-11-penyimpanan-data-aplikasi-android/

https://socs.binus.ac.id/2018/08/08/sqliteopenhelper-dan-sqlitedatabase-dalam-android-studio/

https://developer.android.com/training/data-storage/sqlite