

## **BAB VIII**

### **SQLITE DATABASE**

#### **8.1 Tujuan Pembelajaran**

- a. Mahasiswa dapat mengerti apa yang dimaksud *SQLite*
- b. Mahasiswa dapat mengetahui konsep *SQLite*
- c. Mahasiswa dapat menerapkan *SQLite* kedalam Aplikasi

#### **8.2 Software yang dibutuhkan**

- a. Java JDK
- b. Android Studio 4.2
- c. SDK API 27

#### **8.3 SQLite**

SQLite adalah sebuah engine database SQL yang langsung tertanam atau pada aplikasi. Tidak seperti kebanyakan database SQL lainnya, SQLite tidak memiliki server yang terpisah dari aplikasi. SQLite membaca dan menulis langsung ke file disk biasa. Database SQLite memiliki fitur lengkap dengan banyak tabel, indexes, trigger, dan tampilan, serta tersimpan pada satu file tunggal dalam hard-disk. Format file databasenya bersifat crossplatform. Sehingga kita dapat dengan bebas menyalin database antara sistem 32-bit dan 64bit atau antara arsitektur yang berbeda platform. Fitur-fitur ini membuat SQLite menjadi pilihan populer sebagai Application File Format.

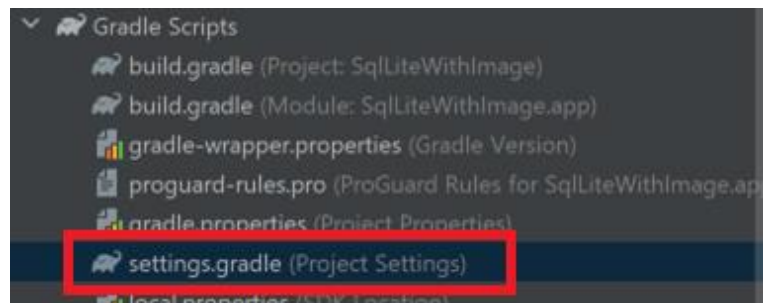
SQLite telah didukung oleh banyak bahasa pemrograman seperti C, C ++, BASIC, C#, Python, Java dan Delphi. Dukungan dari COM (ActiveX) membuat SQLite dapat diakses ke bahasa scripted di Windows seperti VB Script dan JavaScript, sehingga akan menambah kemampuan untuk aplikasi HTML. Dukungan terhadap database ini juga tersedia di sistem operasi embedded seperti iOS, Android, Symbian OS, Maemo, Blackberry dan WebOS karena ukurannya yang kecil dan mudah digunakan. Berikut ini beberapa keunggulan SQLite :

1. SQLite tidak memerlukan proses atau sistem server yang terpisah untuk beroperasi (Serverless).
2. SQLite hadir dengan zero-configuration, yang berarti tidak ada setup atau administrasi yang dibutuhkan.
3. Database SQLite yang lengkap disimpan dalam file tunggal yang tersimpan dalam disk serta bersifat cross-platform.

4. SQLite sangat kecil dan ringan, kurang dari 400KiB untuk konfigurasi lengkap atau kurang dari 250KiB dengan fitur opsional dihilangkan.
5. SQLite bersifat mandiri, yang berarti tidak ada dependensi eksternal.
6. Transaksi SQLite sepenuhnya sesuai dengan ACID, memungkinkan akses yang aman dari banyak proses.
7. SQLite mendukung sebagian besar fitur bahasa query yang ditemukan dalam standar SQL92 (SQL2).
8. SQLite ditulis dalam ANSI-C dan menyediakan API yang sederhana dan mudah digunakan.
9. SQLite tersedia di semua sistem operasi baik ini UNIX (Linux, Mac OS-X, Android, iOS) dan Windows (Win32, WinCE, WinRT).

#### 8.4 Langkah – langkah praktikum

1. Buatlah project baru dengan “Empty Activity” sebagai starter seperti pada praktikum sebelumnya
2. Langkah pertama double klik pada settings.gradle



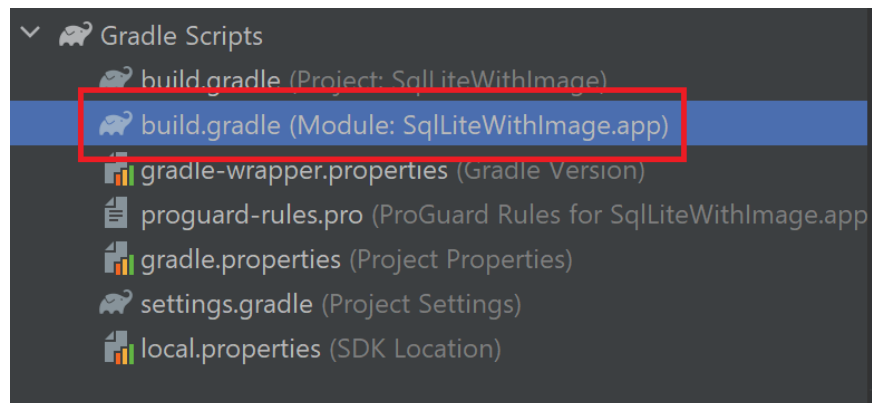
3. Kemudian tambahkan jcenter(), lalu comment //mavenCentral() seperti pada gambar dibawah ini

```
pluginManagement { PluginManagementSpec it ->
    repositories { RepositoryHandler it ->
        gradlePluginPortal()
        google()
        jcenter()
        //mavenCentral()
    }
}

dependencyResolutionManagement { DependencyResolutionManagement it ->
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories { RepositoryHandler it ->
        google()
        jcenter()
        //mavenCentral()
    }
}

rootProject.name = "SQLiteWithImage"
include ':app'
```

4. Kemudian double klik pada build.gradle



5. Kemudian tambahkan library seperti gambar dibawah ini lalu klik “Sync Now” pada pop up gradle yang akan muncul.

```
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
buildFeatures{
    viewBinding true
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'

    //crop image
    api 'com.theartofdev.edmodo:android-image-cropper:2.8.0'
    //loading image in image view
    implementation 'com.squareup.picasso:picasso:2.5.2'
}
```

#### Source Code

```
//activation binding view
buildFeatures{
    viewBinding true
}
//crop image
api 'com.theartofdev.edmodo:android-image-cropper:2.8.0'
//loading image in image view
implementation 'com.squareup.picasso:picasso:2.5.2'
```

6. Kemudian double file AndroidManifest.xml, lalu tambahkan kode seperti gambar dibawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

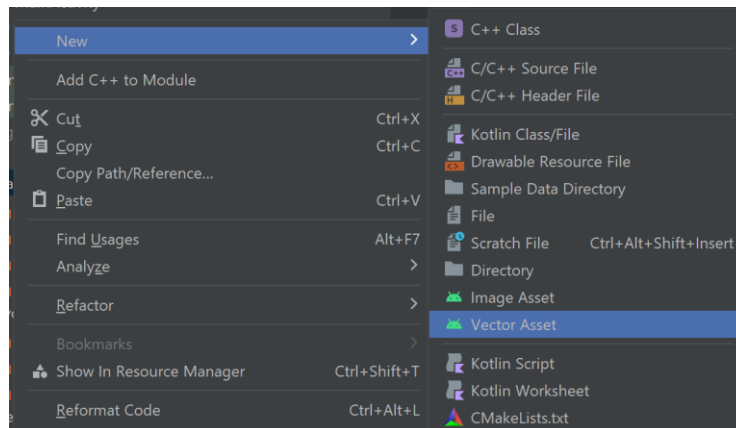
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="SqlLiteWithImage"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SqlLiteWithImage"
        tools:targetApi="31">
        <activity android:name=".DisplayData" android:exported="false">
            <meta-data android:name="android.app.lib_name" android:value="" />
        </activity>
        <activity android:name=".MainActivity" android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <meta-data android:name="android.app.lib_name" android:value="" />
        </activity>
        <activity android:name="com.theartofdev.edmodo.cropper.CropImageActivity" />
    </application>
</manifest>
```

### Source Code

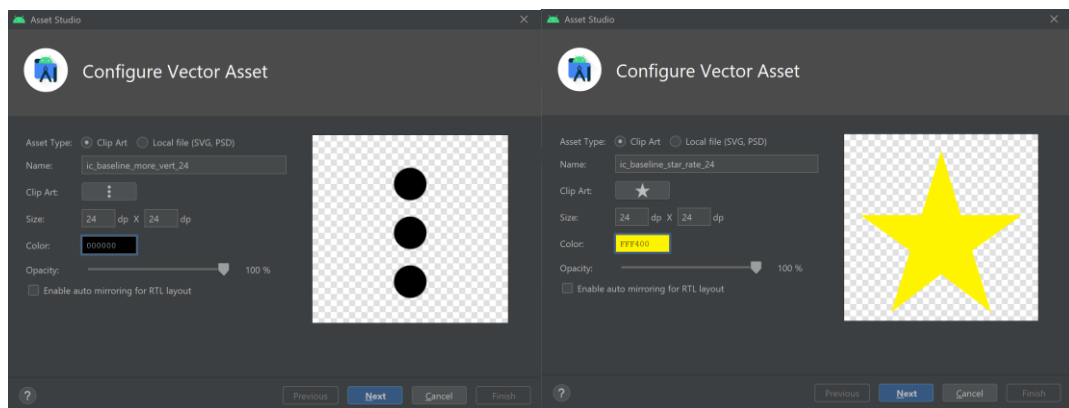
```
//Activation permission camera, read, and write external storage
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

//Activation library cropper
<activity
android:name="com.theartofdev.edmodo.cropper.CropImageActivity"
/>
```

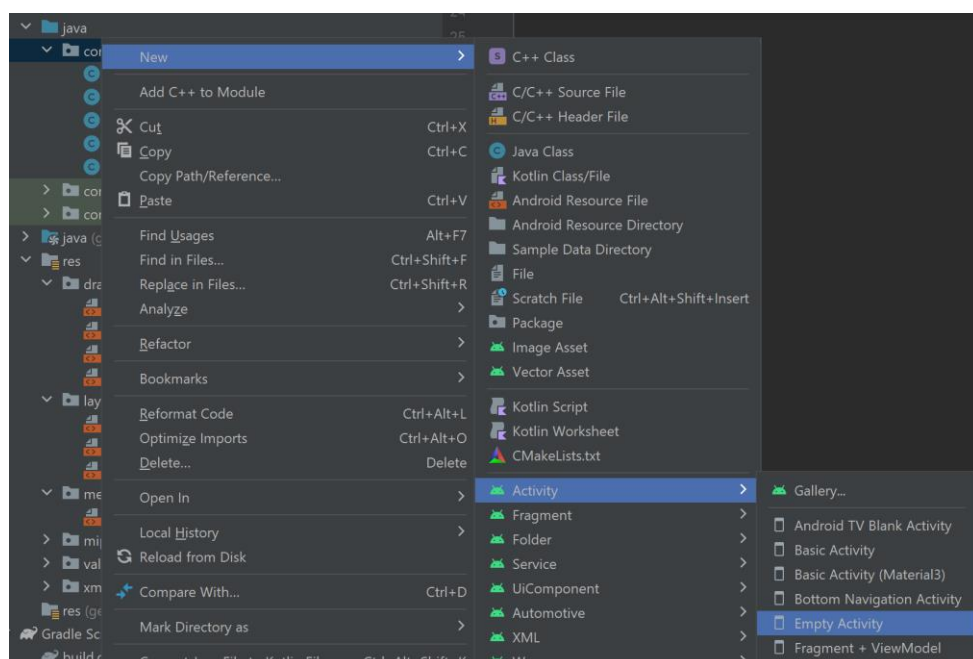
7. Buatlah vector assets pada drawable dengan cara klik kanan pada folder drawable lalu pilih New >> Vector Assets



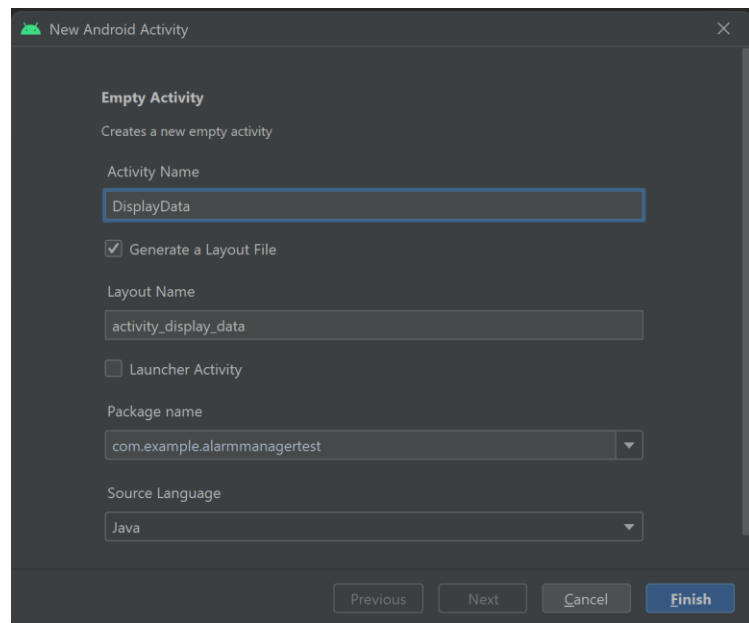
8. Carilah clip art more vert dan star rate sesuai seperti pada gambar dibawah ini lalu simpan.



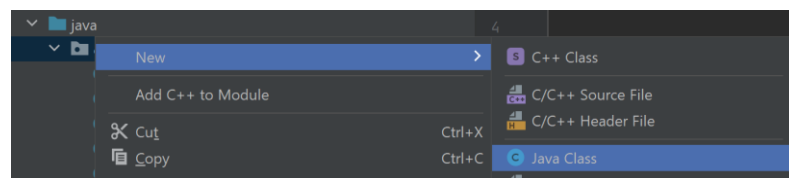
9. Kemudian membuat empty activity dengan cara klik kanan pada com.example > new > Activity > Empty Activity.



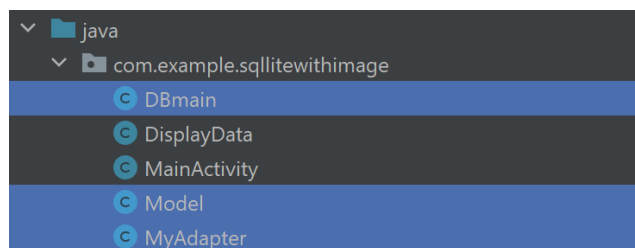
10. Berikan identitas Empty Activity seperti pada gambar dibawah ini.



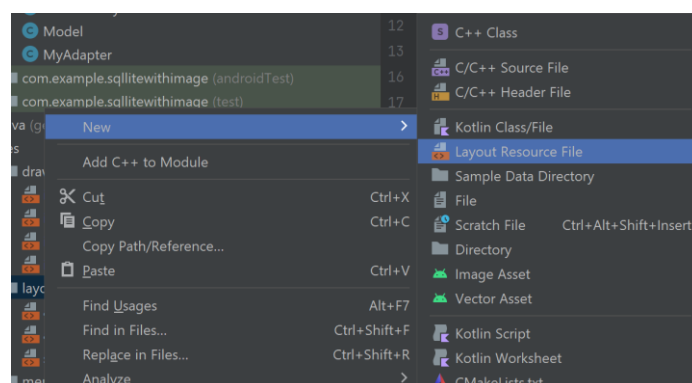
11. Kemudian membuat java class baru dengan cara klik kanan pada com.example > new > Java Class.



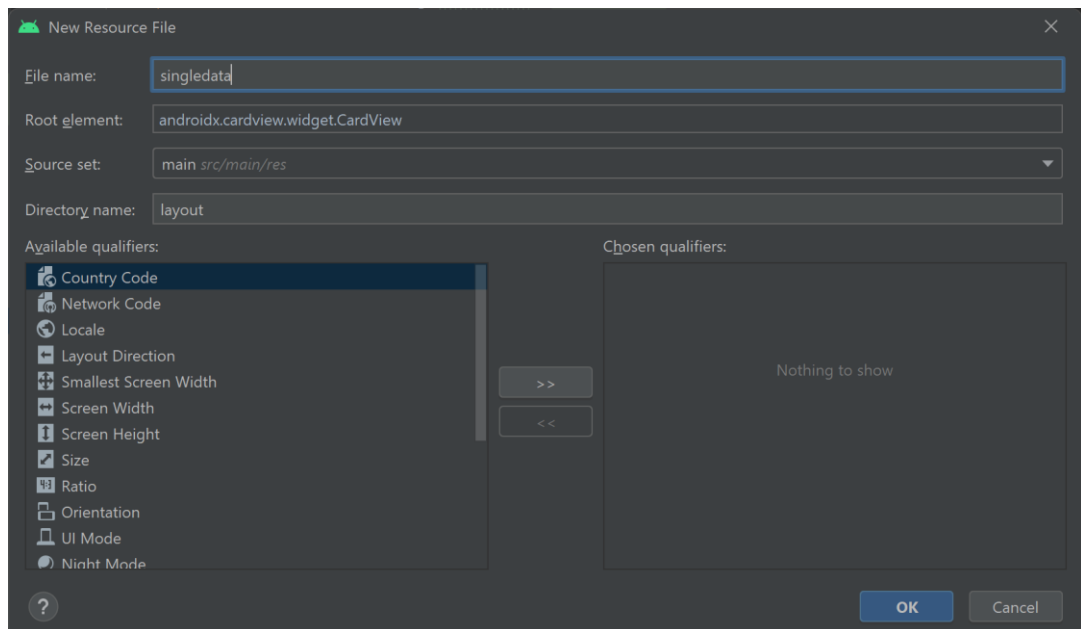
12. Buat 3 file Java class dengan nama masing – masing adalah DBmain, Model, MyAdapeter seperti pada gambar dibawah ini.



13. Kemudian membuat Layout Resource File dengan cara klik kanan pada folder layout > new > Layout Resource File.



14. Beri identitas Layout Resource File seperti pada gambar dibawah ini.



15. Kemudian buka folder layout > double klik file singledata.xml, lalu tambahkan *source code* berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="130dp"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="2dp"
    app:cardPreventCornerOverlap="true"
    app:cardUseCompatPadding="true"
    app:cardCornerRadius="8dp">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <ImageButton
            android:background="@null"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:id="@+id/flowmenu"
            android:layout_centerVertical="true"
            android:layout_alignParentRight="true"
            android:src="@drawable/ic_baseline_more_vert_24"
            android:layout_alignParentTop="true"/>
        <ImageView
            android:scaleType="fitXY"
            android:layout_marginTop="10dp"
            android:layout_marginBottom="10dp"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:paddingLeft="10dp"
            android:id="@+id/viewavatar"
            android:src="@drawable/ic_launcher_background"/>
        <TextView
            android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/viewavatar"
        android:text="@string/app_name"
        android:id="@+id/txt_name"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:layout_toRightOf="@+id/viewavatar"
        android:layout_alignTop="@+id/txt_name"
        android:layout_marginTop="25dp"
        android:layout_marginStart="10dp"
        android:id="@+id/linear"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="30dp"
            android:layout_height="wrap_content"
            android:foregroundGravity="left"
            app:srcCompat="@drawable/ic_baseline_star_rate_24" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_toRightOf="@+id/viewavatar"
            android:text="@string/app_name"
            android:id="@+id/txt_star"/>
    </LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/viewavatar"
        android:text="@string/app_name"
        android:layout_alignTop="@+id/linear"
        android:id="@+id/txt_price"
        android:layout_marginTop="30dp"
        android:layout_marginLeft="20dp"/>
</RelativeLayout>
</androidx.cardview.widget.CardView>

```

16. Kemudian buka folder com.example > double klik file DBmain.java, lalu tambahkan *source code* berikut.

```

package com.example.sqllitewithimage;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DBmain extends SQLiteOpenHelper {
    public static final String DBNAME="makanan.db";
    public static final String TABLENAME="tradisional";
    public static final int VER=1;
    public DBmain(@Nullable Context context) {
        super(context, DBNAME, null, VER);
    }
}

```



```

@Override
public void onCreate(SQLiteDatabase db) {
    String query = "create table "+TABLENAME+"(id integer
primary key, name TEXT, star TEXT, price TEXT, avatar blob)";
    db.execSQL(query);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    String query = "drop table if exists "+TABLENAME+"";
    db.execSQL(query);
    onCreate(db);
}
}

```

17. Kemudian buka folder com.example > double klik file Model.java, lalu tambahkan *source code* berikut.

```

package com.example.sqllitewithimage;

public class Model {
    private int id;
    private byte[]proavatar;
    private String username;
    private String userstar;
    private String userprice;

    //constructor
    public Model(int id, byte[] proavatar, String username,
String userstar, String userprice) {
        this.id = id;
        this.proavatar = proavatar;
        this.username = username;
        this.userstar = userstar;
        this.userprice = userprice;
    }

    //getter and setter method

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public byte[] getProavatar() {
        return proavatar;
    }

    public void setProavatar(byte[] proavatar) {
        this.proavatar = proavatar;
    }

    public String getUsername() {
        return username;
    }
}

```

```

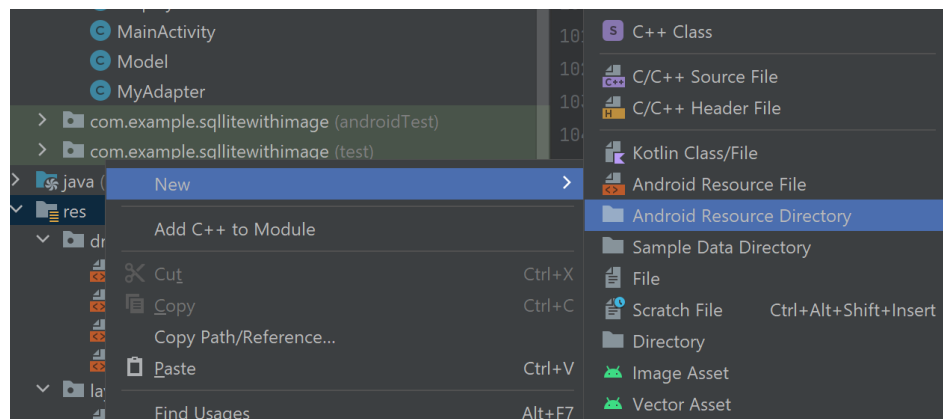
    public void setUsername(String username) {
        this.username = username;
    }
    public String getUserstar() {
        return userstar;
    }

    public void setUserstar(String userstar) {
        this.userstar = userstar;
    }
    public String getUserprice() {
        return userprice;
    }

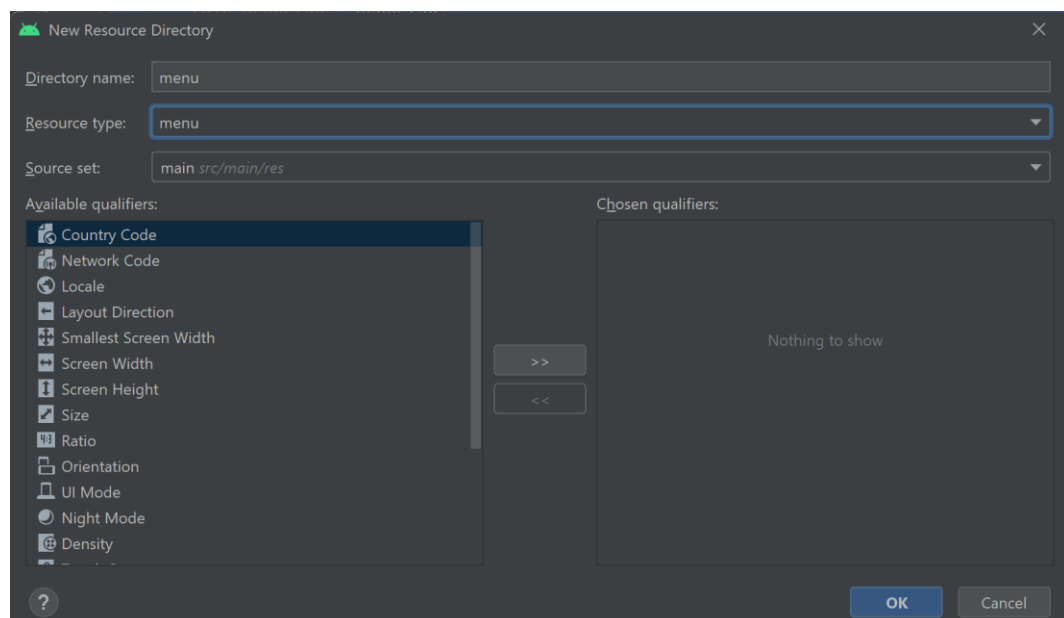
    public void setUserprice(String userprice) {
        this.userprice = userprice;
    }
}

```

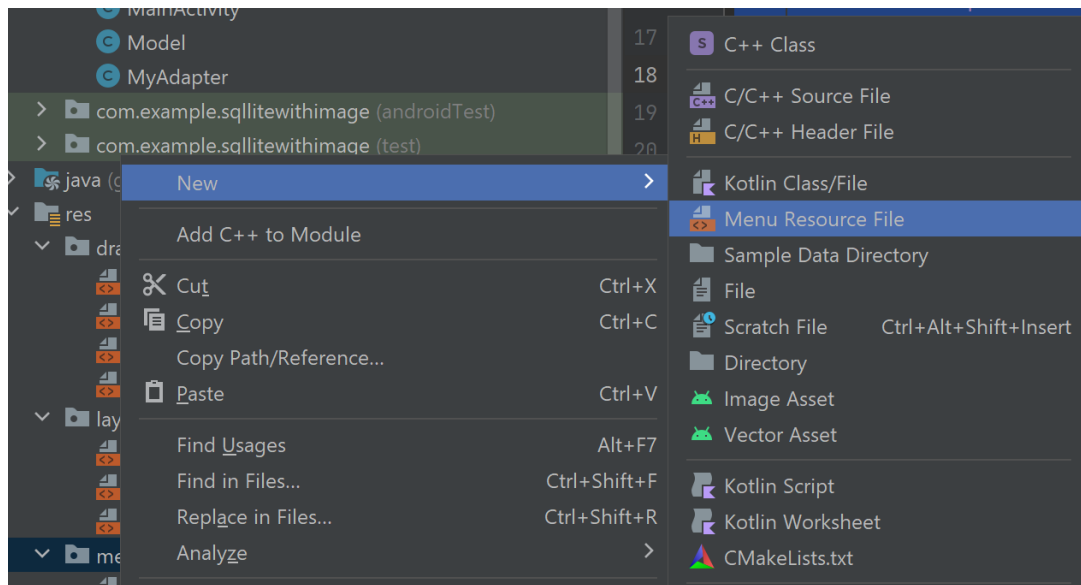
18. Kemudian membuat Android Resource Directory dengan cara klik kanan pada folder res > New > Android Resource Directory.



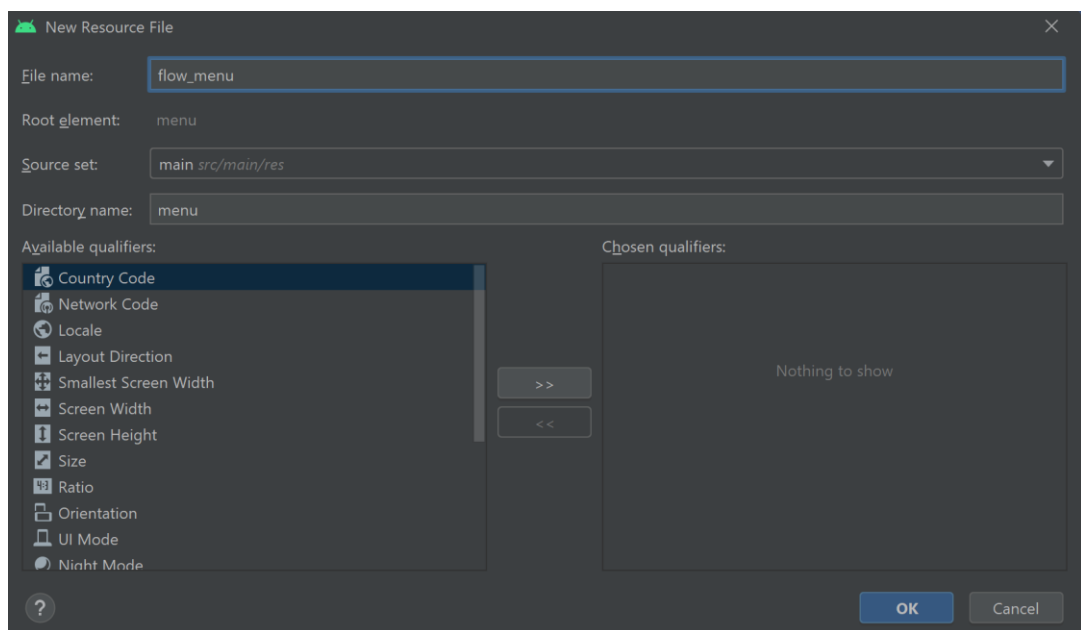
19. Kemudian isikan identitas Android Resource Directory sesuai pada gambar dibawah ini.



20. Kemudian klik kanan pada folder menu > New > Menu Resource File



21. Kemudian berikan identitas Menu Resource File seperti pada gambar dibawah ini.



22. Langkah selanjutnya buka file flow\_menu.xml pada folder menu, kemudian tambahkan *source code* berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="Edit"
          android:id="@+id/edit_menu"/>
    <item android:title="Delete"
          android:id="@+id/delete_menu"/>
</menu>
```

23. Kemudian tambahkan *source code* berikut pada *activity\_main.xml*.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <ImageView
        android:layout_marginTop="14dp"
        android:layout_gravity="center"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:src="@mipmap/ic_launcher"
        android:id="@+id/edtimage"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="text"
        android:id="@+id/edtname"
        android:layout_marginTop="10dp"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Star"
        android:inputType="text"
        android:id="@+id/edtstar"
        android:layout_marginTop="10dp"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Price"
        android:inputType="text"
        android:id="@+id/edtprice"
        android:layout_marginTop="10dp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="10dp">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:id="@+id/btn_submit"
            android:text="Submit"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/btn_edit"
            android:visibility="gone"
            android:text="Edit"
            android:layout_weight="1"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Display"
```

```

        android:id="@+id/btn_display"
        android:layout_weight="1"/>
    </LinearLayout>
</LinearLayout>

```

24. Kemudian tambahkan *source code* berikut pada `activity_display_data.xml`.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DisplayData">
    <Button
        android:layout_marginTop="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn_create"
        android:text="Create" />
    <androidx.recyclerview.widget.RecyclerView
        android:layout_marginTop="75dp"
        android:layout_width="match_parent"
        android:layout_height="1000dp"
        android:id="@+id/rv"/>
</RelativeLayout>

```

25. Kemudian tambahkan *source code* berikut pada `MyAdapter.java`

```

package com.example.sqlлитewithimage;

import static com.example.sqlлитewithimage.DBmain.TABLENAME;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.PopupMenu;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class MyAdapter extends
    RecyclerView.Adapter<MyAdapter.ViewHolder> {
    Context context;

```

```

        int singledata;
        ArrayList<Model>modelArrayList;
        SQLiteDatabase sqLiteDatabase;

        //generate constructor
        public MyAdapter(Context context, int singledata,
        ArrayList<Model> modelArrayList, SQLiteDatabase sqLiteDatabase) {
            this.context = context;
            this.singledata = singledata;
            this.modelArrayList = modelArrayList;
            this.sqliteDatabase = sqLiteDatabase;
        }

        @NonNull
        @Override
        public MyAdapter.ViewHolder onCreateViewHolder(@NonNull
        ViewGroup parent, int viewType) {
            LayoutInflater inflater = LayoutInflater.from(context);
            View view = inflater.inflate(R.layout.singledata, null);
            return new ViewHolder(view);
        }

        @Override
        public void onBindViewHolder(@NonNull MyAdapter.ViewHolder
        holder, @SuppressWarnings("RecyclerView") int position) {
            final Model model = modelArrayList.get(position);
            byte[] image = model.getProavatar();
            Bitmap bitmap = BitmapFactory.decodeByteArray(image, 0,
            image.length);
            holder.imageavatar.setImageBitmap(bitmap);
            holder.txtname.setText(model.getUsername());
            holder.txtstar.setText(model.getUserstar());
            holder.txtprice.setText(model.getUserprice());

            //flow menu
            holder.flowmenu.setOnClickListener(new
            View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    PopupMenu popupMenu = new PopupMenu(context,
                    holder.flowmenu);
                    popupMenu.inflate(R.menu.flow_menu);
                    popupMenu.setOnMenuItemClickListener(new
                    PopupMenu.OnMenuItemClickListener() {
                        @Override
                        public boolean onMenuItemClick(MenuItem item)
                        {
                            switch (item.getItemId()) {
                                case R.id.edit_menu:
                                    //edit operation
                                    Bundle bundle = new Bundle();

                                    bundle.putInt("id",model.getId());

                                    bundle.putString("name",model.getUsername());

                                    bundle.putString("star",model.getUserstar());

                                    bundle.putString("price",model.getUserprice());

                                    bundle.putByteArray("avatar",model.getProavatar());

```

```

Intent intent = new
Intent(context, MainActivity.class);

intent.putExtra("userdata",bundle);
context.startActivity(intent);
break;
case R.id.delete_menu:
//delete operation
DBmain dBmain = new
DBmain(context);
sqliteDatabase =
dBmain.getReadableDatabase();
long recdelete =
sqliteDatabase.delete(TABLENAME,"id="+model.getId(),null);
if (recdelete != -1){
Toast.makeText(context, "Data
Deleted",Toast.LENGTH_SHORT).show();
//remove positon after
deleted
modelArrayList.remove(position);
//update data
notifyDataSetChanged();
}
break;
default:
return false;
}
return false;
}
});
//display menu
popupMenu.show();
}
});
}

@Override
public int getItemCount() {
return modelArrayList.size();
}

public class ViewHolder extends RecyclerView.ViewHolder {
ImageView imageavatar;
TextView txtname,txtstar,txtprice;
ImageButton flowmenu;
public ViewHolder(@NonNull View itemView) {
super(itemView);
imageavatar =
(ImageView)itemView.findViewById(R.id.viewavatar);
txtname = (TextView)
itemView.findViewById(R.id.txt_name);
txtstar = (TextView)
itemView.findViewById(R.id.txt_star);
txtprice = (TextView)
itemView.findViewById(R.id.txt_price);
flowmenu = (ImageButton)
itemView.findViewById(R.id.flowmenu);
}
}
}

```

26. Kemudian tambahkan *source code* berikut pada MainActivity.java

```
package com.example.sqllitewithimage;

import static com.example.sqllitewithimage.DBmain.TABLERNAME;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.ContentValues;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

import com.squareup.picasso.Picasso;
import com.theartofdev.edmodo.cropper.CropImage;
import com.example.sqllitewithimage.databinding.ActivityMainBinding;

import java.io.ByteArrayOutputStream;

public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;
    DBmain dBmain;
    SQLiteDatabase sqLiteDatabase;
    int id = 0;

    public static final int MY_CAMERA_REQUEST_CODE = 100;
    public static final int MY_STORAGE_REQUEST_CODE = 101;

    String cameraPermission[];
    String storagePermission[];

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        dBmain = new DBmain(this);
        //findid();
        insertData();
        editData();

        binding.edtimage.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int avatar = 0;
```



```

        if (avatar == 0) {
            if (!checkCameraPermission()) {
                requestCameraPermission();
            } else {
                pickFromGallery();
            }
        } else if (avatar == 1) {
            if (!checkStoragePermission()) {
                requestStoragePermission();
            } else {
                pickFromGallery();
            }
        }
    }
});

private void editData() {
    if (getIntent().getBundleExtra("userdata") != null) {
        Bundle bundle =
getIntent().getBundleExtra("userdata");
        id = bundle.getInt("id");

        //for set name
        binding.edtname.setText(bundle.getString("name"));
        binding.edtstar.setText(bundle.getString("star"));
        binding.edtprice.setText(bundle.getString("price"));

        //for image
        byte[] bytes = bundle.getBytes("avatar");
        Bitmap bitmap = BitmapFactory.decodeByteArray(bytes,
0, bytes.length);
        binding.edtimage.setImageBitmap(bitmap);

        //visible edit button and hide submit button
        binding.btnSubmit.setVisibility(View.GONE);
        binding.btnEdit.setVisibility(View.VISIBLE);
    }
}

private void requestStoragePermission() {
    requestPermissions(storagePermission,
MY_STORAGE_REQUEST_CODE);
}

private boolean checkStoragePermission() {
    boolean result = ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
    return result;
}

private void pickFromGallery() {
    CropImage.activity().start(this);
}

private void requestCameraPermission() {
    requestPermissions(cameraPermission,
MY_CAMERA_REQUEST_CODE);
}

```

```

        private boolean checkCameraPermission() {
            boolean result = ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
            boolean result1 = ContextCompat.checkSelfPermission(this,
Manifest.permission.CAMERA) ==
(PackageManager.PERMISSION_GRANTED);
            return result && result1;
        }

        private void insertData() {
            binding.btnSubmit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    ContentValues cv = new ContentValues();
                    cv.put("name",
binding.edtname.getText().toString());

                    cv.put("star",binding.edtstar.getText().toString());
                    cv.put("avatar",
imageViewToBy(binding.edtimage));

                    cv.put("price",binding.edtprice.getText().toString());

                    sqLiteDatabase = dBmain.getWritableDatabase();
                    Long rec =
sqLiteDatabase.insert("tradicional",null ,cv);
                    if (rec != null) {
                        Toast.makeText(MainActivity.this, "Data
Inserted", Toast.LENGTH_SHORT).show();
                        binding.edtname.setText("");

                    binding.edtimage.setImageResource(R.mipmap.ic_launcher);
                        binding.edtstar.setText("");
                        binding.edtprice.setText("");
                    } else {
                        Toast.makeText(MainActivity.this, "Something
Wrong", Toast.LENGTH_SHORT).show();
                    }
                }
            });

            //for view display
            binding.btnDisplay.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    startActivity(new Intent( MainActivity.this,
DisplayData.class));
                }
            });

            //for storing new data or update data
            binding.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    ContentValues cv = new ContentValues();

                    cv.put("name",binding.edtname.getText().toString());

```

```

cv.put("star",binding.edtstar.getText().toString());

cv.put("price",binding.edtprice.getText().toString());
    cv.put("avatar",
imageViewToBy(binding.edtimage));

        sqLiteDatabase = dBmain.getWritableDatabase();
        long recedit =
sqLiteDatabase.update(TABLENAME,cv,"id="+id, null);
        if(recedit != -1){
            Toast.makeText(MainActivity.this, "Update
Sucesfully", Toast.LENGTH_SHORT).show();
            //clear data adfte submit
            binding.edtname.setText("");
            binding.edtstar.setText("");
            binding.edtprice.setText("");

binding.edtimage.setImageResource(R.mipmap.ic_launcher);
            //edit hide and submit visible
            binding.btnEdit.setVisibility(View.GONE);

binding.btnSubmit.setVisibility(View.VISIBLE);
            Intent a = new Intent(MainActivity.this,
                DisplayData.class);
            startActivity(a);
        }
    }
});
}

    public static byte[] imageViewToBy(Imageview avatar) {
        Bitmap bitmap = ((BitmapDrawable)
avatar.getDrawable()).getBitmap();
        ByteArrayOutputStream stream = new
ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.JPEG, 50, stream);
        byte[] bytes = stream.toByteArray();
        return bytes;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,
@NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
        switch (requestCode) {
            case MY_CAMERA_REQUEST_CODE: {
                if (grantResults.length > 0) {
                    boolean camera_accepted = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                    boolean storage_accepted = grantResults[1] ==
PackageManager.PERMISSION_GRANTED;
                    if (camera_accepted && storage_accepted) {
                        pickFromGallery();
                    } else {
                        Toast.makeText(this, "enable camera and
storage permission", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }
    }
}

```

```

        break;
        case MY_STORAGE_REQUEST_CODE: {
            boolean storage_accepted = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
            if (storage_accepted) {
                pickFromGallery();
            } else {
                Toast.makeText(this, "please enable storage
permission", Toast.LENGTH_SHORT).show();
            }
        }
        break;
    }
}

@Override
protected void onActivityResult(int requestCode, int
resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode
==CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE){
        CropImage.ActivityResult result =
CropImage.getActivityResult(data);
        if(resultCode == RESULT_OK){
            Uri resultUri = result.getUri();

Picasso.with(this).load(resultUri).into(binding.edtimage);
        }
    }
}
}

```

27. Kemudian tambahkan *source code* berikut pada DisplayData.java

```

package com.example.sqllitewithimage;

import static com.example.sqllitewithimage.DBmain.TABLENAME;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import
com.example.sqllitewithimage.databinding.ActivityDisplayDataBindi
ng;

import java.util.ArrayList;

public class DisplayData extends AppCompatActivity {
    DBmain dBmain;
    SQLiteDatabase sqLiteDatabase;
    RecyclerView recyclerView;
    MyAdapter myAdapter;
    private ActivityDisplayDataBinding binding;

    @Override

```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            binding =
                ActivityDisplayDataBinding.inflate(getLayoutInflater());
            setContentView(binding.getRoot());

            findId();
            dBmain = new DBmain(this);
            displayData();
            recyclerView.setLayoutManager(new
                LinearLayoutManager(this, RecyclerView.VERTICAL, false));

            binding.btnCreate.setOnClickListener(new
                View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent a = new Intent(DisplayData.this,
                            MainActivity.class);
                        startActivity(a);
                    }
                });
        }

        private void displayData() {
            sqLiteDatabase = dBmain.getReadableDatabase();
            Cursor cursor = sqLiteDatabase.rawQuery("select * from
                "+TABLENAME,null);
            ArrayList<Model> models = new ArrayList<>();
            while (cursor.moveToNext()) {
                int id = cursor.getInt(0);
                String name = cursor.getString(1);
                byte[] avatar = cursor.getBlob(4);
                String star = cursor.getString(2);
                String price = cursor.getString(3);
                models.add(new Model(id, avatar, name, star, price));
            }
            cursor.close();
            myAdapter = new MyAdapter(this,
                R.layout.singledata, models, sqLiteDatabase);
            recyclerView.setAdapter(myAdapter);
        }

        private void findId() {
            recyclerView = findViewById(R.id.rv);
        }
    }

```

28. Pastikan *permission* (pengelola izin) untuk *camera* dan *storage* pada smartphone atau emulator kalian sudah diaktifkan. Kemudian jalankan program.

## 8.5 Tugas Rumah

Melanjutkan proyek sesuai tema dengan mengimplementasikan SQLite dan tambahkan menu pada action bar atau navigation bar. Pastikan tampilan UI yang dibuat sebagus dan kreatif mungkin.