

SQLite



SQLite adalah sebuah database open-source yang menggunakan SQL (Structured Query Language) untuk mengakses data



SQLite sudah tersedia di Android secara bawaan sehingga mudah digunakan oleh para pengembang aplikasi



SQLite memiliki kelebihan seperti ukuran file yang kecil, mudah diintegrasikan dengan aplikasi, dan support untuk transaksi SQL.



PENERAPAN

SQLite

Menggunakan SQLite

Untuk menggunakan SQLite pada Kotlin untuk database lokal, Anda memerlukan beberapa file dan dependensi berikut :

1. Kelas Database Helper: Kelas ini digunakan untuk membuat dan mengelola database SQLite. Kelas ini harus mewarisi kelas SQLiteOpenHelper.
2. Kelas Model: Kelas ini mewakili tabel dalam database SQLite dan digunakan untuk mengakses dan memanipulasi data dalam tabel.
3. Kelas Kontrak: Kelas ini berisi konstanta dan nama tabel, kolom, dan perintah SQL lainnya yang diperlukan untuk membuat dan memodifikasi tabel.
4. Kelas Query: Kelas ini digunakan untuk mengeksekusi perintah SQL pada database SQLite, seperti SELECT, INSERT, UPDATE, dan DELETE (perintah query bisa dibuat dalam 1 pada class Helper).
5. Dependensi: Anda perlu menambahkan dependensi SQLite pada file build.gradle Anda. Di Kotlin, Anda dapat menggunakan dependensi `sqlite-jdbc`, atau lebih umumnya, `androidx.sqlite:sqlite-ktx`.

Selain itu, kita juga memerlukan pengetahuan dasar tentang perintah SQL untuk membuat tabel, menyisipkan data, mengambil data, dan memodifikasi data pada database SQLite.



```
class MahasiswaDbHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "Mahasiswa.db"
        private const val DATABASE_VERSION = 1
        private const val SQL_CREATE_ENTRIES =
            "CREATE TABLE ${MahasiswaContract.MahasiswaEntry.TABLE_NAME} (" +
                "${MahasiswaContract.MahasiswaEntry.COLUMN_NAME_EMAIL} TEXT PRIMARY KEY," +
                "${MahasiswaContract.MahasiswaEntry.COLUMN_NAME_NAMA} TEXT," +
                "${MahasiswaContract.MahasiswaEntry.COLUMN_NAME_NIM} TEXT," +
                "${MahasiswaContract.MahasiswaEntry.COLUMN_NAME_PASSWORD} TEXT)"

        private const val SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS ${MahasiswaContract.MahasiswaEntry.TABLE_NAME}"
    }

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(SQL_CREATE_ENTRIES)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
        db.execSQL(SQL_DELETE_ENTRIES)
        onCreate(db)
    }
}
```

1. Kelas Database Helper



```
data class Mahasiswa(  
    val email: String,  
    val nama: String,  
    val nim: String,  
    val password: String  
)
```




```
object MahasiswaContract {  
  
    object MahasiswaEntry : BaseColumns {  
        const val TABLE_NAME = "mahasiswa"  
        const val COLUMN_NAME_EMAIL = "email"  
        const val COLUMN_NAME_NAMA = "nama"  
        const val COLUMN_NAME_NIM = "nim"  
        const val COLUMN_NAME_PASSWORD = "password"  
    }  
}
```



```
fun insertData(mahasiswa: Mahasiswa){
    val db = writableDatabase

    val sql = "INSERT INTO ${MahasiswaContract.MahasiswaEntry.TABLE_NAME} " +
        "(${MahasiswaContract.MahasiswaEntry.COLUMN_EMAIL}, " +
        "${MahasiswaContract.MahasiswaEntry.COLUMN_NAMA}, " +
        "${MahasiswaContract.MahasiswaEntry.COLUMN_NIM}, " +
        "${MahasiswaContract.MahasiswaEntry.COLUMN_PASSWORD} )" +
        "VALUES ( '${mahasiswa.email}', '${mahasiswa.nama}', '${mahasiswa.nim}', '${mahasiswa.password} )'"

    db.execSQL(sql)
    db.close()

}

fun hapusData(email : String){
    val db = writableDatabase
    val table = MahasiswaContract.MahasiswaEntry.TABLE_NAME
    val emailColumn = MahasiswaContract.MahasiswaEntry.COLUMN_EMAIL
    val sql = "DELETE FROM " + table + " WHERE " + emailColumn + " = '" + email + "'"
    db.execSQL(sql)
    db.close()
}

fun updateData(mahasiswa: Mahasiswa){
    val db = writableDatabase

    val sql = "UPDATE ${MahasiswaContract.MahasiswaEntry.TABLE_NAME} SET " +
        "${MahasiswaContract.MahasiswaEntry.COLUMN_NAMA} = '${mahasiswa.nama}', " +
        "${MahasiswaContract.MahasiswaEntry.COLUMN_NIM} = '${mahasiswa.nim}', " +
        "${MahasiswaContract.MahasiswaEntry.COLUMN_PASSWORD} = '${mahasiswa.password}' " +
        "WHERE ${MahasiswaContract.MahasiswaEntry.COLUMN_EMAIL} = '${mahasiswa.email}'"

    db.execSQL(sql)
    db.close()
}
```

4. Kelas Query (Pada Helper)

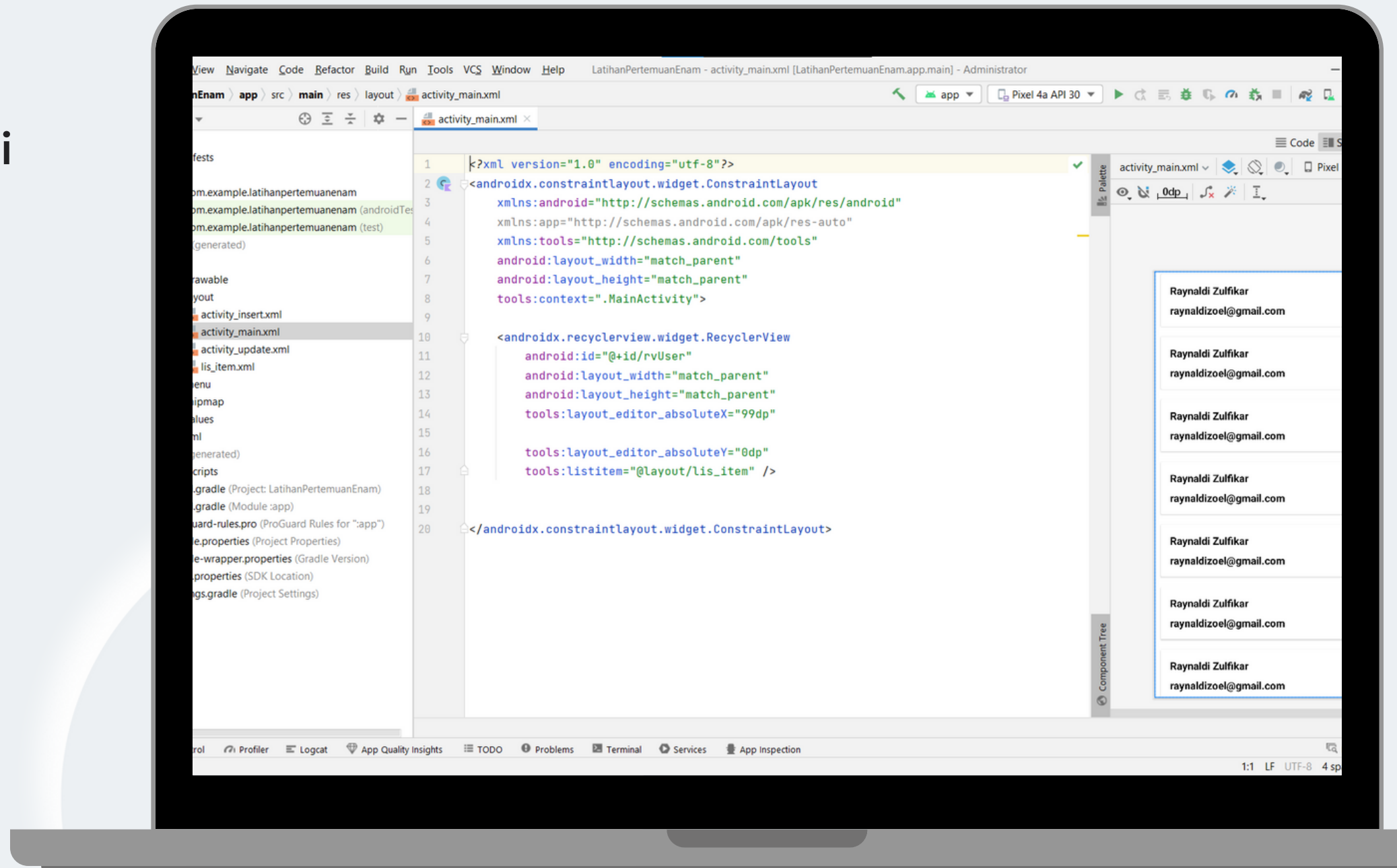


```
// Sqlite  
implementation "androidx.sqlite:sqlite:2.3.1"
```

Video Praktikum

Untuk video praktikum bisa diakses melalui link berikut [ini](#).

Repository Github file project latihan bisa diakses melalui link berikut [ini](#).



**TERIMA
KASIH**