Praktikum Tugas 3
Fragment dan RecycleView

Mata kuliah Pemrograman IV

Kelas 3C

Disusun Oleh:

Hasna Zahidah 1204068

Dosen Pengampu :

Beri Noviansyah,. , S.T., M.T.

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

POLITEKNIK POS INDONESIA

2021

Implementasi FragmentExample

A. Praktikum

- MainActivity.java

```
package com.example.android.fragmentexample;

import android.os.Bundle;
import android.support.v4.app.FragmentManager;
import
android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.view.View; import
android.widget.Button;
 public class MainActivity extends AppCompatActivity
{

    private Button mButton;
    private boolean isFragmentDisplayed = false;

    // Saved instance state key.
    static final String STATE_FRAGMENT = "state_of_fragment";
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
        // Get the button for opening and closing the fragment.
mButton = findViewById(R.id.open_button);

        // If returning from a configuration change, get the
        // fragment state and set the button text.
        if (savedInstanceState != null) {
isFragmentDisplayed =
savedInstanceState.getBoolean(STATE_FRAGMENT);
if (isFragmentDisplayed) {
                // If the fragment is displayed, change button to
"close".
mButton.setText(R.string.close);
            }
        }
        // Set the click listener for the button.
mButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
if (!isFragmentDisplayed) {
displayFragment();
                } else {
                    closeFragment();
                }
            }
        });
    }

    /**
* This method is called when the user clicks the button
* to open the fragment.
     */
    public void displayFragment() {
// Instantiate the fragment.
        SimpleFragment simpleFragment = SimpleFragment.newInstance();
        // Get the FragmentManager and start a transaction.
FragmentManager fragmentManager =
getSupportFragmentManager();
        FragmentTransaction fragmentTransaction = fragmentManager
.beginTransaction();

        // Add the SimpleFragment.
fragmentTransaction.add(R.id.fragment_container,
simpleFragment).addToBackStack(null).commit();
        // Update the Button text.
        mButton.setText(R.string.close);
        // Set boolean flag to indicate fragment is open.
        isFragmentDisplayed = true;
    }

    /**
```

```
* This method is called when the user clicks the button to
* close the fragment.
     */
    public void closeFragment() {
        // Get the FragmentManager.
```

```
        FragmentManager fragmentManager =
getSupportFragmentManager();
        // Check to see if the fragment is already showing.
SimpleFragment simpleFragment = (SimpleFragment) fragmentManager
                .findFragmentById(R.id.fragment_container);
if (simpleFragment != null) {
            // Create and commit the transaction to remove the
fragment.
            FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
            fragmentTransaction.remove(simpleFragment).commit();
}
        // Update the Button text.
        mButton.setText(R.string.open);
        // Set boolean flag to indicate fragment is closed.
isFragmentDisplayed = false;
    }     public void onSaveInstanceState(Bundle savedInstanceState)
{        // Save the state of the fragment (true=open, false=closed).
savedInstanceState.putBoolean(STATE_FRAGMENT, isFragmentDisplayed);
        super.onSaveInstanceState(savedInstanceState);
    }
}
```

- SimpleFragment.java

```
package com.example.android.fragmentexample;

import android.os.Bundle;
import
android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View; import
android.view.ViewGroup; import
android.widget.RadioGroup; import
android.widget.RatingBar; import
android.widget.TextView; import
android.widget.Toast;

/**
* A simple {@link Fragment} subclass.
* Use the {@link SimpleFragment#newInstance} factory method to
* create an instance of this fragment.
 */
public class SimpleFragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private static final String ARG_PARAM1 = "param1";     private
static final String ARG_PARAM2 = "param2";
    // TODO: Rename and change types of
parameters     private String mParam1;
private String mParam2;     private static final
int YES = 0;     private static final int NO = 1;

    public SimpleFragment() {
```

```
        // Required empty public constructor
    }

    /**
* Use this factory method to create a new instance of
* this fragment using the provided parameters.
     *
* @param param1 Parameter 1.
* @param param2 Parameter 2.
* @return A new instance of fragment SimpleFragment.
     */
    // TODO: Rename and change types and number of parameters
public static SimpleFragment newInstance(String param1, String
param2) {
        SimpleFragment fragment = new
SimpleFragment();         Bundle args = new Bundle();
args.putString(ARG_PARAM1, param1);
args.putString(ARG_PARAM2, param2);
fragment.setArguments(args);         return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState)
{         super.onCreate(savedInstanceState);
if (getArguments() != null) {
             mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment.
final View rootView =
                inflater.inflate(R.layout.fragment_simple, container,
false);
        final RadioGroup radioGroup =
rootView.findViewById(R.id.radio_group);
final RatingBar ratingBar =
                rootView.findViewById(R.id.ratingBar);
        // Set the radioGroup onCheckedChanged listener.
radioGroup.setOnCheckedChangeListener(
                new RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int
checkedId) {
                View radioButton =
radioGroup.findViewById(checkedId);
                int index = radioGroup.indexOfChild(radioButton);
TextView textView =
                        rootView.findViewById(R.id.fragment_header);
switch (index) {
                    case YES: // User chose "Yes."
```

```
                        textView.setText(R.string.yes_message);
break;
                        case NO: // User chose "No."
                            textView.setText(R.string.no_message);
```

```
                            break;
default: // No choice made.
                                // Do nothing.
                                break;
                }
            }
        });

        // Set the rating bar onCheckedChanged listener.
//        ratingBar.setOnRatingBarChangeListener
//                (new RatingBar.OnRatingBarChangeListener() {
//                    @Override
//                    public void onRatingChanged(RatingBar
ratingBar,
//                                                float rating,
boolean fromUser) {
//                        // Get rating and show Toast with rating.
//                        String myRating =
(getString(R.string.my_rating) +
//
String.valueOf(ratingBar.getRating()));
//                        Toast.makeText(getContext(), myRating,
//                                Toast.LENGTH_SHORT).show();
//                    }
//                });

// Return the View for the fragment's UI.
return rootView;

}
    public static SimpleFragment newInstance() {
return new SimpleFragment();
    }
}
```

- Activity_Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    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="com.example.android.fragmentexample.MainActivity">
    <ImageView
        android:id="@+id/imageView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/image_left_margin"
android:layout_marginStart="@dimen/image_left_margin"
android:layout_marginTop="@dimen/body_top_margin"
android:scaleType="centerCrop"
        app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/beatles_anthology_box"/>
    <TextView
```

```
        android:id="@+id/title"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginEnd="@dimen/margin_wide"
android:layout_marginLeft="@dimen/margin_wide"
android:layout_marginRight="@dimen/margin_wide"
android:layout_marginStart="@dimen/margin_wide"
android:text="@string/title1"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toRightOf="@+id/imageView"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="@+id/imageView"/>
    <ScrollView
        android:layout_width="0dp"
android:layout_height="0dp"
        android:layout_marginRight="@dimen/standard_margin"
android:layout_marginEnd="@dimen/standard_margin"
app:layout_constraintRight_toRightOf="parent"
android:layout_marginTop="@dimen/standard_margin"
app:layout_constraintTop_toBottomOf="@+id/title"
android:layout_marginLeft="0dp"
android:layout_marginStart="0dp"
        app:layout_constraintLeft_toLeftOf="@+id/title"
app:layout_constraintBottom_toBottomOf="parent"
android:layout_marginBottom="@dimen/standard_margin">
        <TextView
            android:id="@+id/article"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/article1"
tools:layout_editor_absoluteX="8dp"
tools:layout_editor_absoluteY="288dp"/>
    </ScrollView>

    <FrameLayout
        android:id="@+id/fragment_container"
android:name="SimpleFragment"
android:layout_width="0dp"
android:layout_height="wrap_content"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
tools:layout="@layout/fragment_simple"/>
    <Button
        android:id="@+id/open_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/standard_margin"
android:layout_marginStart="@dimen/standard_margin"
android:layout_marginTop="@dimen/standard_margin"
android:text="@string/open"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"/>
</android.support.constraint.ConstraintLayout>
```

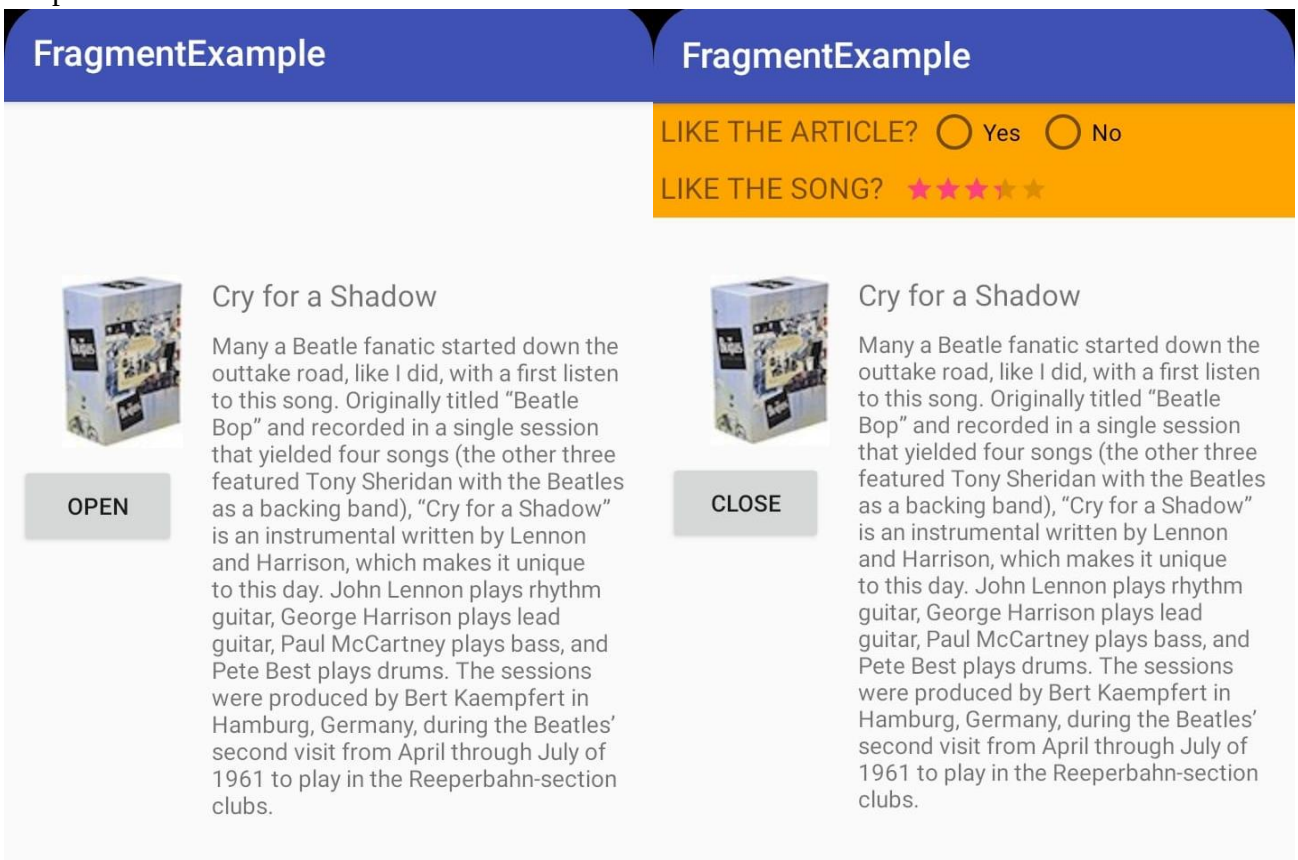
- Fragment_Simple.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/my_fragment_color"
android:orientation="vertical"

tools:context="com.example.android.fragmentexample.SimpleFragment">
    <LinearLayout
        android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">

        <TextView
            android:id="@+id/fragment_header"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="@dimen/small_padding"
android:text="@string/question_article"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"/
>

        <RadioGroup
            android:id="@+id/radio_group"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal">

            <RadioButton
                android:id="@+id/radio_button_yes"
android:layout_width="wrap_content"
android:layout_height="wrap_content"

android:layout_marginRight="@dimen/standard_margin"
android:text="@string/yes"/>
            <RadioButton
                android:id="@+id/radio_button_no"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
                android:layout_marginRight="@dimen/standard_margin"
android:text="@string/no"/>
        </RadioGroup>

    </LinearLayout>

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

        <TextView
            android:id="@+id/fragment_header2"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:padding="@dimen/small_padding"            android:text="LIKE
THE SONG?"
```

```
android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"/
>

        <RatingBar
            android:id="@+id/ratingBar"

style="@android:style/Widget.DeviceDefault.RatingBar.Small"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:numStars="6"
android:isIndicator="false"
            android:padding="@dimen/padding_standard"
android:rating="3.5"              android:stepSize="0.5"/>
    </LinearLayout>
</LinearLayout>
```

B. Output



FragmentExample

ARTICLE: Thanks ◯ Yes ◉ No

LIKE THE SONG? ★★★½☆

CLOSE

Cry for a Shadow

Many a Beatle fanatic started down the outtake road, like I did, with a first listen to this song. Originally titled “Beatle Bop” and recorded in a single session that yielded four songs (the other three featured Tony Sheridan with the Beatles as a backing band), “Cry for a Shadow” is an instrumental written by Lennon and Harrison, which makes it unique to this day. John Lennon plays rhythm guitar, George Harrison plays lead guitar, Paul McCartney plays bass, and Pete Best plays drums. The sessions were produced by Bert Kaempfert in Hamburg, Germany, during the Beatles’ second visit from April through July of 1961 to play in the Reeperbahn-section clubs.

FragmentExample

ARTICLE: Thanks ◯ Yes ◉ No

LIKE THE SONG? ★★★★★

CLOSE

Cry for a Shadow

Many a Beatle fanatic started down the outtake road, like I did, with a first listen to this song. Originally titled “Beatle Bop” and recorded in a single session that yielded four songs (the other three featured Tony Sheridan with the Beatles as a backing band), “Cry for a Shadow” is an instrumental written by Lennon and Harrison, which makes it unique to this day. John Lennon plays rhythm guitar, George Harrison plays lead guitar, Paul McCartney plays bass, and Pete Best plays drums. The sessions were produced by Bert Kaempfert in Hamburg, Germany, during the Beatles’ second visit from April through July of 1961 to play in the Reeperbahn-section clubs.

FragmentExample

ARTICLE: Like (•) Yes ( ) No

LIKE THE SONG? ★★★½☆

CLOSE

Cry for a Shadow

Many a Beatle fanatic started down the outtake road, like I did, with a first listen to this song. Originally titled “Beatle Bop” and recorded in a single session that yielded four songs (the other three featured Tony Sheridan with the Beatles as a backing band), “Cry for a Shadow” is an instrumental written by Lennon and Harrison, which makes it unique to this day. John Lennon plays rhythm guitar, George Harrison plays lead guitar, Paul McCartney plays bass, and Pete Best plays drums. The sessions were produced by Bert Kaempfert in Hamburg, Germany, during the Beatles’ second visit from April through July of 1961 to play in the Reeperbahn-section clubs.

FragmentExample

ARTICLE: Thanks ( ) Yes (•) No

LIKE THE SONG? ★★★★★

CLOSE

Cry for a Shadow

Many a Beatle fanatic started down the outtake road, like I did, with a first listen to this song. Originally titled “Beatle Bop” and recorded in a single session that yielded four songs (the other three featured Tony Sheridan with the Beatles as a backing band), “Cry for a Shadow” is an instrumental written by Lennon and Harrison, which makes it unique to this day. John Lennon plays rhythm guitar, George Harrison plays lead guitar, Paul McCartney plays bass, and Pete Best plays drums. The sessions were produced by Bert Kaempfert in Hamburg, Germany, during the Beatles’ second visit from April through July of 1961 to play in the Reeperbahn-section clubs.

Implementasi RecycleView

A. Praktikum

- MainActivity.java

```
package com.example.android.recyclerview;

import android.os.Bundle;
import
android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.LinearLayoutManager; import
android.support.v7.widget.RecyclerView; import
android.view.View;

import java.util.LinkedList;

/**
*     Implements a basic RecyclerView that displays a list of
generated words.
*     - Clicking an item marks it as clicked.
*     - Clicking the fab button adds a new word to the list.
 */
public class MainActivity extends AppCompatActivity {

    private RecyclerView mRecyclerView;
private WordListAdapter mAdapter;
    private final LinkedList<String> mWordList = new LinkedList<>();
    @Override
    public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

        // Put initial data into the word list.
        for (int i = 0; i < 20; i++) {
mWordList.addLast("Word " + i);          }

        // Create recycler view.
mRecyclerView = (RecyclerView)
findViewById(R.id.recyclerview);
        // Create an adapter and supply the data to be displayed.
        mAdapter = new WordListAdapter(this, mWordList);
// Connect the adapter 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) {
int wordListSize = mWordList.size();
// Add a new word to the wordList.
                mWordList.addLast("+ Word " + wordListSize);
// Notify the adapter, that the data has changed.

mRecyclerView.getAdapter().notifyItemInserted(wordListSize);
                // Scroll to the bottom.
                mRecyclerView.smoothScrollToPosition(wordListSize);
            }
        });
    }
}
```

- WordListAdapter.java

```
package com.example.android.recyclerview;

import android.content.Context;
import
android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater; import
android.view.View; import
android.view.ViewGroup; import
android.widget.TextView;

import java.util.LinkedList;

/**
* Shows how to implement 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> {

    private final LinkedList<String> mWordList;
private LayoutInflater mInflater;
     class WordViewHolder extends
RecyclerView.ViewHolder             implements
View.OnClickListener {

        public final TextView
wordItemView;         final
WordListAdapter mAdapter;
        /**
         * Creates a new custom view holder to hold the view
to display in the RecyclerView.          *
```

```
         * @param itemView The view in which to display the data.
* @param adapter The adapter that manages the the data and views
for the RecyclerView.
         */
        public WordViewHolder(View itemView, WordListAdapter adapter)
{
            super(itemView);
wordItemView = (TextView)
itemView.findViewById(R.id.word);
this.mAdapter = adapter;
            itemView.setOnClickListener(this);
        }

        @Override
        public void onClick(View view) {
            // Get the position of the item that was clicked
int mPosition = getLayoutPosition();
            // Use that to access the affected item in mWordList.
String element = mWordList.get(mPosition);
            // Change the word in the mWordList.
mWordList.set(mPosition, "Clicked! " + element);
            // Notify the adapter that the data has changed so it can
            // update the RecyclerView to display the data.
mAdapter.notifyDataSetChanged();
        }
    }
    public    WordListAdapter(Context    context,    LinkedList<String>
wordList) {
        mInflater = LayoutInflater.from(context);
this.mWordList = wordList;
    }

    /**
*     Inflates an item view and returns a new view holder that
contains it.
*     Called when the RecyclerView needs a new view holder to
represent an item.
     *
*     @param parent The view group that holds the item views.
*     @param viewType Used to distinguish views, if more than one
*     type of item view is used.
*     @return a view holder.
     */
@Override
    public  WordViewHolder  onCreateViewHolder(ViewGroup  parent,  int
viewType) {
        // Inflate an item view.
        View  itemView  =  mInflater.inflate(R.layout.wordlist_item,
parent, false);
        return new WordViewHolder(itemView, this);
    }

    /**
*     Sets the contents of an item at a given position in the
RecyclerView.
```

```
     *
*     @param holder The view holder for that position in the
RecyclerView.
*     @param position The position of the item in the RecycerView.
     */
```

```
    @Override
    public void onBindViewHolder(WordViewHolder holder, int position)
{
        // Retrieve the data for that position.
        String current = mWordList.get(position);
        // Add the data to the view holder.
holder.wordItemView.setText(current);
    }

    /**
* Returns the size of the container that holds the data.
     *
* @return Size of the list of data.
     */
@Override
    public int getItemCount() {
return mWordList.size();
    }
}
```

- activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

    <android.support.v7.widget.RecyclerView
android:id="@+id/recyclerview"
android:layout_width="match_parent"
android:layout_height="match_parent">
</android.support.v7.widget.RecyclerView>
    <android.support.design.widget.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" />

</android.support.design.widget.CoordinatorLayout>
```

- content_main.xml

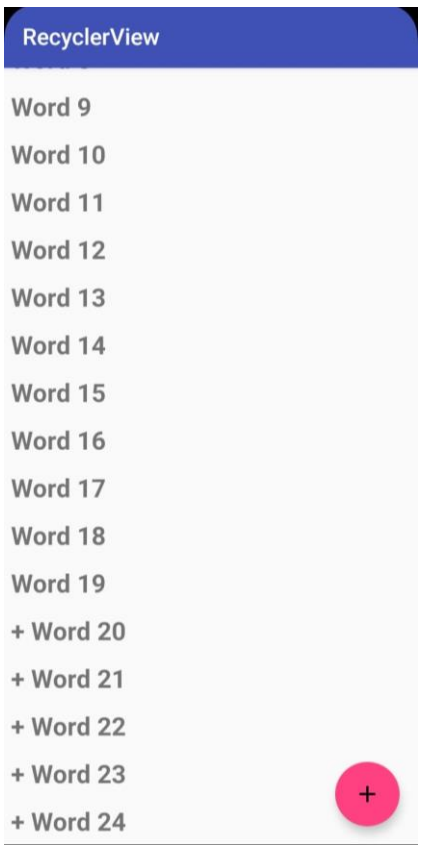
```
<?xml version="1.0" encoding="utf-8"?>
```

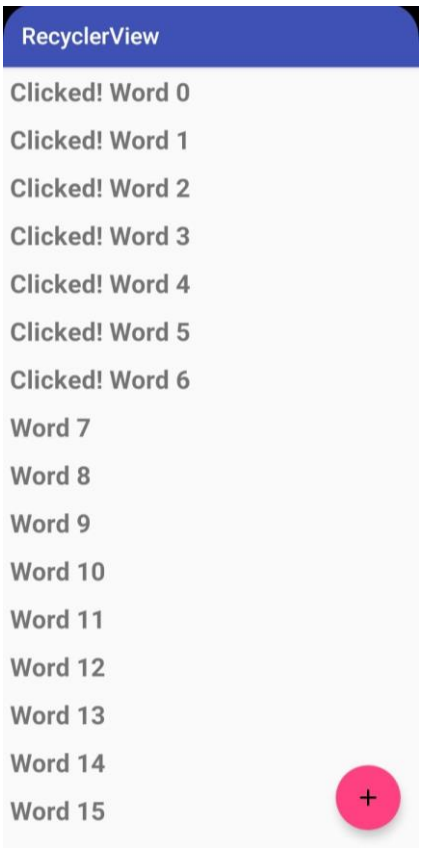
```
<android.support.v7.widget.RecyclerView
android:id="@+id/recyclerview"     android:layout_width="match_parent"
android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent">

    </LinearLayout>

</android.support.v7.widget.RecyclerView>
```

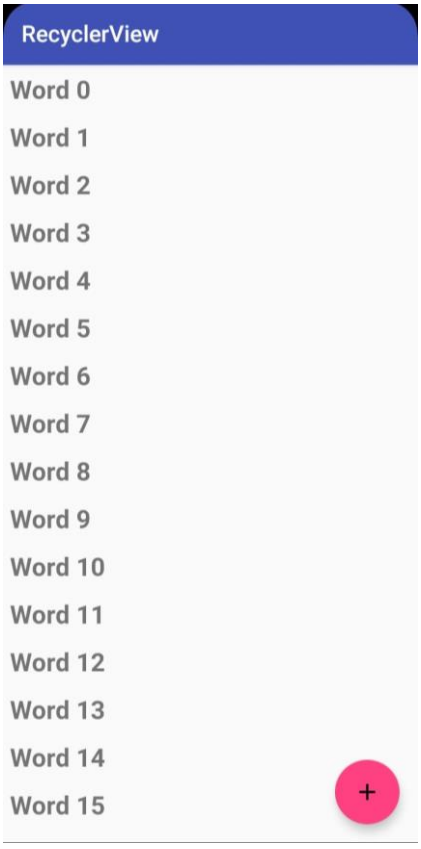
- wordlist_item.xml

```
<?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"
android:layout_height="wrap_content"
android:textSize="24sp"
android:textStyle="bold"
style="@style/word_title" />

</LinearLayout>
```

B. Output


RecyclerView
Word 9
Word 10
Word 11
Word 12
Word 13
Word 14
Word 15
Word 16
Word 17
Word 18
Word 19
+ Word 20
+ Word 21
+ Word 22
+ Word 23
+ Word 24


RecyclerView
Clicked! Word 0
Clicked! Word 1
Clicked! Word 2
Clicked! Word 3
Clicked! Word 4
Clicked! Word 5
Clicked! Word 6
Word 7
Word 8
Word 9
Word 10
Word 11
Word 12
Word 13
Word 14
Word 15


RecyclerView
Word 0
Word 1
Word 2
Word 3
Word 4
Word 5
Word 6
Word 7
Word 8
Word 9
Word 10
Word 11
Word 12
Word 13
Word 14
Word 15