

Mata Kuliah : Pemrograman Mobile – TI – S1

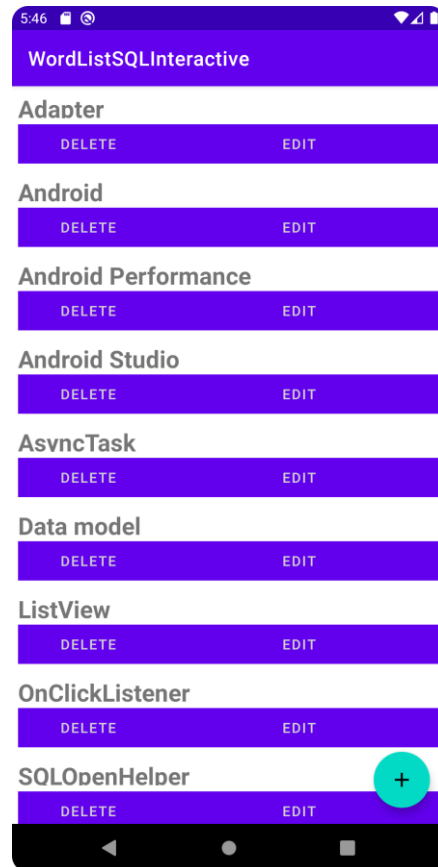
Pertemuan : Minggu 8

NIM : A11.2021.13509

Nama : Ainindzi Nur Meiza Pudjianto

1. WordListSQLInteractive

Hasil Program :



Source Code:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.recyclerview.widget.RecyclerView>
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        android:clickable="true"
        android:src="@drawable/ic_add_24dp" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

activity_edit_word.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.recyclerview.widget.RecyclerView>
    <com.google.android.material.floatingactionbutton.FloatingActionButton
```

```
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    android:clickable="true"
    android:src="@drawable/ic_add_24dp" />
```

```
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

wordlist_item.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="6dp">
```

```
    <TextView
        android:id="@+id/word"
        android:layout_width="match_parent"
        style="@style/word_title" />
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
```

```
        <Button
            android:id="@+id/delete_button"
            android:layout_width="match_parent"
            android:layout_height="@dimen/button_height"
            android:layout_weight="2"
            android:background="@color/colorPrimaryDark"
            android:text="@string/button_delete"
            android:textColor="@color/buttonLabel"/>
```

```
        <Button
            android:id="@+id/edit_button"
            android:layout_width="match_parent"
            android:layout_height="@dimen/button_height"
            android:layout_weight="1"
            android:background="@color/colorPrimary"
            android:text="@string/button_edit"
            android:textColor="@color/buttonLabel"/>
```

```
</LinearLayout>
```

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="@dimen/divider_height"  
    android:background="@color/colorAccent" />
```

```
</LinearLayout>
```

MainActivity.java:

```
package com.example.wordlistsqlineractive;
```

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

```
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Toast;
```

```
import com.google.android.material.floatingactionbutton.FloatingActionButton;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    private static final String TAG = MainActivity.class.getSimpleName();
```

```
    public static final int WORD_EDIT = 1;  
    public static final int WORD_ADD = -1;
```

```
    private WordListOpenHelper mDB;  
    private RecyclerView mRecyclerView;  
    private WordListAdapter mAdapter;  
    private int mLastPosition;
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    mDB = new WordListOpenHelper(this);
```

```
    // Create recycler view.
```

```
    mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
```

```

// Create an mAdapter and supply the data to be displayed.
mAdapter = new WordListAdapter(this, /* mDB.getAllEntries(),*/ mDB);
// Connect the mAdapter with the recycler view.
mRecyclerView.setAdapter(mAdapter);
// Give the recycler view a default layout manager.
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));

// Add a floating action click handler for creating new entries.
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Starts empty edit activity.
        Intent intent = new Intent(getBaseContext(), EditWordActivity.class);
        startActivityForResult(intent, WORD_EDIT);
    }
});
}

public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == WORD_EDIT) {
        if (resultCode == RESULT_OK) {
            String word = data.getStringExtra(EditWordActivity.EXTRA_REPLY);

            // Update the database.
            if (!TextUtils.isEmpty(word)) {
                int id = data.getIntExtra(WordListAdapter.EXTRA_ID, -99);

                if (id == WORD_ADD) {
                    mDB.insert(word);
                } else if (id >= 0) {
                    mDB.update(id, word);
                }
                // Update the UI.
                mAdapter.notifyDataSetChanged();
            } else {
                Toast.makeText(
                    getApplicationContext(),
                    R.string.empty_not_saved,
                    Toast.LENGTH_LONG).show();
            }
        }
    }
}
}

```

```
}
```

EditWordActivity.java:

```
package com.example.wordlistsqlineractive;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.EditText;
```

```
public class EditWordActivity extends AppCompatActivity {
```

```
    private static final String TAG = EditWordActivity.class.getSimpleName();
```

```
    private static final int NO_ID = -99;
```

```
    private static final String NO_WORD = "";
```

```
    private EditText mEditWordView;
```

```
    // Unique tag for the intent reply.
```

```
    public static final String EXTRA_REPLY = "com.example.android.wordlistsql.REPLY";
```

```
    int mId = MainActivity.WORD_ADD;
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_edit_word);
```

```
        mEditWordView = (EditText) findViewById(R.id.edit_word);
```

```
        // Get data sent from calling activity.
```

```
        Bundle extras = getIntent().getExtras();
```

```
        // If we are passed content, fill it in for the user to edit.
```

```
        if (extras != null) {
```

```
            int id = extras.getInt(WordListAdapter.EXTRA_ID, NO_ID);
```

```
            String word = extras.getString(WordListAdapter.EXTRA_WORD, NO_WORD);
```

```
            if ((id != NO_ID) && (word != NO_WORD)) {
```

```
                mId = id;
```

```
                mEditWordView.setText(word);
```

```
            }
```

```
        } // Otherwise, start with empty fields.
```

```

    }

    /**
     * Click handler for the Save button.
     * Creates a new intent for the reply, adds the reply message to it as an extra,
     * sets the intent result, and closes the activity.
     *
     * @param view The view that was clicked.
     */
    public void returnReply(View view) {
        String word = ((EditText) findViewById(R.id.edit_word)).getText().toString();

        Intent replyIntent = new Intent();
        replyIntent.putExtra(EXTRA_REPLY, word);
        replyIntent.putExtra(WordListAdapter.EXTRA_ID, mId);
        setResult(RESULT_OK, replyIntent);
        finish();
    }
}

```

MyButtonOnClickListener.java:

```
package com.example.wordlistsqlineractive;
```

```
import android.view.View;
```

```

public class MyButtonOnClickListener implements View.OnClickListener {
    private static final String TAG = View.OnClickListener.class.getSimpleName();

    int id;
    String word;

    public MyButtonOnClickListener(int id, String word) {
        this.id = id;
        this.word = word;
    }

    public void onClick(View v) {
        // Implemented in WordListAdapter
    }
}

```

WordItem.java:

```
package com.example.wordlistsqlineractive;
```

```
public class WordItem {
```

```

private int mId;
private String mWord;

public WordItem() {}

public int getId() {
    return this.mId;
}

public String getWord() {
    return this.mWord;
}

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

public void setWord(String word) {
    this.mWord = word;
}
}

```

WordListAdapter.java:

```
package com.example.wordlistsqlineractive;
```

```

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

```

```
import androidx.recyclerview.widget.RecyclerView;
```

```
/**
```

```
 * Implements a simple Adapter for a RecyclerView.
```

```
 * Demonstrates how to add a click handler for each item in the ViewHolder.
```

```
 */
```

```

public class WordListAdapter extends
RecyclerView.Adapter<WordListAdapter.WordViewHolder> {

```

```
/**
```



```

* Custom view holder with a text view and two buttons.
*/
class WordViewHolder extends RecyclerView.ViewHolder {
    public final TextView wordItemView;
    Button delete_button;
    Button edit_button;

    public WordViewHolder(View itemView) {
        super(itemView);
        wordItemView = (TextView) itemView.findViewById(R.id.word);
        delete_button = (Button) itemView.findViewById(R.id.delete_button);
        edit_button = (Button) itemView.findViewById(R.id.edit_button);
    }
}

private static final String TAG = WordListAdapter.class.getSimpleName();

public static final String EXTRA_ID = "ID";
public static final String EXTRA_WORD = "WORD";
public static final String EXTRA_POSITION = "POSITION";

private final LayoutInflater mInflater;
WordListOpenHelper mDB;
Context mContext;

public WordListAdapter(Context context, WordListOpenHelper db) {
    mInflater = LayoutInflater.from(context);
    mContext = context;
    mDB = db;
}

@Override
public WordViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = mInflater.inflate(R.layout.wordlist_item, parent, false);
    return new WordViewHolder(itemView);
}

@Override
public void onBindViewHolder(WordViewHolder holder, int position) {
    WordItem current = mDB.query(position);
    holder.wordItemView.setText(current.getWord());
    // Keep a reference to the view holder for the click listener
    final WordViewHolder h = holder; // needs to be final for use in callback
    // Attach a click listener to the DELETE button.
    holder.delete_button.setOnClickListener(new MyButtonOnClickListener(
        current.getId(), null) {

```

```

@Override
public void onClick(View v ) {
    // You have to get the position like this, you can't hold a reference
    Log.d (TAG + "onClick", "VHPos " + h.getAdapterPosition() + " ID " + id);
    int deleted = mDB.delete(id);
    if (deleted >= 0)
        notifyItemRemoved(h.getAdapterPosition());
    }
});

// Attach a click listener to the EDIT button.
holder.edit_button.setOnClickListener(new MyButtonOnClickListener(
    current.getId(), current.getWord()) {

    @Override
    public void onClick(View v) {
        Intent intent = new Intent(mContext, EditWordActivity.class);

        intent.putExtra(EXTRA_ID, id);
        intent.putExtra(EXTRA_POSITION, h.getAdapterPosition());
        intent.putExtra(EXTRA_WORD, word);

        // Start an empty edit activity.
        ((Activity) mContext).startActivityForResult(intent, MainActivity.WORD_EDIT);
    }
});
}

@Override
public int getItemCount() {
    return (int) mDB.count();
}
}

```

WordListOpenHelper.java:

```

package com.example.wordlistsqlineractive;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

```

```
import android.util.Log;
```

```
public class WordListOpenHelper extends SQLiteOpenHelper {
```

```
    private static final String TAG = WordListOpenHelper.class.getSimpleName();
```

```
    // Declaring all these as constants makes code a lot more readable, and looking like SQL.
```

```
    // Versions has to be 1 first time or app will crash.
```

```
    private static final int DATABASE_VERSION = 1;
```

```
    private static final String WORD_LIST_TABLE = "word_entries";
```

```
    private static final String DATABASE_NAME = "wordlist";
```

```
    // Column names...
```

```
    public static final String KEY_ID = "_id";
```

```
    public static final String KEY_WORD = "word";
```

```
    // ... and a string array of columns.
```

```
    private static final String[] COLUMNS =  
        {KEY_ID, KEY_WORD};
```

```
    // Build the SQL query that creates the table.
```

```
    private static final String WORD_LIST_TABLE_CREATE =
```

```
        "CREATE TABLE " + WORD_LIST_TABLE + " (" +  
            KEY_ID + " INTEGER PRIMARY KEY, " + // will auto-increment if no value
```

```
passed
```

```
            KEY_WORD + " TEXT );";
```

```
    private SQLiteDatabase mWritableDatabase;
```

```
    private SQLiteDatabase mReadableDB;
```

```
    public WordListOpenHelper(Context context) {
```

```
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
```

```
        Log.d(TAG, "Construct WordListOpenHelper");
```

```
    }
```

```
    @Override
```

```
    public void onCreate(SQLiteDatabase db) {
```

```
        db.execSQL(WORD_LIST_TABLE_CREATE);
```

```
        fillDatabaseWithData(db);
```

```
        // We cannot initialize mWritableDatabase and mReadableDB here, because this creates an  
infinite
```

```
        // loop of on Create being repeatedly called.
```

```
    }
```

```

/**
 * Adds the initial data set to the database.
 * According to the docs, onCreate for the open helper does not run on the UI thread.
 *
 * @param db Database to fill with data since the member variables are not initialized yet.
 */
public void fillDatabaseWithData(SQLiteDatabase db) {

    String[] words = {"Android", "Adapter", "ListView", "AsyncTask", "Android Studio",
        "SQLiteDatabase", "SQLiteOpenHelper", "Data model", "ViewHolder",
        "Android Performance", "OnClickListener"};

    // Create a container for the data.
    ContentValues values = new ContentValues();

    for (int i=0; i < words.length;i++) {
        // Put column/value pairs into the container. put() overwrites existing values.
        values.put(KEY_WORD, words[i]);
        db.insert(WORD_LIST_TABLE, null, values);
    }
}

/**
 * Queries the database for an entry at a given position.
 *
 * @param position The Nth row in the table.
 * @return a WordItem with the requested database entry.
 */
public WordItem query(int position) {
    String query = "SELECT * FROM " + WORD_LIST_TABLE +
        " ORDER BY " + KEY_WORD + " ASC " +
        "LIMIT " + position + ",1";

    Cursor cursor = null;
    WordItem entry = new WordItem();

    try {
        if (mReadableDB == null) {mReadableDB = getReadableDatabase();}
        cursor = mReadableDB.rawQuery(query, null);
        cursor.moveToFirst();
        entry.setId(cursor.getInt(cursor.getColumnIndex(KEY_ID)));
        entry.setWord(cursor.getString(cursor.getColumnIndex(KEY_WORD)));
    } catch (Exception e) {
        Log.d(TAG, "QUERY EXCEPTION! " + e.getMessage());
    } finally {
        // Must close cursor and db now that we are done with it.

```

```

        cursor.close();
        return entry;
    }
}

/**
 * Gets the number of rows in the word list table.
 *
 * @return The number of entries in WORD_LIST_TABLE.
 */
public long count() {
    if (mReadableDB == null) {mReadableDB = getReadableDatabase();}
    return DatabaseUtils.queryNumEntries(mReadableDB, WORD_LIST_TABLE);
}

/**
 * Adds a single word row/entry to the database.
 *
 * @param word New word.
 * @return The id of the inserted word.
 */
public long insert(String word) {
    long newId = 0;
    ContentValues values = new ContentValues();
    values.put(KEY_WORD, word);
    try {
        if (mWritableDatabase == null) {mWritableDatabase = getWritableDatabase();}
        newId = mWritableDatabase.insert(WORD_LIST_TABLE, null, values);
    } catch (Exception e) {
        Log.d(TAG, "INSERT EXCEPTION! " + e.getMessage());
    }
    return newId;
}

/**
 * Updates the word with the supplied id to the supplied value.
 *
 * @param id Id of the word to update.
 * @param word The new value of the word.
 * @return The number of rows affected or -1 if nothing was updated.
 */
public int update(int id, String word) {
    int mNumberOfRowsUpdated = -1;
    try {
        if (mWritableDatabase == null) {mWritableDatabase = getWritableDatabase();}
        ContentValues values = new ContentValues();

```

```

        values.put(KEY_WORD, word);

        mNumberOfRowsUpdated = mWritableDB.update(WORD_LIST_TABLE, //table to
change
        values, // new values to insert
        KEY_ID + " = ?", // selection criteria for row (in this case, the _id column)
        new String[]{String.valueOf(id)}); //selection args; the actual value of the id

    } catch (Exception e) {
        Log.d (TAG, "UPDATE EXCEPTION! " + e.getMessage());
    }
    return mNumberOfRowsUpdated;
}

/**
 * Deletes one entry identified by its id.
 *
 * @param id ID of the entry to delete.
 * @return The number of rows deleted. Since we are deleting by id, this should be 0 or 1.
 */
public int delete(int id) {
    int deleted = 0;
    try {
        if (mWritableDB == null) {mWritableDB = getWritableDatabase();}
        deleted = mWritableDB.delete(WORD_LIST_TABLE, //table name
        KEY_ID + " = ? ", new String[]{String.valueOf(id)});
    } catch (Exception e) {
        Log.d (TAG, "DELETE EXCEPTION! " + e.getMessage());
    }
    return deleted;
}

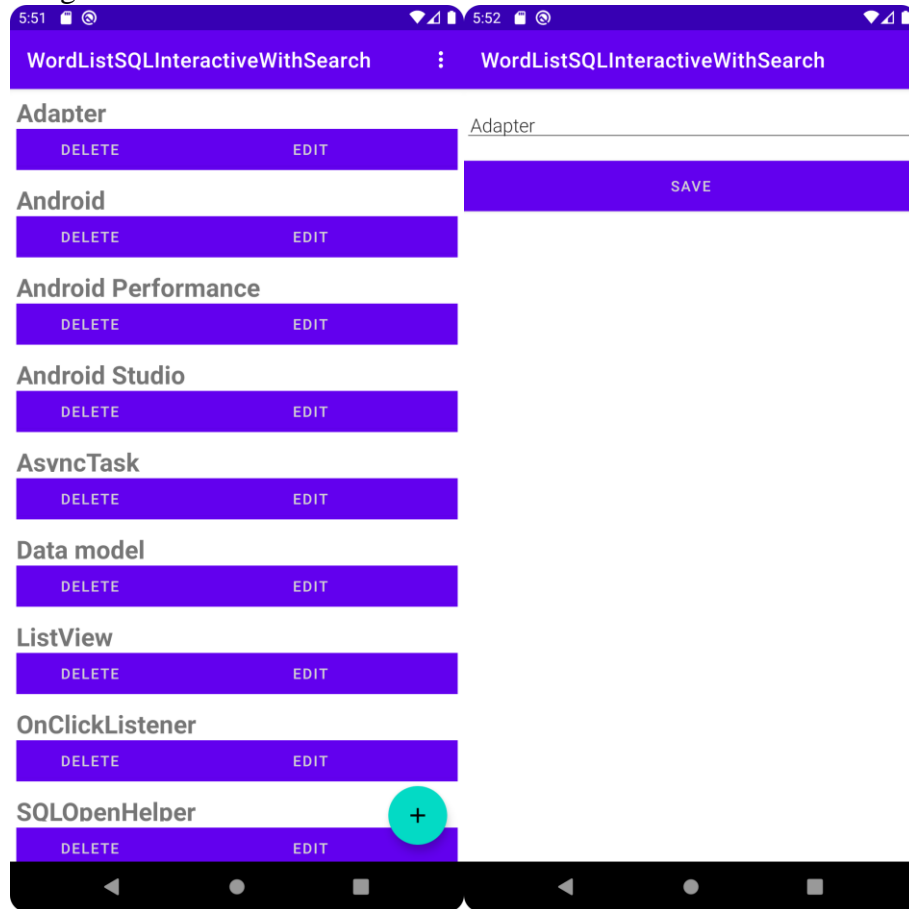
/**
 * Called when a database needs to be upgraded. The most basic version of this method drops
 * the tables, and then recreates them. All data is lost, which is why for a production app,
 * you want to back up your data first. If this method fails, changes are rolled back.
 *
 * @param db
 * @param oldVersion
 * @param newVersion
 */
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.w(WordListOpenHelper.class.getName(),
        "Upgrading database from version " + oldVersion + " to "
        + newVersion + ", which will destroy all old data");
    db.execSQL("DROP TABLE IF EXISTS " + WORD_LIST_TABLE);

```

```
        onCreate(db);  
    }  
}
```

2. WordListSQLInteractiveWithSearch

Hasil Program:



Source Code:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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=".MainActivity">

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</androidx.recyclerview.widget.RecyclerView>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```



```
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        android:clickable="true"
        android:src="@drawable/ic_add_24dp" />
```

```
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

activity_edit_word.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=".EditWordActivity">
```

```
    <EditText
        android:id="@+id/edit_word"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-light"
        android:hint="@string/hint_word"
        android:inputType="textAutoComplete"
        android:padding="@dimen/small_padding"
        android:layout_marginBottom="@dimen/big_padding"
        android:layout_marginTop="@dimen/big_padding"
        android:textSize="18sp" />
```

```
    <Button
        android:id="@+id/button_save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:onClick="returnReply"
        android:text="@string/button_save"
        android:textColor="@color/buttonLabel"
        tools:ignore="OnClick" />
```

```
</LinearLayout>
```

activity_search.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=".SearchActivity">
```

```
<EditText
    android:id="@+id/search_word"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif-light"
    android:hint="@string/hint_word"
    android:inputType="textAutoComplete"
    android:padding="@dimen/small_padding"
    android:layout_marginBottom="@dimen/big_padding"
    android:layout_marginTop="@dimen/big_padding"
    android:textSize="18sp" />
```

```
<Button
    android:id="@+id/button_search"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:onClick="showResult"
    android:text="@string/button_search"
    android:textColor="@color/buttonLabel" />
```

```
<TextView
    android:id="@+id/search_result"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:textSize="18sp"
    android:hint="@string/search_results"/>
```

```
</LinearLayout>
```

wordlist_item.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="6dp">
```

```
<TextView
```

```
    android:id="@+id/word"
    android:layout_width="match_parent"
    style="@style/word_title" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

```
    <Button
        android:id="@+id/delete_button"
        android:layout_width="match_parent"
        android:layout_height="@dimen/button_height"
        android:layout_weight="2"
        android:background="@color/colorPrimaryDark"
        android:text="@string/button_delete"
        android:textColor="@color/buttonLabel"/>
```

```
    <Button
        android:id="@+id/edit_button"
        android:layout_width="match_parent"
        android:layout_height="@dimen/button_height"
        android:layout_weight="1"
        android:background="@color/colorPrimary"
        android:text="@string/button_edit"
        android:textColor="@color/buttonLabel"/>
```

```
</LinearLayout>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="@dimen/divider_height"
    android:background="@color/colorAccent" />
```

```
</LinearLayout>
```

menu_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu 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"
    tools:context="com.example.android.optionsmenusample.MainActivity">
    <item
```

```
        android:id="@+id/action_search"
        android:orderInCategory="1"
        android:title="@string/menu_search"
        app:showAsAction="never" />
</menu>
```

MainActivity.java:

```
package com.example.wordlistsqlinteractivewithsearch;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    public static final int WORD_EDIT = 1;
    public static final int WORD_ADD = -1;

    private WordListOpenHelper mDB;
    private RecyclerView mRecyclerView;
    private WordListAdapter mAdapter;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mDB = new WordListOpenHelper(this);

        // Create recycler view.
        mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
        // Create an mAdapter and supply the data to be displayed.
        mAdapter = new WordListAdapter(this, mDB);
        // Connect the mAdapter with the recycler view.
        mRecyclerView.setAdapter(mAdapter);
```

```

// Give the recycler view a default layout manager.
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));

// Add a floating action click handler for creating new entries.
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Starts empty edit activity.
        Intent intent = new Intent(getApplicationContext(), EditWordActivity.class);
        startActivityForResult(intent, WORD_EDIT);
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_search:
            // Starts search activity.
            Intent intent = new Intent(getApplicationContext(),
com.example.wordlistsqlinteractivewithsearch.SearchActivity.class);
            startActivity(intent);
            return true;
        }
    return super.onOptionsItemSelected(item);
}

public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == WORD_EDIT) {
        if (resultCode == RESULT_OK) {
            String word = data.getStringExtra(EditWordActivity.EXTRA_REPLY);

            // Update the database.
            if (!TextUtils.isEmpty(word)) {
                int id = data.getIntExtra(WordListAdapter.EXTRA_ID, -99);

                if (id == WORD_ADD) {

```

```

        mDB.insert(word);
    } else if (id >= 0) {
        mDB.update(id, word);
    }
    // Update the UI.
    mAdapter.notifyDataSetChanged();
} else {
    Toast.makeText(
        getApplicationContext(),
        R.string.empty_word_not_saved,
        Toast.LENGTH_LONG).show();
}
}
}
}
}

```

EditWordActivity.java:

```
package com.example.wordlistsqlinteractivewithsearch;
```

```
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
public class EditWordActivity extends AppCompatActivity {
```

```
    private static final String TAG = EditWordActivity.class.getSimpleName();
```

```
    private static final int NO_ID = -99;
```

```
    private static final String NO_WORD = "";
```

```
    private EditText mEditWordView;
```

```
    // Unique tag for the intent reply.
```

```
    public static final String EXTRA_REPLY = "com.example.android.wordlistsql.REPLY";
```

```
    int mId = MainActivity.WORD_ADD;
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_edit_word);
```

```

mEditWordView = (EditText) findViewById(R.id.edit_word);

// Get data sent from calling activity.
Bundle extras = getIntent().getExtras();

// If we are passed content, fill it in for the user to edit.
if (extras != null) {
    int id = extras.getInt(WordListAdapter.EXTRA_ID, NO_ID);
    String word = extras.getString(WordListAdapter.EXTRA_WORD, NO_WORD);
    if (id != NO_ID && word != NO_WORD) {
        mId = id;
        mEditWordView.setText(word);
    }
} // Otherwise, start with empty fields.
}

public void returnReply(View view) {
    String word = ((EditText) findViewById(R.id.edit_word)).getText().toString();

    Intent replyIntent = new Intent();
    replyIntent.putExtra(EXTRA_REPLY, word);
    replyIntent.putExtra(WordListAdapter.EXTRA_ID, mId);
    setResult(RESULT_OK, replyIntent);
    finish();
}
}

```

MyButtonOnClickListener.java:

```
package com.example.wordlistsqlinteractivewithsearch;
```

```
import android.view.View;
```

```

public class MyButtonOnClickListener implements View.OnClickListener {
    private static final String TAG = View.OnClickListener.class.getSimpleName();

    int id;
    String word;

    public MyButtonOnClickListener(int id, String word) {
        this.id = id;
        this.word = word;
    }

    public void onClick(View v) {
        // Implemented in WordListAdapter
    }
}

```

```
}  
  
}
```

SearchActivity.java:

```
package com.example.wordlistsqlinteractivewithsearch;  
import android.database.Cursor;  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SearchActivity extends AppCompatActivity {  
  
    private static final String TAG = EditWordActivity.class.getSimpleName();  
  
    private WordListOpenHelper mDB;  
  
    private EditText mEditWordView;  
    private TextView mTextView;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_search);  
  
        mDB = new WordListOpenHelper(this);  
  
        mEditWordView = ((EditText) findViewById(R.id.search_word));  
        mTextView = ((TextView) findViewById(R.id.search_result));  
    }  
  
    // Click handler for Search button.  
    public void showResult(View view) {  
        String word = mEditWordView.getText().toString();  
        mTextView.setText("Result for " + word + ":\n\n");  
  
        // Search for the word in the database.  
        Cursor cursor = mDB.search(word);  
        // You must move the cursor to the first item.  
        cursor.moveToFirst();  
        // Only process a non-null cursor with rows.
```



```

if (cursor != null & cursor.getCount() > 0) {
    int index;
    String result;
    // Iterate over the cursor, while there are entries.
    do {
        // Don't guess at the column index. Get the index for the named column.
        index = cursor.getColumnIndex(WordListOpenHelper.KEY_WORD);
        // Get the value from the column for the current cursor.
        result = cursor.getString(index);
        // Add result to what's already in the text view.
        mTextView.append(result + "\n");
    } while (cursor.moveToNext());
    cursor.close();
} else {
    mTextView.append(getString(R.string.no_result));
}
}
}

```

WordItem.java:

```
package com.example.wordlistsqlinteractivewithsearch;
```

```

public class WordItem {
    private int mId;
    private String mWord;

    public WordItem() {}

    public int getId() {
        return this.mId;
    }

    public String getWord() {
        return this.mWord;
    }

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

    public void setWord(String word) {
        this.mWord = word;
    }
}

```

WordListAdapter.java:

```
package com.example.wordlistsqlinteractivewithsearch;
```

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
```

```
import androidx.recyclerview.widget.RecyclerView;
```

```
public class WordListAdapter extends
RecyclerView.Adapter<WordListAdapter.WordViewHolder> {
```

```
    /**
```

```
     * Custom view holder with a text view and two buttons.
```

```
    */
```

```
    class WordViewHolder extends RecyclerView.ViewHolder {
```

```
        public final TextView wordItemView;
```

```
        Button delete_button;
```

```
        Button edit_button;
```

```
        public WordViewHolder(View itemView) {
```

```
            super(itemView);
```

```
            wordItemView = (TextView) itemView.findViewById(R.id.word);
```

```
            delete_button = (Button) itemView.findViewById(R.id.delete_button);
```

```
            edit_button = (Button) itemView.findViewById(R.id.edit_button);
```

```
        }
```

```
    }
```

```
    private static final String TAG = WordListAdapter.class.getSimpleName();
```

```
    public static final String EXTRA_ID = "ID";
```

```
    public static final String EXTRA_WORD = "WORD";
```

```
    public static final String EXTRA_POSITION = "POSITION";
```

```
    private final LayoutInflater mInflater;
```

```
    WordListOpenHelper mDB;
```

```
    Context mContext;
```

```
    public WordListAdapter(Context context, WordListOpenHelper db) {
```

```
        mInflater = LayoutInflater.from(context);
```

```
        mContext = context;
```

```

        mDB = db;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.inflate(R.layout.wordlist_item, parent, false);
        return new ViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        // Keep a reference to the view holder for the click listener
        final ViewHolder h = holder; // needs to be final for use in callback

        WordItem current = mDB.query(position);
        holder.wordItemView.setText(current.getWord());

        // Attach a click listener to the DELETE button.
        holder.delete_button.setOnClickListener(new MyButtonOnClickListener(
            current.getId(), null) {

            @Override
            public void onClick(View v ) {
                // Remove from the database.
                int deleted = mDB.delete(id);
                if (deleted >= 0) {
                    // Redisplay the view.
                    notifyItemRemoved(h.getAdapterPosition());
                }
            }
        });

        // Attach a click listener to the EDIT button.
        holder.edit_button.setOnClickListener(new MyButtonOnClickListener(
            current.getId(), current.getWord()) {

            @Override
            public void onClick(View v) {
                Intent intent = new Intent(mContext, EditWordActivity.class);

                intent.putExtra(EXTRA_ID, id);
                intent.putExtra(EXTRA_POSITION, h.getAdapterPosition());
                intent.putExtra(EXTRA_WORD, word);

                // Start an empty edit activity.
                ((Activity) mContext).startActivityForResult(intent, MainActivity.WORD_EDIT);
            }
        });
    }

```

```

    }
    });
}

@Override
public int getItemCount() {
    return (int) mDB.count();
}
}

```

WordListOpenHelper.java:

```
package com.example.wordlistsqlinteractivewithsearch;
```

```

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

```

```
public class WordListOpenHelper extends SQLiteOpenHelper {
```

```
    private static final String TAG = WordListOpenHelper.class.getSimpleName();
```

```
    // Declaring all these as constants makes code a lot more readable and looking like SQL.
```

```
    // Version has to be 1 first time or app will crash.
```

```
    private static final int DATABASE_VERSION = 1;
```

```
    private static final String WORD_LIST_TABLE = "word_entries";
```

```
    private static final String DATABASE_NAME = "wordlist";
```

```
    // Column names...
```

```
    public static final String KEY_ID = "_id";
```

```
    public static final String KEY_WORD = "word";
```

```
    // ... and a string array of columns.
```

```
    private static final String[] COLUMNS =
        {KEY_ID, KEY_WORD};
```

```
    // Build the SQL query that creates the table.
```

```
    private static final String WORD_LIST_TABLE_CREATE =
        "CREATE TABLE " + WORD_LIST_TABLE + " (" +
```

KEY_ID + " INTEGER PRIMARY KEY, " + // will auto-increment if no value
passed

KEY_WORD + " TEXT);";

```
private SQLiteDatabase mWritableDB;  
private SQLiteDatabase mReadableDB;
```

```
public WordListOpenHelper(Context context) {  
    super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    Log.d(TAG, "Construct WordListOpenHelper");  
}
```

@Override

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL(WORD_LIST_TABLE_CREATE);  
    fillDatabaseWithData(db);  
}
```

```
public void fillDatabaseWithData(SQLiteDatabase db) {
```

```
    String[] words = {"Android", "Adapter", "ListView", "AsyncTask", "Android Studio",  
        "SQLiteDatabase", "SQLOpenHelper", "Data model", "ViewHolder",  
        "Android Performance", "OnClickListener"};
```

```
    // Create a container for the data.
```

```
    ContentValues values = new ContentValues();
```

```
    for (int i=0; i < words.length; i++) {  
        // Put column/value pairs for current row into the container.  
        values.put(KEY_WORD, words[i]); // put() overrides existing values.  
        // Insert the row.  
        db.insert(WORD_LIST_TABLE, null, values);  
    }  
}
```

```
public Cursor search(String searchString) {  
    String[] columns = new String[]{KEY_WORD};  
    String where = KEY_WORD + " LIKE ?";  
    searchString = "%" + searchString + "%";  
    String[] whereArgs = new String[]{searchString};
```

```
    Cursor cursor = null;
```

```
    try {  
        if (mReadableDB == null) {  
            mReadableDB = getReadableDatabase();  
        }
```

```

        cursor = mReadableDB.query(WORD_LIST_TABLE, columns, where, whereArgs, null,
null, null);
    } catch (Exception e) {
        Log.d(TAG, "SEARCH EXCEPTION! " + e); // Just log the exception
    }
    return cursor;
}

```

```

@SuppressLint("Range")
public WordItem query(int position) {
    String query = "SELECT * FROM " + WORD_LIST_TABLE +
        " ORDER BY " + KEY_WORD + " ASC " +
        "LIMIT " + position + ",1";

```

```

    Cursor cursor = null;
    WordItem entry = new WordItem();

```

```

    try {
        if (mReadableDB == null) {
            mReadableDB = getReadableDatabase();
        }
        cursor = mReadableDB.rawQuery(query, null);
        cursor.moveToFirst();
        entry.setId(cursor.getInt(cursor.getColumnIndex(KEY_ID)));
        entry.setWord(cursor.getString(cursor.getColumnIndex(KEY_WORD)));
    } catch (Exception e) {
        Log.d(TAG, "QUERY EXCEPTION! " + e); // Just log the exception
    } finally {
        // Must close cursor and db now that we are done with it.
        cursor.close();
        return entry;
    }
}

```

```

public long count() {
    if (mReadableDB == null) {
        mReadableDB = getReadableDatabase();
    }
    return DatabaseUtils.queryNumEntries(mReadableDB, WORD_LIST_TABLE);
}

```

```

public long insert(String word) {
    long newId = 0;
    ContentValues values = new ContentValues();
    values.put(KEY_WORD, word);
    try {

```

```

        if (mWritableDB == null) {
            mWritableDB = getWritableDatabase();
        }
        newId = mWritableDB.insert(WORD_LIST_TABLE, null, values);
    } catch (Exception e) {
        Log.d(TAG, "INSERT EXCEPTION! " + e);
    }
    return newId;
}

public int update(int id, String word) {
    int mNumberOfRowsUpdated = -1;
    try {
        if (mWritableDB == null) {
            mWritableDB = getWritableDatabase();
        }
        ContentValues values = new ContentValues();
        values.put(KEY_WORD, word);

        mNumberOfRowsUpdated = mWritableDB.update(WORD_LIST_TABLE, //table to
change
        values, // new values to insert
        KEY_ID + " = ?", // selection criteria for row (in this case, the _id column)
        new String[]{String.valueOf(id)}); //selection args; the actual value of the id

    } catch (Exception e) {
        Log.d (TAG, "UPDATE EXCEPTION! " + e);
    }
    return mNumberOfRowsUpdated;
}

public int delete(int id) {
    int deleted = 0;
    try {
        if (mWritableDB == null) {
            mWritableDB = getWritableDatabase();
        }
        deleted = mWritableDB.delete(WORD_LIST_TABLE, //table name
        KEY_ID + " =?", new String[]{String.valueOf(id)});
    } catch (Exception e) {
        Log.d (TAG, "DELETE EXCEPTION! " + e);    }
    return deleted;
}

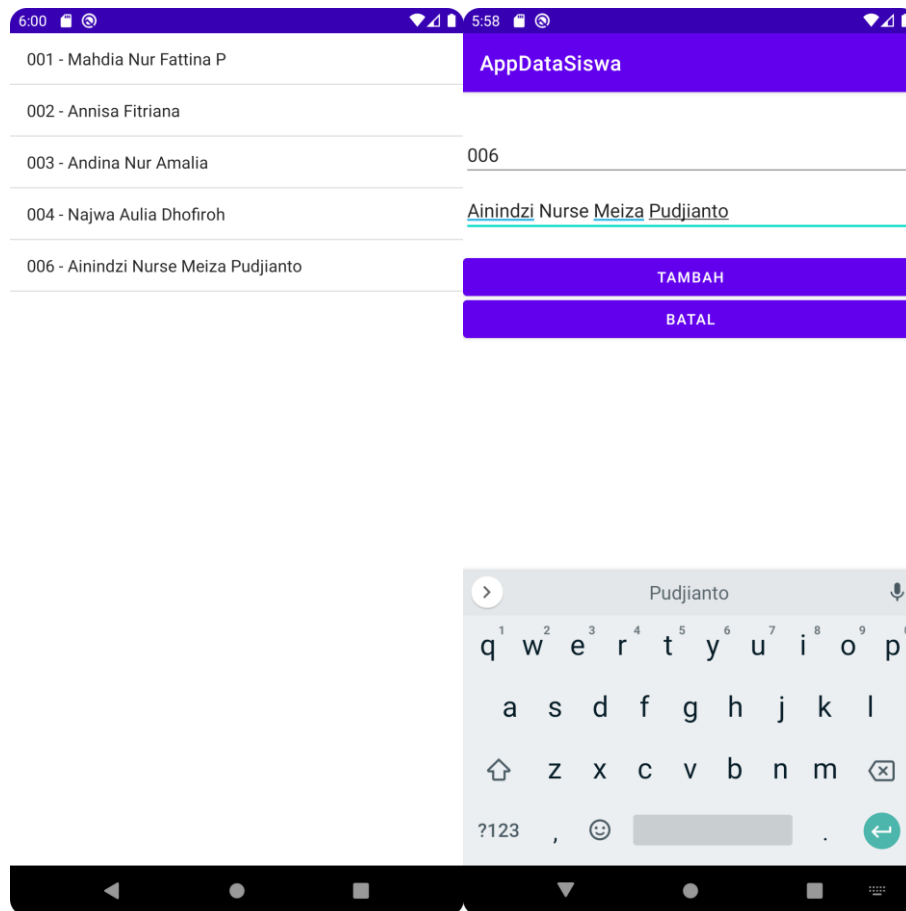
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

```

```
Log.w(WordListOpenHelper.class.getName(),
    "Upgrading database from version " + oldVersion + " to "
    + newVersion + ", which will destroy all old data");
db.execSQL("DROP TABLE IF EXISTS " + WORD_LIST_TABLE);
onCreate(db);
}
}
```


3. App Data Siswa

Hasil Program:



Source Code:

activity_main.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=".MainActivity">

    <ListView
        android:id="@+id/simple_list_item_1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</RelativeLayout>
```

activity_tambah.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"
    android:orientation="horizontal"
    tools:context=".TambahActivity">
```

```
<EditText
    android:id="@+id/editNis"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="37dp"
    android:hint="Input Nis" />
```

```
<EditText
    android:id="@+id/editNama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="90dp"
    android:hint="Input Name" />
```

```
<Button
    android:id="@+id/btnTambah"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="152dp"
    android:layout_weight="0.5"
    android:text="Tambah" />
```

```
<Button
    android:id="@+id/btnBatal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="192dp"
    android:layout_weight="0.5"
    android:text="Batal" />
```

</RelativeLayout>

activity_update.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=".UpdateActivity">
```

```
    <EditText
        android:id="@+id/editNis"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="37dp"
        android:hint="Input Nis" />
```

```
    <EditText
        android:id="@+id/editNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="90dp"
        android:hint="Input Name" />
```

```
    <Button
        android:id="@+id/btnUpdate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="152dp"
        android:layout_weight="0.5"
        android:text="Update" />
```

```
    <Button
        android:id="@+id/btnBatal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="192dp"
```

```
        android:layout_weight="0.5"
        android:text="Batal" />
```

```
</RelativeLayout>
```

MainActivity.java:

```
package com.example.appdatasiswa;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.app.ListActivity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.util.Log;
```

```
import android.view.ContextMenu;
```

```
import android.view.MenuItem;
```

```
import android.view.View;
```

```
import android.widget.AdapterView;
```

```
import android.widget.AdapterView;
```

```
import java.util.List;
```

```
public class MainActivity extends ListActivity {
```

```
    String dataSiswa[] = null;
```

```
    String dS[] = null;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        //setContentView(R.layout.activity_main);
```

```
        //tambah siswa
```

```
        DatabaseHandler db = new DatabaseHandler(this);
```

```
        //membaca semua siswa
```

```
        Log.d("Baca Siswa: ", "Membaca Semua Data Siswa...");
```

```
        List<Siswa> siswa = db.getSemuaSiswa();
```

```
        dataSiswa = new String[siswa.size()];
```

```
        dS = new String[siswa.size()];
```

```
        int i = 0;
```

```
        for (Siswa s : siswa) {
```

```
            String log = "NIS: " + s.getNis() + " , Nama: " + s.getNama();
```

```
            // Writing Contacts to log
```

```
            Log.d("Name: ", log);
```

```
            dataSiswa[i] = s.getNis() + " - " + s.getNama();
```

```

        dS[i] = s.getNis();
        i++;
    }

    //cek data
    if(i == 0)
    {
        Log.d("Tambah Siswa: ", "Menambah Data Siswa..");
        db.addSiswa(new Siswa("001", "Ghiyatsi Miftahur Rahmat"));
        db.addSiswa(new Siswa("002", "Annisa Fitriana"));
        db.addSiswa(new Siswa("003", "Andina Nur Amalia"));
        db.addSiswa(new Siswa("004", "Najwa Aulia Dhofiroh"));
    }

    setListAdapter(new ArrayAdapter<Object>(this, android.R.layout.simple_list_item_1,
dataSiswa));
    registerForContextMenu(getListView());
}

@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.setHeaderTitle(" Action");
    menu.add(0,0,0,"Tambah");
    menu.add(0,1,1,"Hapus");
    menu.add(0,2,2,"Update");
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    try{
        switch(item.getItemId()){
            case 0:
                Class c = Class.forName("com.example.appdatasiswa.TambahActivity");
                Intent i = new Intent(MainActivity.this, c);
                startActivity(i);
                break;
            case 1:
                DatabaseHandler db = new DatabaseHandler(this);
                AdapterView.AdapterContextMenuInfo info =
(AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
                String[] args = {String.valueOf(info.id)};
                int xpos=Integer.parseInt(args[0]);
                db.deleteRow(dS[xpos]);
                Class c1 = Class.forName("com.example.appdatasiswa.MainActivity");

```

```

        Intent i1 = new Intent(MainActivity.this, c1);
        startActivity(i1);
        break;
    case 2:
        DatabaseHandler db2 = new DatabaseHandler(this);
        AdapterView.AdapterContextMenuInfo info2 =
(AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
        String[] args2={String.valueOf(info2.id)};
        Log.d("args0 : ",args2[0]);
        int xpos2 = Integer.parseInt(args2[0]);
        db2.getSiswa(dS[xpos2]);
        String namax = db2.getSiswa(dS[xpos2]).getNama();
        Intent i2 = new Intent(this, UpdateActivity.class);
        Bundle bun = new Bundle();
        bun.putString("nis", dS[xpos2]);
        bun.putString("nama", namax);
        i2.putExtras(bun);
        startActivity(i2); break;
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
return true;
}
}

```

DatabaseHandler.java:

```
package com.example.appdatisiswa;
```

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
public class DatabaseHandler extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION= 1;
    // NamaDatabase
    private static final String DATABASE_NAME= "Sekolah";
    // NamaTable
    private static final String TABLE_SISWA= "Siswa";
    // NamaKolomTable Siswa
    private static final String KEY_NIS= "nis";

```

```

private static final String KEY_NAMA= "nama";

public DatabaseHandler(Context context) {
    // TODOAuto-generated constructor stub
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
@Override
public void onCreate(SQLiteDatabase db) {
    // TODOAuto-generated method stub
    String query_table_siswa = "CREATE TABLE "+
        TABLE_SISWA+ "(" + KEY_NIS+ " TEXT PRIMARY KEY,"+ KEY_NAMA+ "
TEXT)";
    db.execSQL(query_table_siswa);
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODOAuto-generated method stub
    // Drop older table if existed
    db.execSQL("DROP TABLE IF EXISTS "+ TABLE_SISWA);
    // Create tables again
    onCreate(db);
}

// MenambahSiswaBaru
public void addSiswa(Siswa siswa) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_NIS, siswa.getNis());
    values.put(KEY_NAMA, siswa.getNama());
    // Inserting Row
    db.insert(TABLE_SISWA, null, values);
    db.close();
}

// MembacaSiswa
public Siswa getSiswa(String nis) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE_SISWA, new String[] {KEY_NIS, KEY_NAMA},
KEY_NIS+ "=?", new String[] {nis}, null, null, null, null);
    if(cursor != null)
        cursor.moveToFirst();
    Siswa siswa = new Siswa(cursor.getString(0), cursor.getString(1));
    return siswa;
}

// MembacaSemuaSiswa

```

```

public List<Siswa> getSemuaSiswa() {
    List<Siswa> siswaList = new ArrayList<Siswa>();
    String query_select_siswa = "SELECT * FROM " + TABLE_SISWA;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query_select_siswa, null);
    if(cursor.moveToFirst()) {
        do{
            Siswa siswa = new Siswa(cursor.getString(0), cursor.getString(1));
            siswaList.add(siswa);
        } while(cursor.moveToNext());
    }
    return siswaList;
}

public void deleteSiswa(Siswa siswa) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_SISWA, KEY_NIS+ " = ?", new String[]{ siswa.getNis()});
    db.delete(TABLE_SISWA, KEY_NIS+ "=" +siswa.getNis()+"" ,null);
    db.close();
    System.out.println("Data terhapus " +siswa.getNis());
}

public void deleteRow(String xnis) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_SISWA, KEY_NIS+ "=" + xnis+"" ,null);
    db.close();
    System.out.println("Data terhapus " +xnis);
}

public void updateMethod(String nis, String nama){
    SQLiteDatabase db = this.getWritableDatabase();
    db.execSQL("update "+TABLE_SISWA+" set nama='"+nama+ "' where nis='"+nis+"'");
    db.close();
    System.out.println("Data sudah di update " +nis);
}
}

```

Siswa.java:

```
package com.example.appdatasiswa;
```

```

public class Siswa {
    private String nis;
    private String nama;
    public Siswa() {
    }
}

```



```

public Siswa(String nis, String nama) {
    this.nis= nis;
    this.nama= nama;
}
public String getNis() {
    return nis;
}
public void setNis(String nis) {
    this.nis= nis;
}
public String getNama() {
    return nama;
}
public void setNama(String nama) {
    this.nama= nama;
}
}

```

TambahActivity.java:

```

package com.example.appdatasiswa;

```

```

import androidx.appcompat.app.AppCompatActivity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

```

```

public class TambahActivity extends AppCompatActivity {

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah);

```

```

        final DatabaseHandler db = new DatabaseHandler(this);
        final EditText editNis = (EditText) findViewById(R.id.editNis);
        final EditText editNama = (EditText) findViewById(R.id.editNama);
        Button btnTambah = (Button) findViewById(R.id.btnTambah);
        Button btnBatal = (Button) findViewById(R.id.btnBatal);

```

```

        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String nis = editNis.getText().toString();

```

```

        String nama = editNama.getText().toString();
        db.addSiswa(new Siswa(nis, nama));
        editNis.setText("");
        editNama.setText("");
        try {
            Class c = Class.forName("com.example.appdatasiswa.MainActivity");
            Intent i = new Intent(TambahActivity.this, c);
            startActivity(i);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
});

btnBatal.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            Class c = Class.forName("com.example.appdatasiswa.MainActivity");
            Intent i = new Intent(TambahActivity.this, c);
            startActivity(i);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
});
}
}

```

UpdateActivity.java:

```

package com.example.appdatasiswa;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class UpdateActivity extends AppCompatActivity {

    private String xnis, xnama;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_update);

final DatabaseHandler db = new DatabaseHandler(this);
final EditText editNis = (EditText) findViewById(R.id.editNis);
final EditText editNama = (EditText) findViewById(R.id.editNama);
Button btnTambah = (Button) findViewById(R.id.btnUpdate);
Button btnBatal = (Button) findViewById(R.id.btnBatal);

//ambil data siswa dari extras
Bundle bun = getIntent().getExtras();
xnis = bun.getString("nis");
xnama = bun.getString("nama");

//masukkan data-data siswa tersebut ke field editor
editNis.setText(xnis);
editNama.setText(xnama);

btnTambah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String nis = editNis.getText().toString();
        String nama = editNama.getText().toString();
        //db.addSiswa(new Siswa(nis, nama));
        //db.updateData(nis,nama);
        db.updateMethod(nis,nama);
        editNis.setText("");
        editNama.setText("");

        try {
            Class c = Class.forName("com.example.sqliteku.MainActivity");
            Intent i = new Intent(UpdateActivity.this, c);
            startActivity(i);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
});

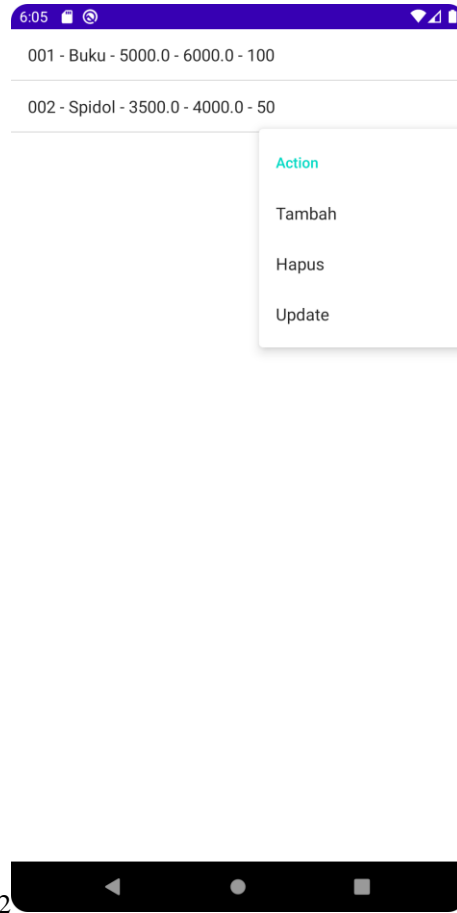
btnBatal.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            Class c = Class.forName("com.example.sqliteku.MainActivity");
            Intent i = new Intent(UpdateActivity.this, c);
            startActivity(i);
        }
    }
});

```

```
        } catch (ClassNotFoundException e) {  
            e.printStackTrace();  
        }  
    }  
});  
}  
}
```

4. AppDataBarang

Hasil Program:



Code Program:

activity_main.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=".MainActivity">
```

```
<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

```
</RelativeLayout>
```

activity_tambah.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=".TambahActivity">
```

```
<EditText
    android:id="@+id/editKode"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="37dp"
    android:hint="Input Kode" />
```

```
<EditText
    android:id="@+id/editNama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="90dp"
    android:hint="Input Nama"/>
```

```
<EditText
    android:id="@+id/editBeli"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="140dp"
    android:hint="Input Harga Beli"/>
```

```
<EditText
```

```
android:id="@+id/editJual"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginTop="192dp"
android:hint="Input Harga Jual"/>
```

```
<EditText
    android:id="@+id/editStok"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="245dp"
    android:hint="Input Jumlah Stok"/>
```

```
<Button
    android:id="@+id/btnTambah"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="292dp"
    android:layout_weight="0.5"
    android:text="Tambah" />
```

```
<Button
    android:id="@+id/btnBatal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="340dp"
    android:layout_weight="0.5"
    android:text="Batal" />
```

</RelativeLayout>

activity_update.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=".UpdateActivity">
```

```
<EditText
    android:id="@+id/editKode"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="37dp"
    android:hint="Input Kode"
    android:enabled="false"/>
```

```
<EditText
    android:id="@+id/editNama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="90dp"
    android:hint="Input Nama"/>
```

```
<EditText
    android:id="@+id/editBeli"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="140dp"
    android:hint="Input Harga Beli"/>
```

```
<EditText
    android:id="@+id/editJual"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="192dp"
    android:hint="Input Harga Jual"/>
```

```
<EditText
    android:id="@+id/editStok"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="245dp"
```



```

        android:hint="Input Jumlah Stok"/>

<Button
    android:id="@+id/btnUpdate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="292dp"
    android:text="Update"/>

<Button
    android:id="@+id/btnBatal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="335dp"
    android:text="Batal"/>
</RelativeLayout>

```

MainActivity.java:

```

package com.example.appdatabarang;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.util.List;

public class MainActivity extends ListActivity {

    String dataBarang[] = null;
    String dS[] = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

//setContentView(R.layout.activity_main);
// Tambah Barang
DatabaseHandler db = new DatabaseHandler(this);
// Membaca Semua Barang
Log.d("Baca Barang: ", "Membaca Semua Data Barang..");
List<Barang> barang = db.getAllBarang();
dataBarang= new String[barang.size()];
dS= new String[barang.size()];
int i=0;
for(Barang s : barang) {
    String log = "Kode : " + s.getKdbrg() + ",Nama Barang: " +
        s.getNmbrg();
    Log.d("Name : ", log);
    dataBarang[i] = s.getKdbrg() + " - " + s.getNmbrg()+
        " - " + s.getHrgbeli()+" - " +s.getHrgjual()+" - " +s.getStok();
    dS[i] = s.getKdbrg();
    i++;
}
//cek data
if(i==0)
{
    Log.d("Tambah Barang: ", "Menambah Data Barang..");
    db.addBarang(new Barang("001", "Buku",5000,6000,100));
    db.addBarang(new Barang("002", "Spidol",3500,4000,50));
}
setListAdapter(new ArrayAdapter<Object>(this,
    android.R.layout.simple_list_item_1, dataBarang));
registerForContextMenu(getListView());
}
@Override
public void onCreateContextMenu(ContextMenu menu, View
    v, ContextMenu.ContextMenuInfo menuInfo) {
// TODOAuto-generated method stub
super.onCreateContextMenu(menu, v, menuInfo);
menu.setHeaderTitle("Action");
menu.add(0,0,0,"Tambah");
menu.add(0,1,1,"Hapus");
menu.add(0,2,2,"Update");
}
@Override
public boolean onContextItemSelected(MenuItem item) {
try{
    switch(item.getItemId()){
        case 0:{
            Class c = Class.forName("com.example.appdatabarang.TambahActivity");
            Intent i = new Intent(MainActivity.this, c);

```

```

        startActivity(i);
        break;
    }
    case 1:{
        DatabaseHandler db = new DatabaseHandler(this);
        AdapterView.AdapterContextMenuInfo info =
(AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
        String[] args = {String.valueOf(info.id)};
        int xpos=Integer.parseInt(args[0]);
        db.deleteRow(dS[xpos]);
        Class c = Class.forName("com.example.appdatabarang.MainActivity");
        Intent i = new Intent(MainActivity.this, c);
        startActivity(i);
        break;
    }
    case 2:{
        DatabaseHandler db = new DatabaseHandler(this);
        AdapterView.AdapterContextMenuInfo info
=(AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
        String[] args ={String.valueOf(info.id)};
        Log.d("args0 : ",args[0]);
        int xpos = Integer.parseInt(args[0]);
        db.getBarang(dS[xpos]);
        String kodex=db.getBarang(dS[xpos]).getKdbrg();
        String namax=db.getBarang(dS[xpos]).getNmbrg();
        double beli=db.getBarang(dS[xpos]).getHrgbeli();
        double jual=db.getBarang(dS[xpos]).getHrgjual();
        int stok=db.getBarang(dS[xpos]).getStok();
        Intent i = new Intent(this, UpdateActivity.class);
        Bundle bun = new Bundle();
        bun.putString("kdbrg", kodex);
        bun.putString("nmbrg", namax);
        bun.putDouble("hrgbeli", beli);
        bun.putDouble("hrgjual", jual);
        bun.putInt("stok", stok);
        i.putExtras(bun);
        startActivity(i);
        break;
    }
    }
} catch(ClassNotFoundException e) {
    e.printStackTrace();
}
return true;
}
}

```

```

TambahActivity.java:
package com.example.appdatabarang;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class TambahActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah);

        final DatabaseHandler db = new DatabaseHandler(this);
        final EditText editKode = (EditText) findViewById(R.id.editKode);
        final EditText editNama = (EditText) findViewById(R.id.editNama);
        final EditText editBeli = (EditText) findViewById(R.id.editBeli);
        final EditText editJual = (EditText) findViewById(R.id.editJual);
        final EditText editStok = (EditText) findViewById(R.id.editStok);
        Button btnTambah = (Button) findViewById(R.id.btnTambah);
        Button btnBatal = (Button) findViewById(R.id.btnBatal);

        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String kode = editKode.getText().toString();
                String nama = editNama.getText().toString();
                double beli = Double.parseDouble(editBeli.getText().toString());
                double jual = Double.parseDouble(editJual.getText().toString());
                int stok = Integer.parseInt(editStok.getText().toString());
                db.addBarang(new Barang(kode, nama, beli, jual, stok));
                editKode.setText("");
                editNama.setText("");
                editBeli.setText("");
                editJual.setText("");
                editStok.setText("");
                try {
                    Class c = Class.forName("com.example.appdatabarang.MainActivity");
                    Intent i = new Intent(TambahActivity.this, c);
                    startActivity(i);
                }
            }
        });
    }
}

```

```

        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
});

btnBatal.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            Class c = Class.forName("com.example.appdatabarang.MainActivity");
            Intent i = new Intent(TambahActivity.this, c);
            startActivity(i);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
});
}
}

```

UpdateActivity.java:

```
package com.example.appdatabarang;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
public class UpdateActivity extends AppCompatActivity {
```

```
    private String xkode, xnama;
```

```
    private double xbeli, xjual;
```

```
    private int xstok;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_update);
```

```
        final DatabaseHandler db = new DatabaseHandler(this);
```

```
        final EditText editKode = (EditText) findViewById(R.id.editKode);
```

```
        final EditText editNama = (EditText) findViewById(R.id.editNama);
```

```

final EditText editBeli = (EditText) findViewById(R.id.editBeli);
final EditText editJual = (EditText) findViewById(R.id.editJual);
final EditText editStok = (EditText) findViewById(R.id.editStok);
Button btnUpdate = (Button) findViewById(R.id.btnUpdate);
Button btnBatal = (Button) findViewById(R.id.btnBatal);

// ambil data barang dari extras
Bundle bun = this.getIntent().getExtras();
xkode = bun.getString("kdbrg");
xnama = bun.getString("nmbrg");
xbeli = bun.getDouble("hrgbeli");
xjual = bun.getDouble("hrgjual");
xstok = bun.getInt("stok");

// masukkan data-data barang tersebut ke field editor
editKode.setText(xkode);
editNama.setText(xnama);
editBeli.setText(String.valueOf(xbeli));
editJual.setText(String.valueOf(xjual));
editStok.setText(String.valueOf(xstok));

btnUpdate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String kode = editKode.getText().toString();
        String nama = editNama.getText().toString();
        double beli = Double.parseDouble(editBeli.getText().toString());
        double jual = Double.parseDouble(editJual.getText().toString());
        int stok = Integer.parseInt(editStok.getText().toString());
        db.updateMethod(kode, nama, beli, jual, stok);
        editKode.setText("");
        editNama.setText("");
        editBeli.setText("");
        editJual.setText("");
        editStok.setText("");
        try{
            Class c = Class.forName("com.example.appdatabarang.MainActivity");
            Intent i = new Intent(UpdateActivity.this, c);
            startActivity(i);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
});

btnBatal.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    try{
        Class c = Class.forName("com.example.appdatabarang.MainActivity");
        Intent i = new Intent(UpdateActivity.this, c);
        startActivity(i);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
});
}
}

```

Barang.Java:

```
package com.example.appdatabarang;
```

```

public class Barang {
    private String kdbrg;
    private String nmbrg;
    private double hrgbeli;
    private double hrgjual;
    private int stok;

    public Barang(String kdbrg, String nmbrg, double hrgbeli, double hrgjual, int stok) {
        this.kdbrg = kdbrg;
        this.nmbrg = nmbrg;
        this.hrgbeli = hrgbeli;
        this.hrgjual = hrgjual;
        this.stok = stok;
    }

    public String getKdbrg() { return kdbrg; }
    public void setKdbrg(String kdbrg) { this.kdbrg = kdbrg; }
    public String getNmbrg() { return nmbrg; }
    public void setNmbrg(String nmbrg) { this.nmbrg = nmbrg; }
    public double getHrgbeli() { return hrgbeli; }
    public void setHrgbeli(double hrgbeli) { this.hrgbeli = hrgbeli; }
    public double getHrgjual() { return hrgjual; }
    public void setHrgjual(double hrgjual) { this.hrgjual = hrgjual; }
    public int getStok() { return stok; }
    public void setStok(int stok) { this.stok = stok; }
}

```

DatabaseHandler.java:

```
package com.example.appdatabarang;
```

```

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

import java.util.ArrayList;
import java.util.List;

public class DatabaseHandler extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;

    // Nama database
    private static final String DATABASE_NAME = "Persediaan";
    // Nama tabel
    private static final String TABLE_BARANG = "Barang";
    // Nama kolom tabel
    private static final String KEY_KODE = "kdbrg";
    private static final String KEY_NAMA = "nmbrg";
    private static final String KEY_HRGBELI = "hrgbeli";
    private static final String KEY_HRGJUAL = "hrgjual";
    private static final String KEY_STOK = "stok";

    public DatabaseHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String query_table_barang = "CREATE TABLE " + TABLE_BARANG + "("
            + KEY_KODE + " TEXT PRIMARY KEY,"
            + KEY_NAMA + " TEXT,"
            + KEY_HRGBELI + " DOUBLE,"
            + KEY_HRGJUAL + " DOUBLE,"
            + KEY_STOK + " INTEGER" + ")";
        db.execSQL(query_table_barang);
    }

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

    // Method untuk menambahkan data barang

```



```

public void addBarang(Barang barang) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_KODE, barang.getKdbrg());
    values.put(KEY_NAMA, barang.getNmbrg());
    values.put(KEY_HRGBELI, barang.getHrgbeli());
    values.put(KEY_HRGJUAL, barang.getHrgjual());
    values.put(KEY_STOK, barang.getStok());
    // Menambahkan baris
    db.insert(TABLE_BARANG, null, values);
    db.close();
}

// Method untuk membaca barang
public Barang getBarang(String kode) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE_BARANG, new String[]{KEY_KODE,
        KEY_NAMA, KEY_HRGBELI, KEY_HRGJUAL, KEY_STOK}, KEY_KODE
+ "=?",
        new String[]{kode}, null, null, null, null);
    if (cursor != null)
        cursor.moveToFirst();
    Barang barang = new Barang(cursor.getString(0),
        cursor.getString(1), cursor.getDouble(2),
        cursor.getDouble(3), cursor.getInt(4));
    return barang;
}

// Method untuk membaca semua barang
public List<Barang> getAllBarang() {
    List<Barang> listBarang = new ArrayList<Barang>();
    // Select All Query
    String selectQuery = "SELECT * FROM " + TABLE_BARANG;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);
    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {
            Barang barang = new Barang(cursor.getString(0),
                cursor.getString(1), cursor.getDouble(2),
                cursor.getDouble(3), cursor.getInt(4));
            // Menambahkan barang ke list
            listBarang.add(barang);
        } while (cursor.moveToNext());
    }
    // return barang list
}

```

```

        return listBarang;
    }

    // Method untuk menghapus barang
    public void deleteBarang(Barang barang) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_BARANG, KEY_KODE + " =" +
            barang.getKdbrg() + "", null);
        db.close();
        System.out.println("Barang dengan kode " + barang.getKdbrg() + " berhasil dihapus");
    }

    // Method untuk hapus baris barang
    public void deleteRow(String xkode) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_BARANG, KEY_KODE + " =" +
            xkode + "", null);
        db.close();
        System.out.println("Barang dengan kode " + xkode + " berhasil dihapus");
    }

    // Method untuk mengupdate barang
    public void updateMethod(String kode, String nama, double beli, double jual, double stok){
        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL("update "+TABLE_BARANG+" set nmbrg='"+nama+
            ",hrgbeli='"+beli+",hrgjual='"+jual+",stok='"+stok+
            "' where kdbrg='"+kode+"'");
        db.close();
        System.out.println("Data sudah di update "+kode);
    }
}

```