Mata Kuliah : Pemgraman Mobile – TI – S1

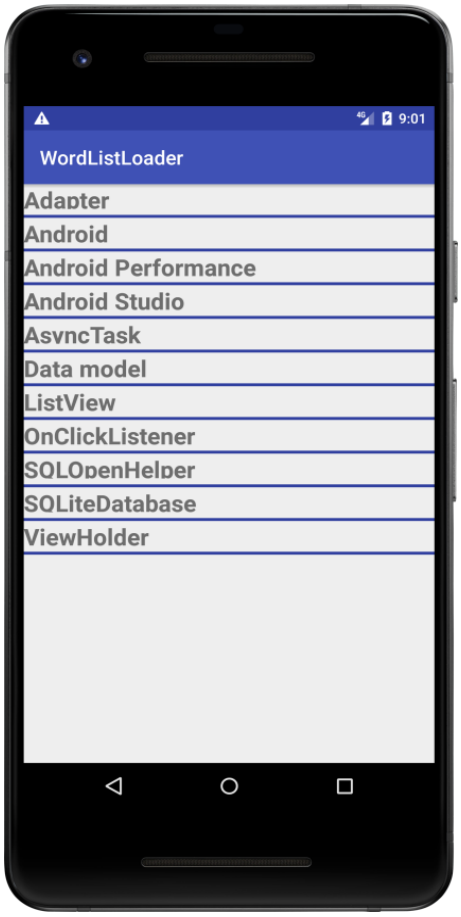
Pertemuan : Minggu 9

NIM : A11.2021.13509

Nama : Ainindzi Nur Meiza Pudjianto

Buat App WordListLoader

Hasil Program :



Source code :

Contract.java

```
package com.example.wordlistloader;

import android.net.Uri;
import android.provider.BaseColumns;

/**
 * Contract class for word list content provider.
 */
public final class Contract {
```

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

    // Prevent class from being instantiated.
    private Contract() {}

    public static final int ALL_ITEMS = -2;
    public static final String COUNT = "count";

    public static final String AUTHORITY =
            "com.android.example.wordlistsqlwithcontentprovider.provider";

    // Only one public table.
    public static final String CONTENT_PATH = "words";

    // Content URI for this table. Returns all items.
    public static final Uri CONTENT_URI = Uri.parse("content://" + AUTHORITY
+ "/" + CONTENT_PATH);

    // URI to get the number of entries.
    public static final Uri ROW_COUNT_URI =
            Uri.parse("content://" + AUTHORITY + "/" + CONTENT_PATH + "/" +
COUNT);

    static final String SINGLE_RECORD_MIME_TYPE =
            "vnd.android.cursor.item/vnd.com.example.provider.words";
    static final String MULTIPLE_RECORDS_MIME_TYPE =
            "vnd.android.cursor.item/vnd.com.example.provider.words";

    /*
     * Constants for the database are moved out of WordListOpenHelper into
the contract.
     * A common way to organize a contract class is to put definitions that
are global to your
     * database in the root level of the class. Then create an inner class
for each table
     * that enumerates its columns.
     */

    public static final String DATABASE_NAME = "wordlist";

    /**
     * Inner class that defines the table contents
     *
     * By implementing the BaseColumns interface, your inner class can
inherit a primary
     * key field called _ID that some Android classes such as cursor adaptors
will expect it to
     * have. It's not required, but this can help your database work
harmoniously with the
     * Android framework.
     */
    public static abstract class WordList implements BaseColumns {

        // Table
        public static final String WORD_LIST_TABLE = "word_entries";
```

```
        // Column names
        public static final String KEY_ID = "_id";
        public static final String KEY_WORD = "word";
    }

}
```

MainActivity.java

```
package com.example.wordlistloader;

import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;

import static com.example.wordlistloader.Contract.CONTENT_PATH;
import static com.example.wordlistloader.Contract.CONTENT_URI;

import androidx.appcompat.app.AppCompatActivity;
import androidx.loader.app.LoaderManager;
import androidx.loader.content.CursorLoader;
import androidx.loader.content.Loader;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

/**
 * Implements a RecyclerView that displays a list of words from a SQL
database using a Loader.
 */
public class MainActivity extends AppCompatActivity implements
LoaderManager.LoaderCallbacks<Cursor> {

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

    private RecyclerView mRecyclerView;
    private WordListAdapter mAdapter;

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

        getSupportLoaderManager().initLoader(0, null, this);

        // Create recycler view.
        mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
        // Create an adapter and supply the data to be displayed.
        mAdapter = new WordListAdapter(this);
        // Connect the adapter with the recycler view.
        mRecyclerView.setAdapter(mAdapter);
        // Give the recycler view a default layout manager.
        mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
    }

    @Override
    public Loader<Cursor> onCreateLoader(int id, Bundle args) {
```

```
        String queryUri = CONTENT_URI.toString();
        String[] projection = new String[] {CONTENT_PATH};
        return new CursorLoader(this, Uri.parse(queryUri), projection, null,
null, null);
    }

    @Override
    public void onLoadFinished(Loader<Cursor> loader, Cursor cursor) {
        mAdapter.setData(cursor);
    }

    @Override
    public void onLoaderReset(Loader<Cursor> loader) {
        mAdapter.setData(null);
    }
}
```

WordItem.java

```
package com.example.wordlistloader;

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.wordlistloader;

import android.content.Context;
import android.database.Cursor;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;
import android.widget.TextView;

import static com.example.wordlistloader.R.id.word;

import androidx.recyclerview.widget.RecyclerView;

/**
 * Simple Adapter for a RecyclerView with click handler for each item in the
ViewHolder.
 */
public class WordListAdapter
        extends RecyclerView.Adapter<WordListAdapter.WordViewHolder> {

    class WordViewHolder extends RecyclerView.ViewHolder {
        public final TextView wordItemView;

        public WordViewHolder(View itemView) {
            super(itemView);
            wordItemView = (TextView) itemView.findViewById(word);
        }
    }

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

    private Cursor mCursor = null;

    private final LayoutInflater mInflater;
    private Context mContext;

    public WordListAdapter(Context context) {
        mInflater = LayoutInflater.from(context);
        this.mContext = context;
    }

    public void setData(Cursor cursor) {
        mCursor = cursor;
        notifyDataSetChanged();
    }

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

    @Override
    public void onBindViewHolder(WordViewHolder holder, int position) {

        String word = "";

        if (mCursor != null) {
            if (mCursor.moveToPosition(position)) {
                int indexWord =
mCursor.getColumnIndex(Contract.WordList.KEY_WORD);
```

```
                word = mCursor.getString(indexWord);
                holder.wordItemView.setText(word);
            } else {
                holder.wordItemView.setText(R.string.error_no_word);
            }
        } else {
            Log.e (TAG, "onBindViewHolder: Cursor is null.");
        }
    }

    @Override
    public int getItemCount() {
        if (mCursor != null) {
            return mCursor.getCount();
        } else {
            return -1;
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

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

</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">

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

    <View
        android:layout_width="match_parent"
        android:layout_height="@dimen/divider_height"
        android:background="@color/colorPrimaryDark" />
</LinearLayout>
```