



DIGITAL
TALENT
SCHOLARSHIP



VOCATIONAL SCHOOL GRADUATE ACADEMY

Mobile Programmer

Pertemuan 10 : Database pada Mobile



KOMINFO



#JADIJAGOANDIGITAL

Badan Penelitian dan Pengembangan Sumber Daya Manusia

PROFIL PENGAJAR



Jabatan Akademik / Kepala LAB Prodi Teknik Komputer

Latarbelakang Pendidikan Pengajar

- S1– STMIK Budi Darma (Skripsi : Aps Mobile Kompresi SMS)
- S2– Universitas Putra Indonesia YPTK Padang (Tesis : Aps Mobile Security SMS)

Riwayat Pekerjaan

- Dosen Tetap Politeknik Negeri Medan
- Trainer Pemrograman Java dan Mobile, Networking, Cyber Security, OS Server
- Konsultan Bidang Aplikasi, Networking, Cyber Security dan Server
- CEO PT. Nusa Tirta Teknologi

Sertifikat Kompetensi :

- Program : Senior Programmer (BNSP)
- Networking : Mikrotik, CISCO
- Server : Windows Server, Redhat
- Project : Comptia Project +

Contact Pengajar

Ponsel :-

Email : azanuddin@polmed.ac.id

Deskripsi Pelatihan

Deskripsi Singkat mengenai Topik

Topik ini membahas cara mendesain SQLite database pada aplikasi berbasis mobile dan mendesain database mobile dengan model layer

Tujuan Pelatihan

Setelah pertemuan selesai peserta pelatihan mampu:

1. Menjelaskan konseptual SQLite database pada perangkat basis mobile sesuai dengan buku panduan.
2. Membuat Database dan tabel pada aplikasi berbasis mobile dibuat sesuai dengan kebutuhan.
3. Menjelaskan Konsep dasar structured Query Language dijelaskan sesuai dengan buku panduan
4. Membangun aplikasi mobile dengan menggunakan media penyimpanan SQLite database dengan proses create, read, update dan delete data berjalan sesuai dengan kasus yang diberikan.
5. Membangun Aplikasi berbasis mobile untuk layer input dan retrieve data berbasis sesuai dengan kasus yang diberikan.
6. Membuat Salah satu bahasa pemrograman berbasis web sebagai layer perantara antara layer mobile dengan layer database mysql berdasarkan kasus yang diberikan.
7. Merancang database dengan menggunakan ER-Diagram atau class diagram sesuai dengan kasus yang diberikan.
8. Membuat Database pada perangkat PC dengan menggunakan DBMS (Database Management Software) tertentu sesuai dengan kebutuhan.
9. Mengintegrasikan layer input (aplikasi mobile), layer perantara (aplikasi web) dan database server (layer database pada PC) sesuai dengan kasus yang diberikan.

Spesifikasi Perangkat Jaringan

Materi Yang akan disampaikan:

1. SQLite
2. Perintah Dasar SQL

Tugas :

Tugas Pelatihan

Outcome/Capaian Pelatihan

deskripsi Capaian

SQLite

Sub Materi :

- ❖ Overview SQLite
- ❖ Pengguna SQLite
- ❖ Pembuatan Database dan Tabel SQLite
 - ❖ Menggunakan Database Editor
 - ❖ Menggunakan Pemrograman Berbasis Mobile
- ❖ Structured Query Language di SQLite

SQLite - Overview

Apa SQLite ??

- ❖ Merupakan basis data tertanam yang open source
- ❖ Didesain oleh D. Richard Hipp.
- ❖ Asal mula digunakan untuk menyimpan data yang digunakan pada sistem kendali missile.
- ❖ Tujuannya untuk memudahkan program berjalan tanpa adanya instalasi basis data.

SQLite - Overview

Perkembangan dari SQLite ??

- ❖ Versi 1.0 tahun 2000, menggunakan dasar GDBM (GNU Database Manager)
- ❖ Versi 2.0 tahun 2001.
- ❖ Versi 3.0 tahun 2004, (versi terakhir 3.28.0 release tahun 2019).

SQLite - Pengguna

Pengguna SQLite

- ❖ Adobe – menggunakan SQLite di aplikasi Photoshop dan Acrobat Reader
- ❖ Apple – beberapa fitur apple menggunakannya
 - ❖ Apple mail
 - ❖ Safari web browser
- ❖ Mozilla – Web browser nya menggunakan SQLite
- ❖ Google – Google Desktop dan Google Gears menggunakannya.

SQLite - Pengguna

Pengguna SQLite ...

- ❖ McAfee – menggunakan SQLite di beberapa program antivirusnya
- ❖ PHP – PHP menyisipkan built in SQLite 2 dan SQLite 3 di librarinya
- ❖ Python – SQLite di paketkan dalam Bahasa pemrograman Python

Pembuatan Database SQLite

Pembuatan Database SQLite dapat dilakukan dengan beberapa cara :

- Perangkat Database editor
- Pemrograman berbasis mobile

Pembuatan Database SQLite dengan Editor

- Langkah 1
 - Unduh DBBrowserForSQLite
- Langkah 2
 - Instalasi DBBrowserForSQLLite
- Langkah 3
 - Membuat Database (jika belum ada)
 - Membuka Database (jika sudah ada)
- Langkah 4
 - Membuat Tabel
- Langkah 5
 - Melihat dan menambah data di Table
- Langkah 6
 - (Optional) menghapus data di Tabel
- Langkah 7
 - (Optional) menghapus
 - Tabel

Pembuatan Database SQLite dengan Editor

Langkah 1 :

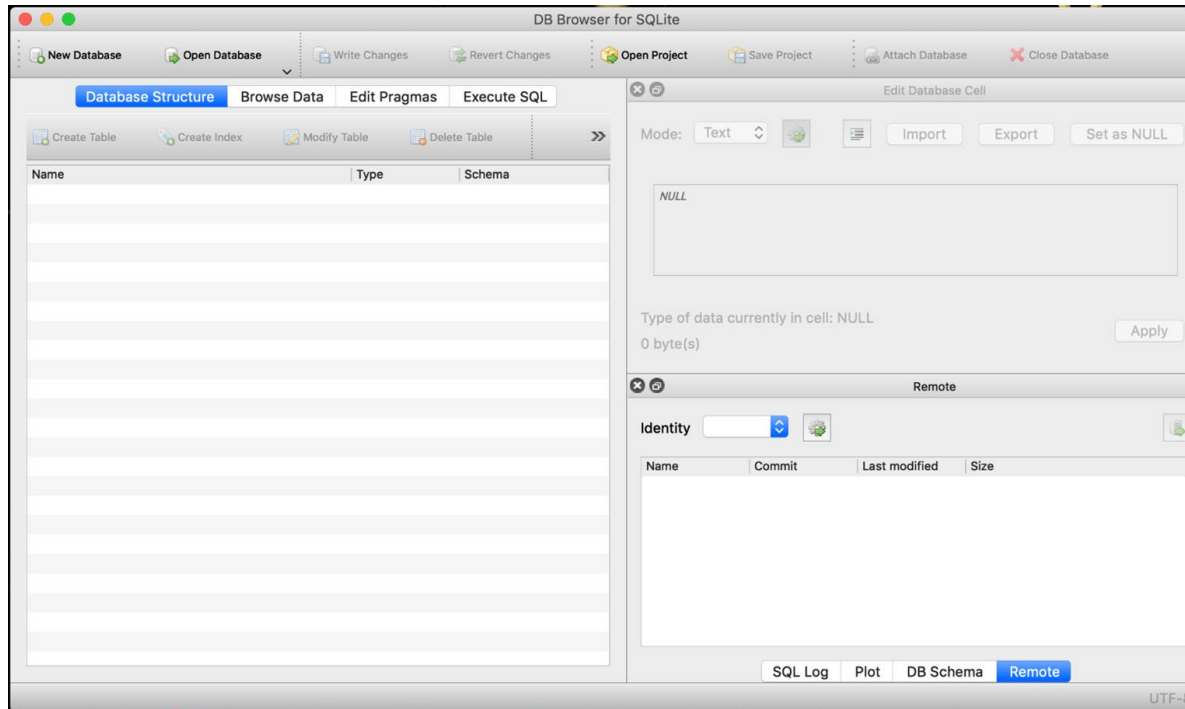
Unduh perangkat lunak dan pilih sesuai dengan sistem operasi yang sesuai di

<https://SQLitebrowser.org/dl/>

Pembuatan Database SQLite dengan Editor

Langkah 2 :

Install perangkat lunak setelah selesai diunduh. Kemudian jalankan aplikasinya.



Pembuatan Database SQLite dengan Editor

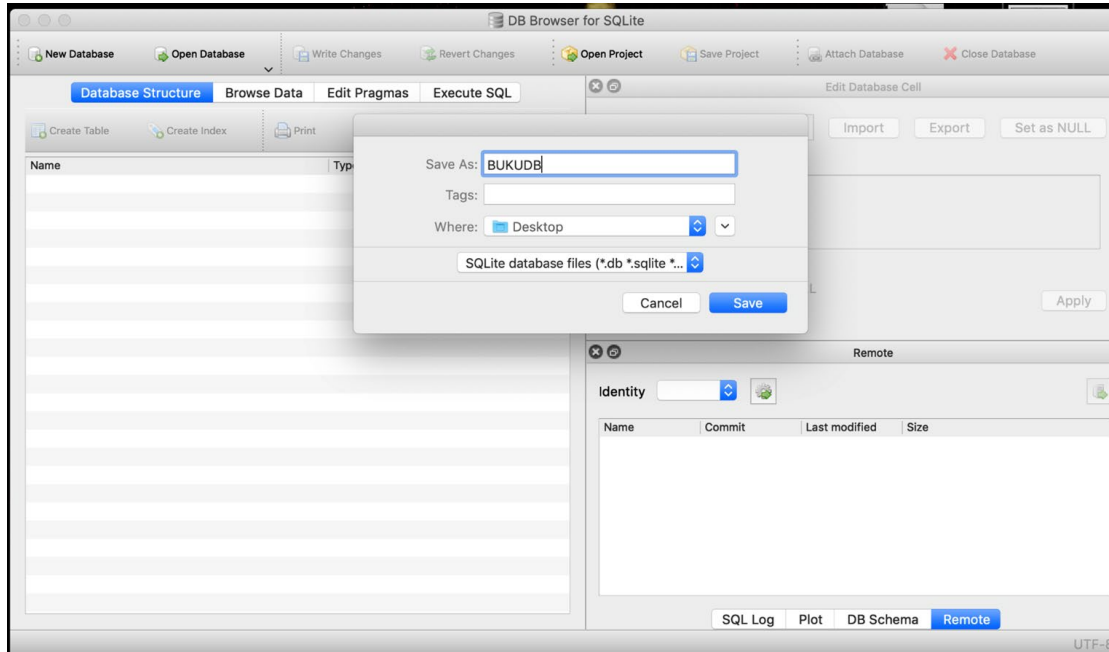
Langkah 3 :

Membuat database (jika masih belum ada).

- Pilih dan tekan tombol “New Database” di pojok kiri atas.
- Kemudian beri nama database sesuai kebutuhan.
- Contoh untuk materi ini adalah “BUKUDB”.
- Pilih tempat penyimpanan database tersebut, misal Desktop
- File akan terbentuk di tempat penyimpanan yang ditentukan

Pembuatan Database SQLite dengan Editor

Hasil Langkah 3 :



Pembuatan Database SQLite dengan Editor

Langkah 3 :

Membuka database (jika sudah ada).

- Pilih dan tekan tombol “Open Database” di pojok kiri atas.
- Kemudian pilih file database. Contoh untuk materi ini adalah ”BUKUDB”.
- Klik Open

Pembuatan Database SQLite dengan Editor

Langkah 4 :

Membuat Tabel yang dibutuhkan.

- Setelah database terbentuk, maka kita bisa membuat table di database tersebut.
- Pilih tombol “Create Table” di sebelah kiri atas. Akan muncul form untuk memasukkan informasi table yang akan dibuat.
- Beri nama table. Di materi ini akan dibuat contoh table BUKU untuk menyimpan data buku.

Pembuatan Database SQLite dengan Editor

Lanjutan Langkah 4 :

- Kemudian tambahkan field sesuai kebutuhan. Untuk contoh table BUKU, field yang dibutuhkan :
 - `_id`, tipe int autoincrement
 - `title`, tipe text
 - `isbn`, tipe text
 - `publisher`, tipe text
- Pada kolom bawah akan tampak definisi dari SQL yang akan dieksekusi
- Tekan tombol OK untuk membuat table

Hasil Langkah 4 : Tabel akan terbentuk seperti pada gambar kanan

#JADIJAGOANDIGITAL

Pembuatan Database SQLite dengan Editor

Langkah 5 :

Melihat dan menambah data di Tabel.

- Setelah tabel terbentuk, maka kita bisa melihat dan menambahkan record ke dalam table.
- Pilih table yang diinginkan, dalam contoh ini adalah “BUKU”, yang ada di daftar tabel
- Kemudian klik kanan pada nama table tersebut. Pilih “Browse Table”.
- Tampak data yang tersimpan dalam table tersebut.

Pembuatan Database SQLite dengan Editor

Lanjutan Langkah 5 :

- Masing – masing kolom bisa langsung diisi nilai sesuai kebutuhan. Contoh untuk materi ini :
 - Kolom _id : dikosongi (karena diisi otomatis oleh sistem)
 - Kolom isbn : 909921129
 - Kolom title : Belajar SQLite
 - Kolom publisher : Jaya Abadi
- Kemudian tekan tombol “New Record” di bagian atas.
- Langkah diatas bisa dilakukan berulang untuk menambah data yang berbeda

Pembuatan Database SQLite dengan Editor

Hasil Langkah 5 : Mengisi 3 record di table BUKU

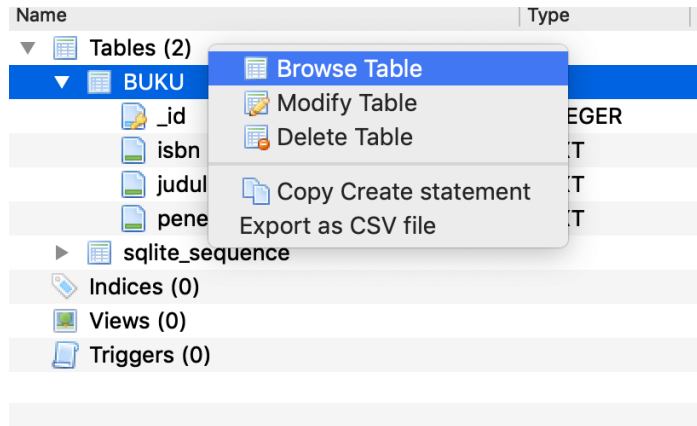


Table: BUKU

New Record Delete Record

	_id	isbn	judul	penerbit
Filter	Filter	Filter	Filter	
1	3	909921122	SQLite Tutorial	Jaya Abadi
2	4	99101010	MySQL	Press
3	6	909921122	SQLite Tutorial	Jaya Abadi

1 - 3 of 3

Go to: 1

Pembuatan Database SQLite dengan Editor

Langkah 6 (pilihan):

- Untuk mengupdate informasi data pada table dapat langsung dilakukan pada baris yang dimaksud.
- Setelah merubah pastikan tekan tombol “Write Changes” di bagian atas.

Table: BUKU

	_id	isbn	judul	penerbit
Filter	Filter	Filter	Filter	Filter
1	3	909921122	SQLite Tutorial	Jaya Abadi
2	4	99101010	MySQL	Press
3	6	909921122	SQLite Tutorial	Jaya Abadi

1 - 3 of 3

Go to: 1



Table: BUKU

	_id	isbn	judul	penerbit
Filter	Filter	Filter	Filter	Filter
1	3	909921122	Belajar SQLite	Jaya Abadi
2	4	99101010	MySQL	Press
3	6	909921122	SQLite Tutorial	Jaya Abadi

1 - 3 of 3

Go to: 1

Pembuatan Database SQLite dengan Editor

Langkah 7 (pilihan):

- Untuk menghapus informasi record pada table dapat langsung dilakukan pada baris yang dimaksud. Kemudian tekan tombol “Delete Record”
- Setelah merubah pastikan tekan tombol “Write Changes” di bagian atas.

Table: BUKU

	_id	isbn	judul	penerbit
Filter	Filter	Filter	Filter	Filter
1	3	909921122	SQLite Tutorial	Jaya Abadi
2	4	99101010	MySQL	Press
3	6	909921122	SQLite Tutorial	Jaya Abadi

Go to: 1



Table: BUKU

1	4	99101010	MySQL	Press
2	6	909921122	SQLite Tutorial	Jaya Abadi

Go to: 1

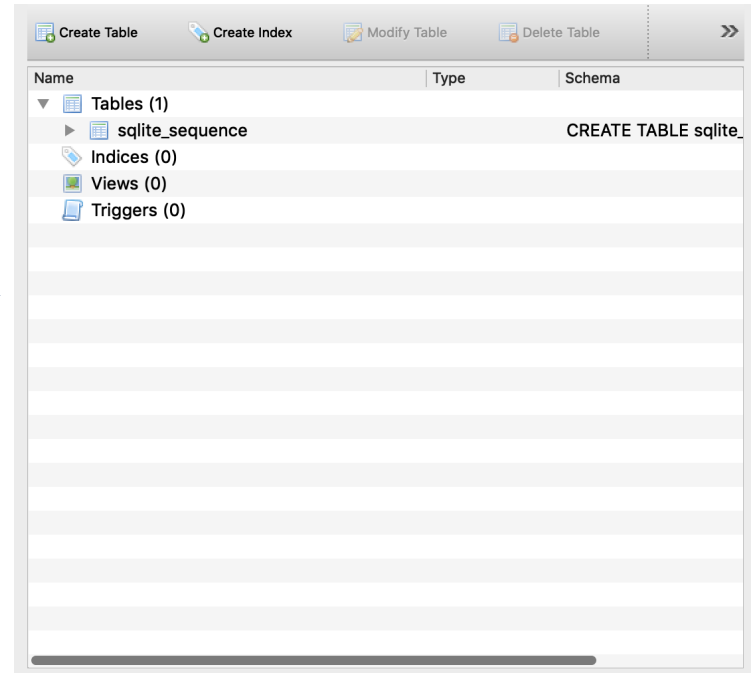
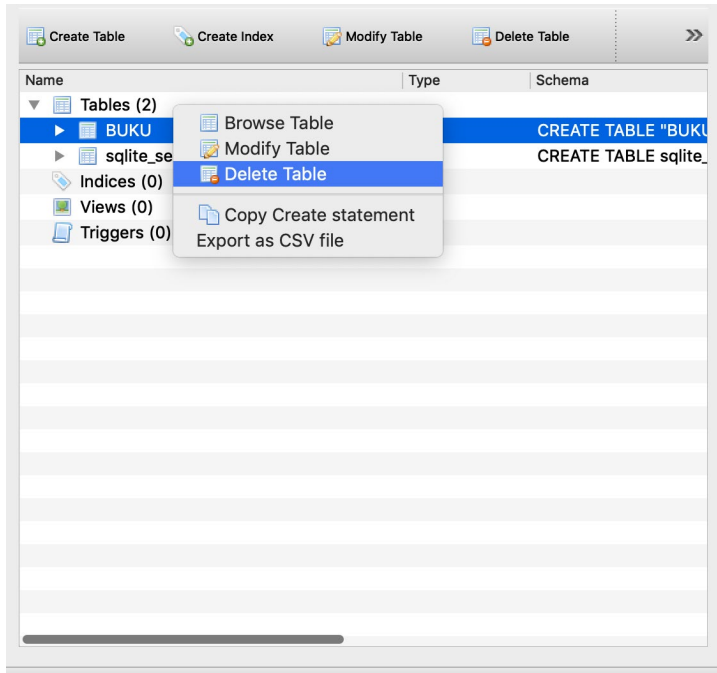
Pembuatan Database SQLite dengan Editor

Langkah 8 (pilihan):

- Untuk menghapus Tabel, dapat dilakukan dengan memilih table yang dimaksud pada daftar table. Kemudian klik kanan pilih “Delete Table”.
- Jika yakin tekan tombol “Yes” pada kotak dialog
- Setelah merubah pastikan tekan tombol “Write Changes” di bagian atas.

Pembuatan Database SQLite dengan Editor

Hasil Langkah 8 :



Pembuatan Database SQLite dengan Program

Di pemrograman Android, untuk membuka atau membuat database SQLite menggunakan fungsi berikut :

```
SQLiteDatabase.openDatabase( myDbPath,  
    null,  
    SQLiteDatabase.CREATE_IF_NECESSARY);
```

Parameter path

Parameter factory

Parameter flags

Pembuatan Database SQLite dengan Program

```
package cis470.matos.sqldatabases;
public class MainActivity extends Activity {
    SQLiteDatabase db;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView txtMsg = (TextView) findViewById(R.id.txtMsg);

        // path to the external SD card (something like: /storage/sdcard/...)
        // String storagePath = Environment.getExternalStorageDirectory().getPath();

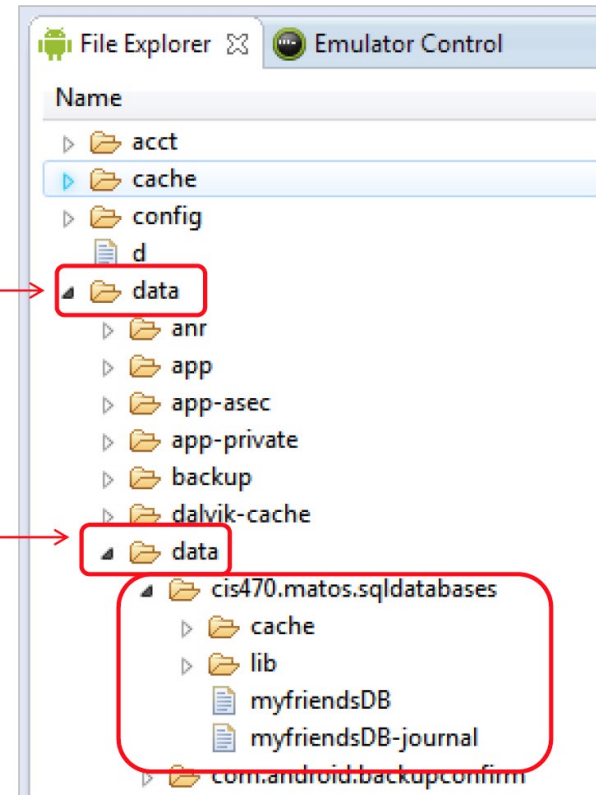
        // path to internal memory file system (data/data/cis470.matos.databases)
        File storagePath = getApplication().getFilesDir();

        String myDbPath = storagePath + "/" + "myfriends";

        txtMsg.setText("DB Path: " + myDbPath);
        try {
            db = SQLiteDatabase.openDatabase(myDbPath, null,
                                           SQLiteDatabase.CREATE_IF_NECESSARY);

            // here you do something with your database ...
            db.close();
            txtMsg.append("\nAll done!");
        } catch (SQLException e) {
            txtMsg.append("\nERROR " + e.getMessage());
        }
    } // onCreate
} // class
```

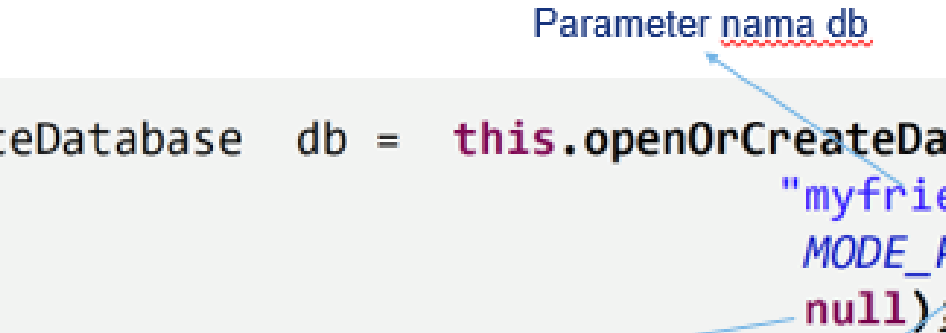
13 -6



Pembuatan Database SQLite dengan Program

Alternatif lain untuk membuat / membuka SQLite dengan menggunakan fungsi berikut :

```
SQLiteDatabase db = this.openOrCreateDatabase(  
    "myfriendsDB",  
    MODE_PRIVATE,  
    null);
```



Parameter factory

Parameter mode

Pembuatan Database SQLite dengan Program

Tipe perintah SQL di android terdiri dari 2 kategori :

- Queri aksi (Action Queries)
- Queri Retrieval (Retrieval Queries)

Pembuatan Database SQLite dengan Program

Membuat dan mengisi table menggunakan *action queries*

➤ Fungsi `execSQL()`

```
db.execSQL("create table tblAMIGO ("
    + " recID integer PRIMARY KEY autoincrement, "
    + " name text, "
    + " phone text ); " );

db.execSQL( "insert into tblAMIGO(name, phone) values ('AAA', '555-1111');" );
db.execSQL( "insert into tblAMIGO(name, phone) values ('BBB', '555-2222');" );
db.execSQL( "insert into tblAMIGO(name, phone) values ('CCC', '555-3333');" );
```

Pembuatan Database SQLite dengan Program

Untuk menjalankan query SELECT, fungsi `rawQuery()` bisa digunakan dengan beberapa pendekatan :

- Tanpa parameter
- Menggunakan parameter

Pembuatan Database SQLite dengan Program

Untuk menjalankan query SELECT dengan menggunakan fungsi `rawQuery()` tanpa parameter

```
Cursor c1 = db.rawQuery("select * from tblAMIGO", null);
```



Parameter SQL Statement

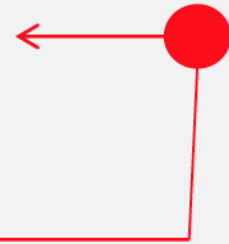
Pembuatan Database SQLite dengan Program

Untuk menjalankan query SELECT dengan menggunakan fungsi `rawQuery()` dengan parameter

```
String mySQL = "select count(*) as Total "  
               + " from tblAmigo "  
               + " where recID > ? "  
               + "    and name  = ? ";
```

```
String[] args = {"1", "BBB"};
```

```
Cursor c1 = db.rawQuery(mySQL, args);
```



Pembuatan Database SQLite dengan Program

SQL Cursor :

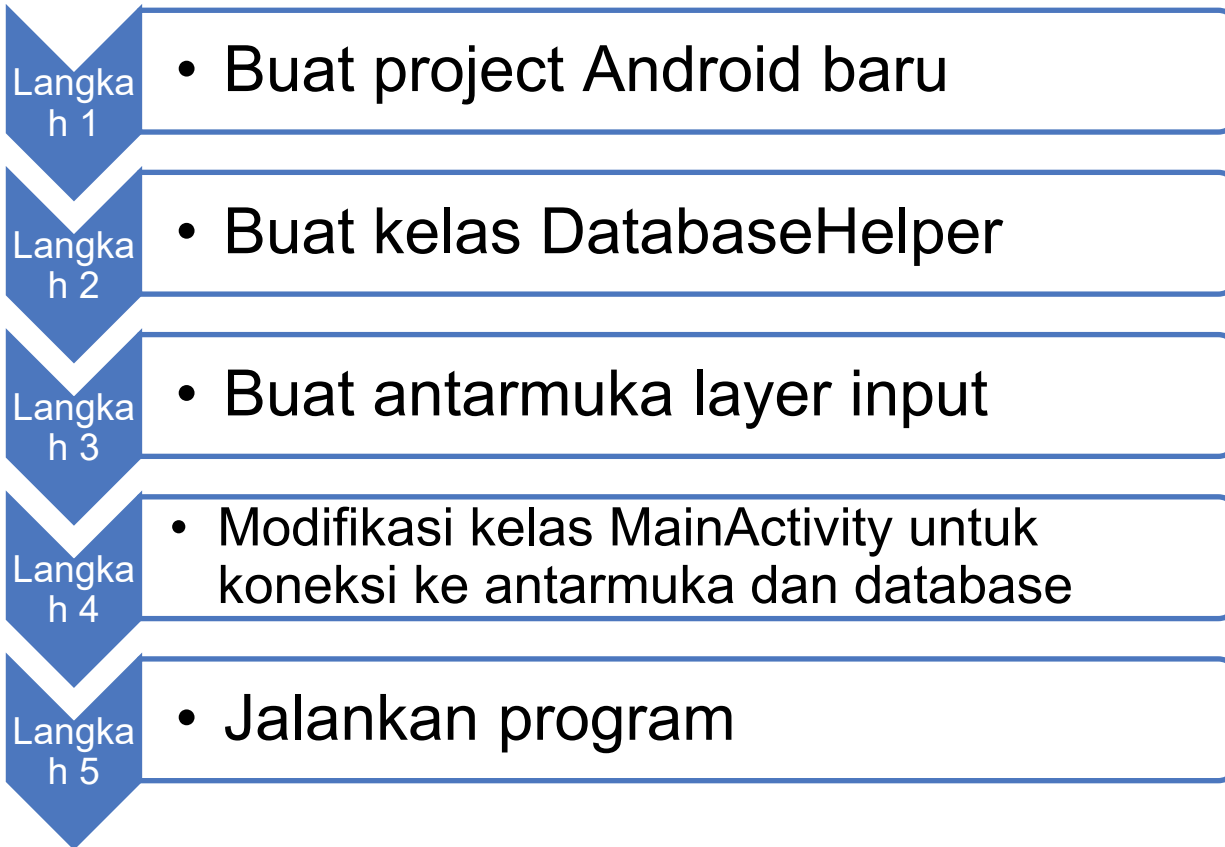
- **Pengenali Posisi:** isFirst(), isLast(), isBeforeFirst(), isAfterLast().
- **Navigasi Record:** moveToFirst(), moveToLast(), moveToNext(), moveToPrevious(), move(n).
- **Pengambilan Informasi Field:** getInt, getString, getFloat, getBlob, getDouble, etc.
- **Inspeksi Skema:** getColumnName(), getColumnNames(), getColumnIndex(), getColumnCount(), getCount().

Pembuatan Database SQLite dengan Program

Untuk mengiterasi hasil query obyek Cursor dapat dilakukan dengan contoh berikut :

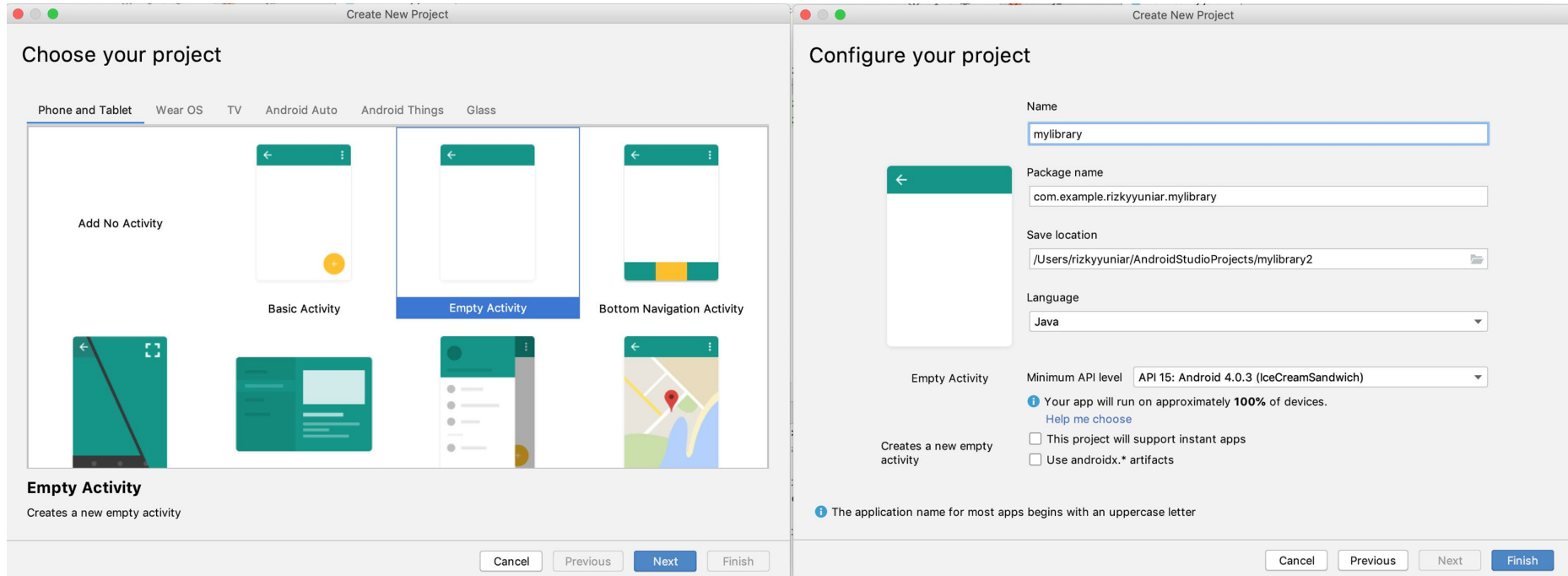
```
String sql = "select * from tblAmigo";  
1 → Cursor c1 = db.rawQuery(sql, null);  
  
c1.moveToPosition(-1);  
2 → while ( c1.moveToNext() ){  
    int recId = c1.getInt(0);  
    String name = c1.getString(1);  
    String phone = c1.getString(c1.getColumnIndex("phone"));  
  
    // do something with the record here...  
}
```

Pembuatan Database SQLite dengan Program



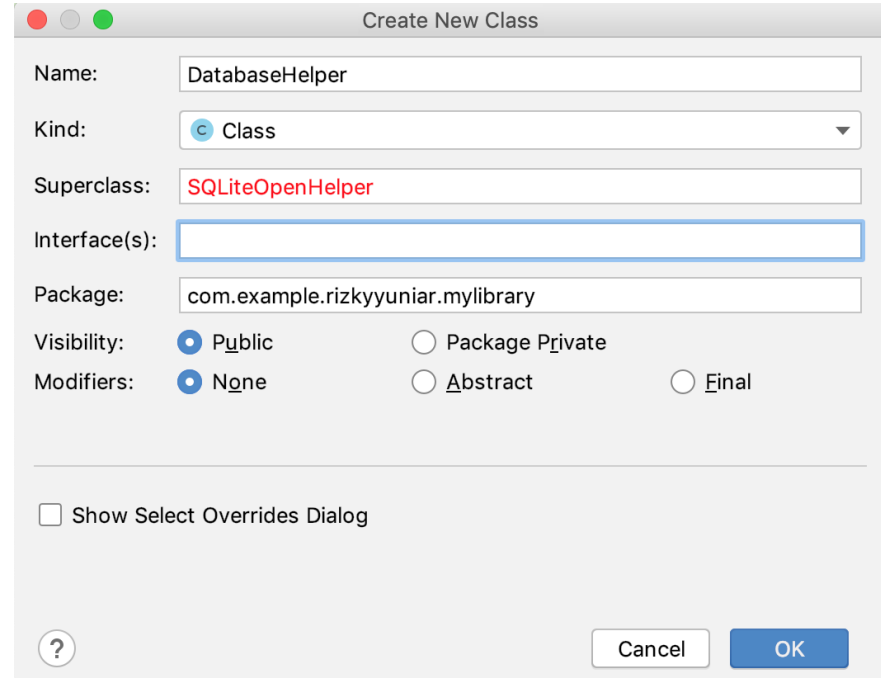
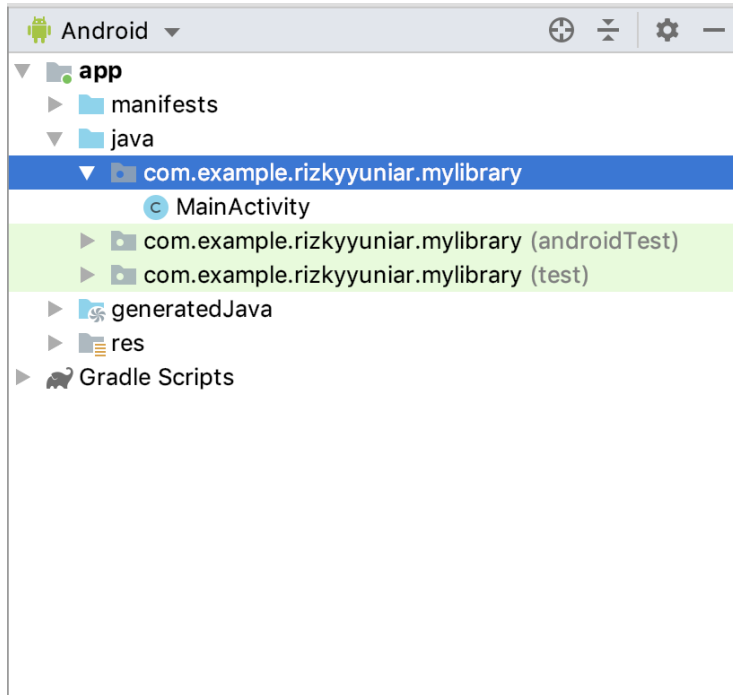
Pembuatan Database SQLite dengan Program

Langkah 1 : Buat project Android baru



Pembuatan Database SQLite dengan Program

Langkah 2 : Buat kelas DatabaseHelper



Pembuatan Database SQLite dengan Program

Langkah 2, lanjutan : Buat kelas DatabaseHelper

1

```
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.util.Log;  
import java.util.ArrayList;
```

2

```
public static String DATABASE_NAME = "student_database";  
private static final int DATABASE_VERSION = 1;  
private static final String TABLE_STUDENTS = "students";  
private static final String KEY_ID = "id";  
private static final String KEY_FIRSTNAME = "name";  
  
private static final String CREATE_TABLE_STUDENTS = "CREATE TABLE "  
    + TABLE_STUDENTS + "(" + KEY_ID  
    + " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_FIRSTNAME + " TEXT );";
```


Pembuatan Database SQLite dengan Program

Langkah 2, lanjutan : Buat kelas DatabaseHelper

3

```
public DatabaseHelper(Context context) {  
    super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);  
  
    Log.d( tag: "table", CREATE_TABLE_STUDENTS);  
}
```

4

```
@Override  
public void onCreate(SQLiteDatabase db) {  
    db.execSQL(CREATE_TABLE_STUDENTS);  
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    db.execSQL("DROP TABLE IF EXISTS '" + TABLE_STUDENTS + "'");  
    onCreate(db);  
}
```

Pembuatan Database SQLite dengan Program

Langkah 2, lanjutan : Buat kelas DatabaseHelper

```
5 public long addStudentDetail(String student) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    // Creating content values  
    ContentValues values = new ContentValues();  
    values.put(KEY_FIRSTNAME, student);  
    // insert row in students table  
    long insert = db.insert(TABLE_STUDENTS,  
        nullColumnHack: null, values);  
  
    return insert;  
}
```

Pembuatan Database SQLite dengan Program

Langkah 2, lanjutan : Buat kelas DatabaseHelper

```
6 public ArrayList<String> getAllStudentsList() {  
    ArrayList<String> studentsArrayList = new ArrayList<String>();  
    String name="";  
    String selectQuery = "SELECT * FROM " + TABLE_STUDENTS;  
    SQLiteDatabase db = this.getReadableDatabase();  
    Cursor c = db.rawQuery(selectQuery, selectionArgs: null);  
    // looping through all rows and adding to list  
    if (c.moveToFirst()) {  
        do {  
            name = c.getString(c.getColumnIndex(KEY_FIRSTNAME));  
            // adding to Students list  
            studentsArrayList.add(name);  
        } while (c.moveToNext());  
        Log.d( tag: "array", studentsArrayList.toString());  
    }  
    return studentsArrayList;  
}
```

Pembuatan Database SQLite dengan Program

Langkah 3: Modifikasi layout sesuai dengan tampilan berikut



TextView

EditText, dengan id = **etname**

Button, dengan id = **btnStore**

Button, dengan id = **btnget**

TextView, dengan id = **tvnames**

Layout -> LinearLayout

Pembuatan Database SQLite dengan Program

Langkah 4 : Modifikasi kelas MainActivity

```
1  import android.support.v7.app.AppCompatActivity;
   import android.os.Bundle;
   import android.view.View;
   import android.widget.Button;
   import android.widget.EditText;
   import android.widget.TextView;
   import android.widget.Toast;
   import java.util.ArrayList;

2      private Button btnStore, btnGetall;
      private EditText etname;
      private DatabaseHelper databaseHelper;
      private TextView tvnames;
      private ArrayList<String> arrayList;
```

Pembuatan Database SQLite dengan Program

3

Langkah 4, lanjutan :

Modifikasi kelas MainActivity

```
databaseHelper = new DatabaseHelper(context: this);
tvnames = (TextView) findViewById(R.id.tvnames);

btnStore = (Button) findViewById(R.id.btnstore);
btnGetall = (Button) findViewById(R.id.btnget);
etname = (EditText) findViewById(R.id.etname);

btnStore.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        databaseHelper.addStudentDetail(etname.getText().toString());
        etname.setText("");
        Toast.makeText(context: MainActivity.this,
            text: "Stored Successfully!", Toast.LENGTH_SHORT).show();
    }
});

btnGetall.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        arrayList = databaseHelper.getAllStudentsList();
        tvnames.setText("");
        for (int i = 0; i < arrayList.size(); i++){
            tvnames.setText(tvnames.getText().toString()+" "+arrayList.get(i));
        }
    }
});
```

Pembuatan Database SQLite dengan Program

Langkah 5 : Jalankan program

The screenshot shows an Android application interface with a light gray background. At the top, the text "Enter Name to store in SQLite" is displayed. Below this is a text input field with the placeholder "Enter Name". To the right of the input field is a large black number "1". Below the input field is a gray button labeled "STORE". To the right of this button is a large black number "2". Below the "STORE" button is another gray button labeled "GET ALL NAMES FROM SQLITE". To the right of this button is a large black number "3". Below the "GET ALL NAMES FROM SQLITE" button, the text ", anton, desi, gusi" is displayed. To the right of this text is a large black number "4". At the bottom of the screen is a black navigation bar with three white icons: a triangle, a circle, and a square.

Structured Query Language

Sub Materi :

- Pengenalan SQL
- Konsep Dasar SQL

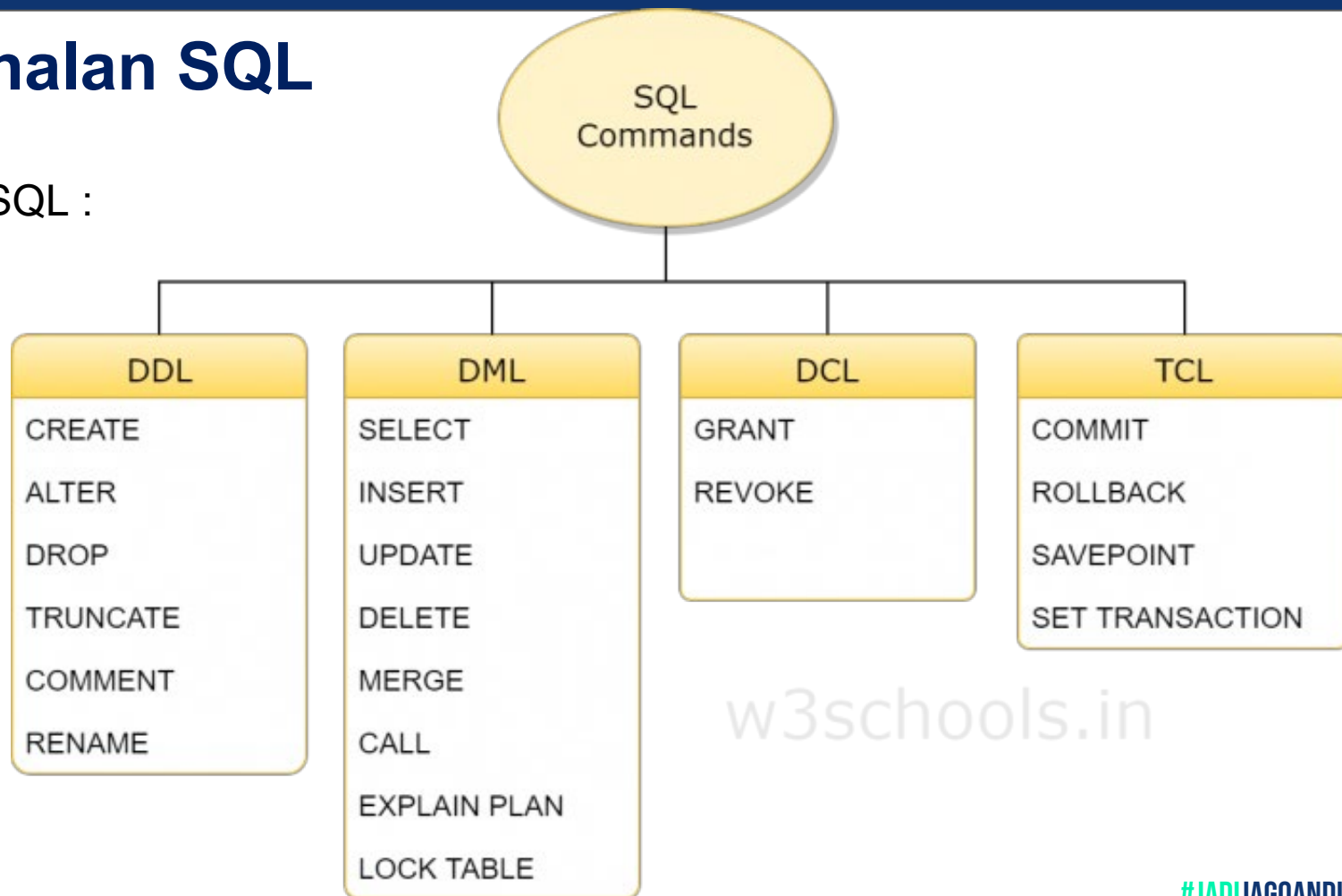
Pengenalan SQL

Apa itu SQL?

- ❖ Structured Query Language
- ❖ Bahasa standar untuk menyimpan, memanipulasi dan menerima data di basis data
- ❖ Memiliki 4 kategori fungsi
 - ❖ Data Definition Language (DDL)
 - ❖ Data Manipulation Language (DML)
 - ❖ Data Control Language (DCL)
 - ❖ Transaction Control Language (TCL)

Pengenalan SQL

Perintah SQL :



DDL – Create Table di SQLite

- ❖ Berikut adalah sintaks dasar untuk membuat table di basis data

```
CREATE TABLE table_name
(
    column_name column_type,
    [...]
);
```

- ❖ Berikut adalah contoh untuk membuat table BUKU di basis data

```
CREATE TABLE "BUKU" (
    "_id" INTEGER PRIMARY KEY AUTOINCREMENT,
    "isbn" TEXT,
    "title" TEXT,
    "publisher" TEXT
);
```

DDL – Alter Table di SQLite

- ❖ Berikut adalah sintaks dasar untuk mengubah table di basis data

```
ALTER TABLE database_name.table_name RENAME TO new_table_name;  
ALTER TABLE database_name.table_name ADD COLUMN column_def...;
```

DDL – Delete Table di SQLite

- ❖ Berikut adalah sintaks dasar untuk menghapus table di basis data

```
DROP TABLE table_name;
```

DML – Insert di SQLite

- ❖ Berikut adalah sintaks dasar untuk menyisipkan baris pada table di basis data

```
INSERT INTO table_name (column_name [, ...]) VALUES (new_value [, ...]);
```

```
INSERT INTO table_name VALUES (new_value [, ...]);
```

- ❖ Contoh Insert SQL :

```
INSERT INTO parts ( name, stock, status ) VALUES ( 'Widget', 17, 'IN STOCK' );
```

DML – Update di SQLite

- ❖ Berikut adalah sintaks dasar untuk mengupdate baris pada table di basis data

```
UPDATE table_name SET column_name=new_value [, ...] WHERE expression
```

- ❖ Contoh Update SQL :

```
-- Update the price and stock of part_id 454:
```

```
UPDATE parts SET price = 4.25, stock = 75 WHERE part_id = 454;
```

DML – Delete di SQLite

- ❖ Berikut adalah sintaks dasar untuk mengupdate baris pada table di basis data

```
DELETE FROM table_name WHERE expression;
```

- ❖ Contoh Delete SQL :

```
-- Delete the row with rowid 385:  
DELETE FROM parts WHERE part_id = 385;
```

```
-- Delete all rows with a rowid greater than or equal to 43  
-- and less than or equal to 246:  
DELETE FROM parts WHERE part_id >= 43 AND part_id <= 246;
```


DML – Perintah SELECT di SQLite

- ❖ Berikut adalah sintaks umum perintah SELECT di basis data SQLite

```
SELECT [DISTINCT] select_heading
FROM source_tables
WHERE filter_expression
GROUP BY grouping_expressions
        HAVING filter_expression
ORDER BY ordering_expressions
LIMIT count
        OFFSET count
```

Tugas SQL – Contoh SQL

❖ Jalankan perintah SQL berikut untuk membuat Tabel x, y dan z

```
CREATE TABLE x ( a, b );  
INSERT INTO x VALUES ( 1, 'Alice' );  
INSERT INTO x VALUES ( 2, 'Bob' );  
INSERT INTO x VALUES ( 3, 'Charlie' );
```

```
CREATE TABLE y ( c, d );  
INSERT INTO y VALUES ( 1, 3.14159 );  
INSERT INTO y VALUES ( 1, 2.71828 );  
INSERT INTO y VALUES ( 2, 1.61803 );
```

```
CREATE TABLE z ( a, e );  
INSERT INTO z VALUES ( 1, 100 );  
INSERT INTO z VALUES ( 1, 150 );  
INSERT INTO z VALUES ( 3, 300 );  
INSERT INTO z VALUES ( 9, 900 );
```

Tugas SQL – Contoh SQL

❖ Jalankan perintah SQL `SELECT` Sederhana

```
sqlite> SELECT * FROM x;
```

a	b
-----	-----
1	Alice
2	Bob
3	Charlie

```
sqlite> SELECT d, d*d AS dSquared FROM y;
```

d	dSquared
-----	-----
3.14159	9.8695877281
2.71828	7.3890461584
1.61803	2.6180210809

Tugas SQL – Contoh SQL

- ❖ Jalankan perintah SQL
- ❖ SELECT dengan JOIN

```
sqlite> SELECT * FROM x JOIN y;  
sqlite> SELECT * FROM x CROSS JOIN y;  
sqlite> SELECT * FROM x, y;
```

a	b	c	d
-----	-----	-----	-----
1	Alice	1	3.14159
1	Alice	1	2.71828
1	Alice	2	1.61803
2	Bob	1	3.14159
2	Bob	1	2.71828
2	Bob	2	1.61803
3	Charlie	1	3.14159
3	Charlie	1	2.71828
3	Charlie	2	1.61803

Tugas SQL – Contoh SQL

❖ Jalankan perintah SQL `SELECT` dengan `JOIN...ON`

```
sqlite> SELECT * FROM x JOIN y ON a = c;
```

a	b	c	d
-----	-----	-----	-----
1	Alice	1	3.14159
1	Alice	1	2.71828
2	Bob	2	1.61803

Tugas SQL – Contoh SQL

- ❖ Jalankan perintah SQL `SELECT` dengan `JOIN...USING`, `NATURAL JOIN`

```
sqlite> SELECT * FROM x JOIN z USING ( a );  
sqlite> SELECT * FROM x NATURAL JOIN z;
```

a	b	e
-----	-----	-----
1	Alice	100
1	Alice	150
3	Charlie	300

Tugas SQL – Contoh SQL

- ❖ Jalankan perintah SQL `SELECT` dengan `OUTER JOIN`

```
sqlite> SELECT * FROM x LEFT OUTER JOIN z USING ( a );
```

a	b	e
-----	-----	-----
1	Alice	100
1	Alice	150
2	Bob	[NULL]
3	Charlie	300

Tugas SQL – Contoh SQL

❖ Jalankan perintah SQL `SELECT` dengan `COMPOUND JOIN`

```
sqlite> SELECT * FROM x JOIN y ON x.a = y.c LEFT OUTER JOIN z ON y.c = z.a;
```

a	b	c	d	a	e
-----	-----	-----	-----	-----	-----
1	Alice	1	3.14159	1	100
1	Alice	1	3.14159	1	150
1	Alice	1	2.71828	1	100
1	Alice	1	2.71828	1	150
2	Bob	2	1.61803	[NULL]	[NULL]

Tugas SQL – Contoh SQL

❖ Jalankan perintah SQL `SELECT` dengan `WHERE` berikut

Contoh 1 `sqlite> SELECT * FROM y WHERE d BETWEEN 1.0 AND 3.0;`

c	d
-----	-----
1	2.71828
2	1.61803

Contoh 2 `sqlite> SELECT c, d, c+d AS sum FROM y WHERE sum < 4.0;`

c	d	sum
-----	-----	-----
1	2.71828	3.71828
2	1.61803	3.61803

Tugas SQL – Contoh SQL

❖ Jalankan perintah SQL `SELECT` dengan `GROUP BY` berikut

Contoh 1 `sqlite> SELECT a, count(a) AS count FROM z GROUP BY a;`

a	count
-----	-----
1	2
3	1
9	1

Contoh 2

```
sqlite> SELECT a, sum(e), count(e),
...>      sum(e)/count(e) AS expr, avg(e) AS agg
...>      FROM z GROUP BY a;
```

a	sum(e)	count(e)	expr	agg
-----	-----	-----	-----	-----
1	250	2	125	125.0
3	300	1	300	300.0
9	900	1	900	900.0

Tugas SQL – Contoh SQL

- ❖ Jalankan perintah SQL `SELECT` dengan `ORDER BY` berikut

```
sqlite> SELECT * FROM y ORDER BY d;
```

c	d
-----	-----
2	1.61803
1	2.71828
1	3.14159

Kesimpulan

1. Desain database SQLite dapat menggunakan pendekatan perangkat editor maupun pemrograman berbasis mobile
1. Aplikasi berbasis mobile dapat menggunakan SQLite sebagai tempat penyimpanan data.

Referensi

1. <https://demonuts.com/SQLite-android/>
2. Owens, Michael (2006). The Definitive Guide to SQLite.
3. <https://www.SQLite.org>

Tim Penyusun

- Alif Akbar Fitrawan, S.Pd, M. Kom (Politeknik Negeri Banyuwangi);
- Anwar, S.Si, MCs. (Politeknik Negeri Lhokseumawe);
- Eddo Fajar Nugroho (BPPTIK Cikarang);
- Eddy Tungadi, S.T., M.T. (Politeknik Negeri Ujung Pandang);
- Fitri Wibowo (Politeknik Negeri Pontianak);
- Ghifari Munawar (Politeknik Negeri Bandung);
- Hetty Meileni, S.Kom., M.T. (Politeknik Negeri Sriwijaya) ;
- I Wayan Candra Winetra, S.Kom., M.Kom (Politeknik Negeri Bali) ;
- Irkham Huda (Vokasi UGM) ;
- Josseano Amakora Koli Parera, S.Kom., M.T. (Politeknik Negeri Ambon) ;
- I Komang Sugiartha, S.Kom., MMSI (Universitas Gunadarma) ;
- Lucia Sri Istiyowati, M.Kom (Institut Perbanas) ;
- Maksy Sendiang, ST, MIT (Politeknik Negeri Manado) ;
- Medi Noviana (Universitas Gunadarma) ;
- Muhammad Nashrullah (Politeknik Negeri Batam) ;
- Nat. I Made Wiryana, S.Si., S.Kom., M.Sc. (Universitas Gunadarma) ;
- Rika Idmayanti, ST, M.Kom (Politeknik Negeri Padang) ;
- Rizky Yuniar Hakkun (Politeknik Elektronik Negeri Surabaya) ;
- Robinson A.Wadu, ST., MT (Politeknik Negeri Kupang) ;
- Roslina. M.IT (Politeknik Negeri Medan) ;
- Sukamto, SKom., MT. (Politeknik Negeri Semarang) ;
- Syamsi Dwi Cahya, M.Kom. (Politeknik Negeri Jakarta) ;
- Syamsul Arifin, S.Kom, M.Cs (Politeknik Negeri Jember) ;
- Usmanudin (Universitas Gunadarma) ;
- Wandy Alifha Saputra (Politeknik Negeri Banjarmasin) ;

#JADIJAGOANDIGITAL TERIMA KASIH



digitalent.kominfo



DTS_kominfo



digitalent.kominfo



digital talent scholarship