# Materi 6: RecyclerView

## Pokok Bahasan

Melakukan implementasi RecyclerView pada Android

## Tujuan

Di Akhir pertemuan ini, diharapkan mahasiswa dapat memahami dan juga mengimplementasikan penggunaan RecyclerView pada saat membuat aplikasi Android.

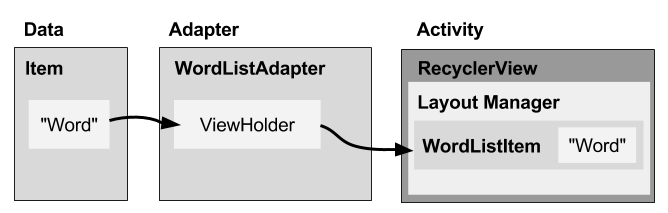
## Penjelasan Singkat

Pada pertemuan kali ini anda akan memahami dan juga mengimplementasikan cara menggunakan RecyclerView. RecyclerView merupakan komponen yang cukup penting dikarenakan sering digunakan untuk menampilkan list pada sebuah aplikasi. [RecyclerView](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html) merupakan scrollable container yang dipergunakan untuk menampilkan data dalam jumlah besar. Dengan cara ini salah satunya merupakan cara efisiensi untuk membatasi elemen yang ditampilkan pada View, kemudian ketika ada perubahan data dalam elemen RecyclerView akan dengan mudah dilakukan penambahan dan juga update.

Adapun komponen yang dipergunakan dalam RecyclerView diantaranya sebagai berikut:

1. **Data**
2. **RecyclerView** merupakan komponen scrolling list untuk setiap item
3. **Layout untuk satu item data** yang ditulis dalam XML file
4. **Layout Manager** yang dipergunakan untuk menghandle untuk komponen UI dalam satu View ([Recyclerview.LayoutManager](https://developer.android.com/.../RecyclerView.LayoutManager.html))
5. **Adapter** yang dipergunakan untuk menghandle koneksi data ke RecyclerView ([RecyclerView.Adapter](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.Adapter.html))
6. **ViewHolder** yang menyediakan informasi untuk menampilkan satu item ([RecyclerView.ViewHolder](https://developer.android.com/.../RecyclerView.ViewHolder.html))

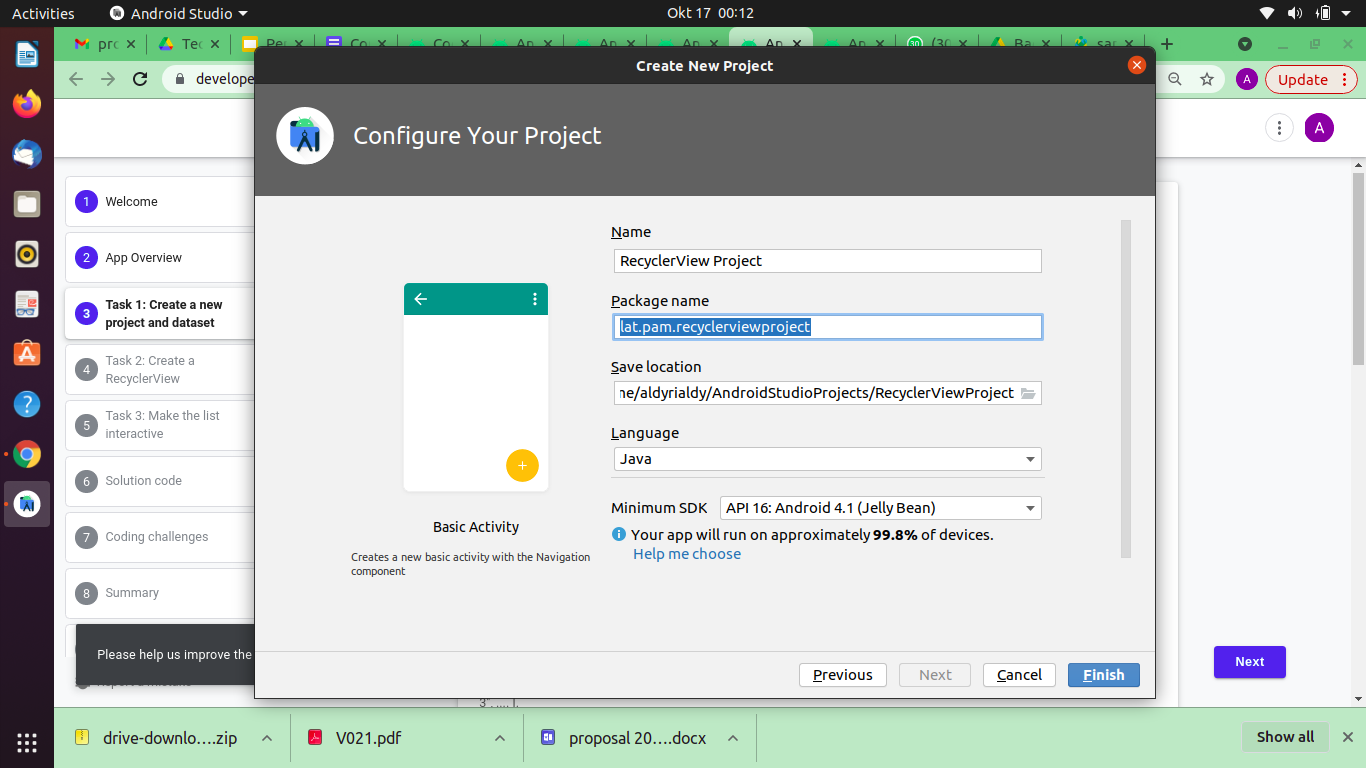
Berikut ini merupakan gambaran dari cara kerja pada RecyclerView yang ditampilkan pada gambar dibawah ini.



## Modul 1: Melakukan implementasi RecyclerView pada aplikasi Android

Pada modul pertama ini anda akan membuat satu buah tampilan yang berisi daftar yang ditampilkan dalam bentuk RecyclerView. Adapun langkah yang perlu dilakukan diantaranya sebagai berikut:

1. Buat satu buah project baru dengan nama RecyclerView Project, kemudian pilih Basic Activity. Kemudian sesuaikan project tersebut dengan nama package lat.pam.recyclerviewproject. Gunakan bahasa pemrograman Java dengan minimum SDK API 16: Android 4.1 (Jelly Bean) seperti dibawah ini.



1. Setelah project tersebut berhasil dibuat, anda dapat menambahkan elemen RecyclerView pada fragment\_first.xml. Gantilah tulisan Hello World (tampilan default) menjadi kode XML berikut.

<androidx.recyclerview.widget.RecyclerView

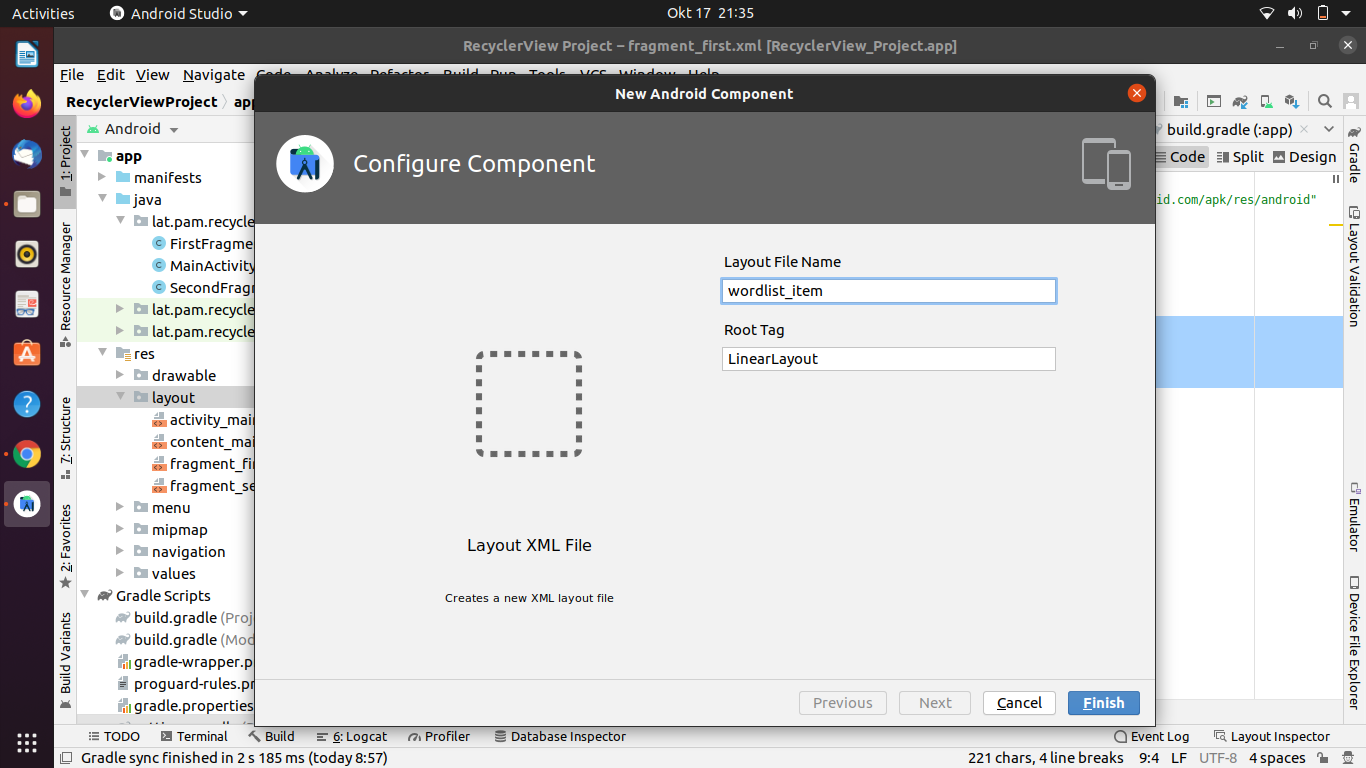
android:id="@+id/recyclerview"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</androidx.recyclerview.widget.RecyclerView>

1. Kemudian buatlah satu file baru dengan nama wordlist\_item.xml pada app > res > layout dengan menggunakan LinearLayout, seperti pada tampilan berikut.



1. Tambahkan atribut TextView pada wordlist\_item.xml sehingga kode XML yang dituliskan menjadi berikut.

*<?*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"

style="@style/word\_title" />

</LinearLayout>

1. Selain itu, tambahkan style.xml seperti dibawah ini, sehingga wordlist\_item dapat mendapatkan style yang diambil dari file tersebut.

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

<resources>

<style name="word\_title">

<item name="android:layout\_width">match\_parent</item>

<item name="android:layout\_height">wrap\_content</item>

<item name="android:textSize">24sp</item>

<item name="android:textStyle">bold</item>

</style>

</resources>

1. Pindahkan desain fab pada first\_fragment, kemudian ubah layout menjadi menggunakan CoordinatorLayout seperti berikut ini.

*<?*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=".FirstFragment">

<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="@dimen/fab\_margin"

app:srcCompat="@android:drawable/ic\_dialog\_email"

tools:ignore="MissingConstraints"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

1. Pada tahap awal ini anda perlu menginisialisasi data LinkedList yang anda taruh pada FirstFragment pada method onViewCreated()

final LinkedList<String> mWordList = new LinkedList<>();

1. Selanjutnya data pada LinkedList tersebut akan ditambahkan word pada linkedlist yang anda taruh kodenya pada method onViewCreated()

*// Put initial data into the word list.*

for (int i = 0; i < 20; i++) {

mWordList.addLast("Word " + i);

}

1. Selanjutnya, untuk mengkoneksikan antara data dan juga View dibutuhkan Adapter. Sehingga anda perlu menambahkan Java class adapter dengan cara pilih New > Java Class pada package android project tersebut. Kemudian beri nama WordListAdapter.
2. Kemudian class Adapter tersebut anda tambahkan extends yang mengarah ke RecyclerView.Adapter<WordListAdapter.WordViewHolder>, sehingga kelas yang dibuat akan seperti berikut.

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

}

1. Setelah itu buatlah satu buah inner class dengan nama WordViewHolder. Inner class ini mempunyai parent/ extends kepada RecyclerView.ViewHolder. Kemudian tambahkan kode berikut.

public final TextView wordItemView;

final WordListAdapter mAdapter;

1. Setelah itu buatlah konstruktor untuk WordViewHolder dengan diisi parameter itemView dan WordListAdapter seperti yang ditunjukkan dibawah ini. Sehingga kode yang ditampilkan seperti berikut.

public WordViewHolder(@NonNull View itemView, WordListAdapter adapter) {

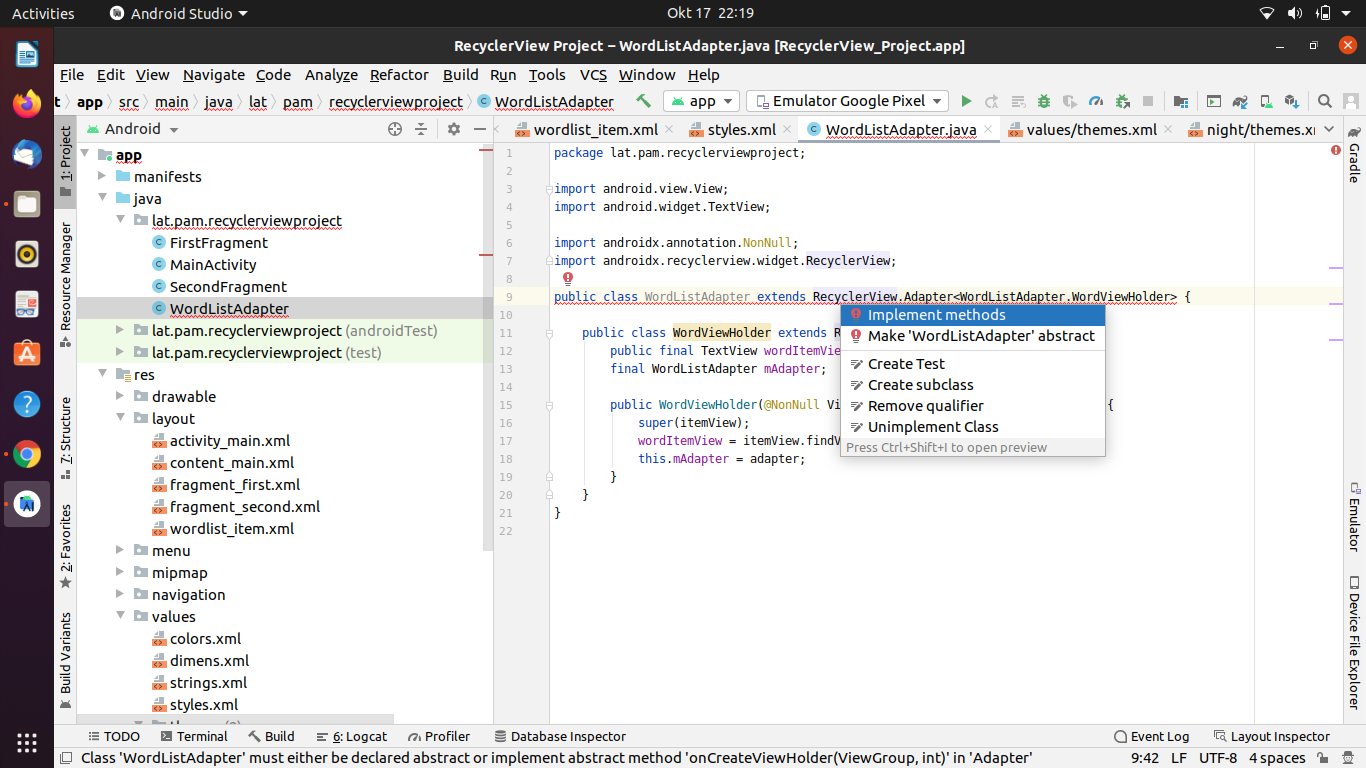
super(itemView);

wordItemView = itemView.findViewById(R.id.*word*);

this.mAdapter = adapter;

}

1. Kemudian lakukan implement methods dengan menambahkan method diantaranya yaitu onCreateViewHolder, onBindViewHolder dan getItemCount, cara tersebut anda dapat lakukan dengan cara klik Alt+Enter pada kode program yang ditandai merah (pada class WordListAdapter).



1. Selanjutnya, anda perlu melakukan inisialisasi awal pada adapter, sehingga data tersebut dapat ditampung pada Adapter, pertama anda perlu melakukan inisialisasi mWordList.

private final LinkedList<String> mWordList;

1. Kemudian tambahkan method getItemCount untuk mendapatkan jumlah dari list yang dikeluarkan dari mWordList.

@Override

public int getItemCount() {

return mWordList.size();

}

1. Selanjutnya inisialisasi LayoutInflater untuk memanggil item pada list sehingga dapat dimunculkan pada View,kode program ini disimpan setelah baris private final LinkedList<String> mWordList;

private LayoutInflater mInflater;

1. Kemudian tambahkan LayoutInflater tersebut pada WordListAdapter seperti berikut.

public WordListAdapter(FirstFragment context, LinkedList<String> wordList) {

mInflater = LayoutInflater.*from*(context.getContext());

this.mWordList = wordList;

}

1. Tambahkan kode program berikut pada onCreateViewHolder()

*// Inflate an item view.*

View mItemView = mInflater.inflate(

R.layout.*wordlist\_item*, parent, false);

return new WordViewHolder(mItemView, this);

1. Tambahkan kode program berikut pada onBindViewHolder()

String mCurrent = mWordList.get(position);

holder.wordItemView.setText(mCurrent);

1. Pada tahapan selanjutnya anda perlu menambahkan inisialisasi RecyclerView pada class FirstFragment dengan menambahkan kode berikut.

private RecyclerView mRecyclerView;

private WordListAdapter mAdapter;

1. Kemudian tambahkan kode dibawah ini untuk mengkoneksikan adapter dengan data dengan data, kode tersebut anda simpan pada onViewCreated()

*// Get a handle to the RecyclerView.*

mRecyclerView = view.findViewById(R.id.*recyclerview*);

*// Create an adapter and supply the data to be displayed.*

mAdapter = new WordListAdapter(this, mWordList);

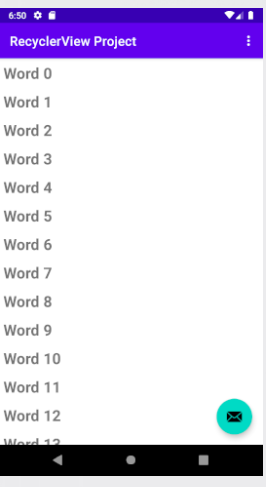
*// Connect the adapter with the RecyclerView.*

mRecyclerView.setAdapter(mAdapter);

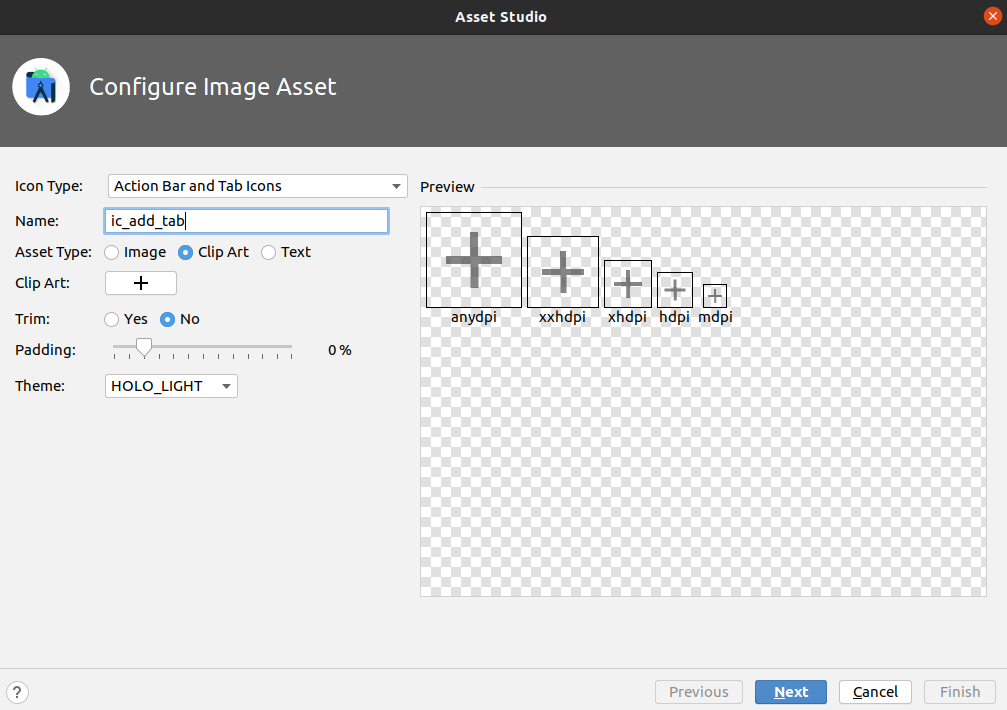
*// Give the RecyclerView a default layout manager.*

mRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));

1. Jalankan aplikasi sehingga tampilannya akan seperti berikut.



1. Pada tahapan selanjutnya anda perlu mengganti icon pada FAB dengan icon yang sesuai dengan melakukan klik kanan pada Project kemudian pilih New > Image Asset dan pilih Action Bar and Tab Items, pilih icon yang merepresentasikan “tambah” seperti pada tampilan berikut. Kemudian klik Next dan Finish.



1. Ganti icon pada fab dengan baris berikut.

app:srcCompat="@drawable/ic\_add\_tab"

1. Tambahkan kode program pada konstruktor WordViewHolder()

itemView.setOnClickListener(this);

1. Tambahkan kode program berikut dengan tujuan agar memberikan aksi ketika item pada list tersebut dilakukan klik.

@Override

public void onClick(View v) {

*// 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();

}

1. Kemudian anda dapat menambahkan aksi pada FAB ketika tombol tersebut diklik, anda dapat menambahkan kode berikut.

FloatingActionButton fab = view.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);

}

});

1. Jalankan aplikasi tersebut sehingga dapat memberikan aksi seperti berikut.
   1. Pada saat item pada list diklik akan muncul Clicked! Word…
   2. Sedangkan pada saat tombol tambah (FAB) diklik akan muncul + Word selanjutnya pada akhir dari list

