FILKOM
FAKULTAS ILMU KOMPUTER

PEMROGRAMAN APLIKASI PERANGKAT BERGERAK (MOBILE)

RecyclerView, Collection, Adapter

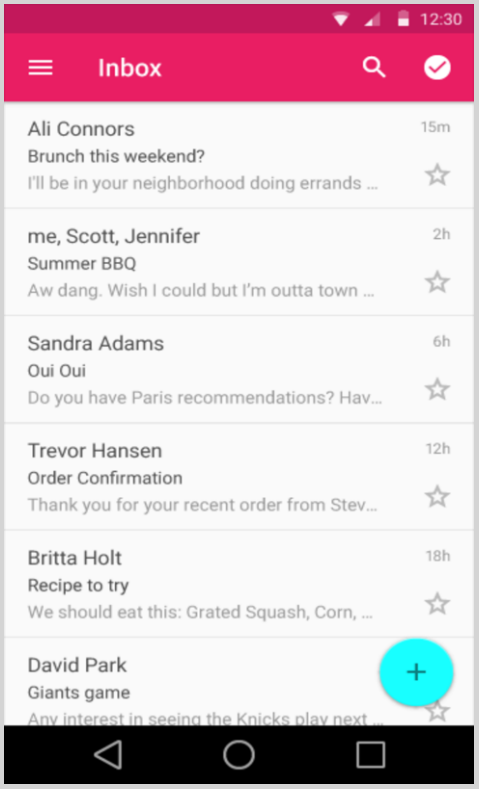
K Candra Brata
andra.course@gmail.com

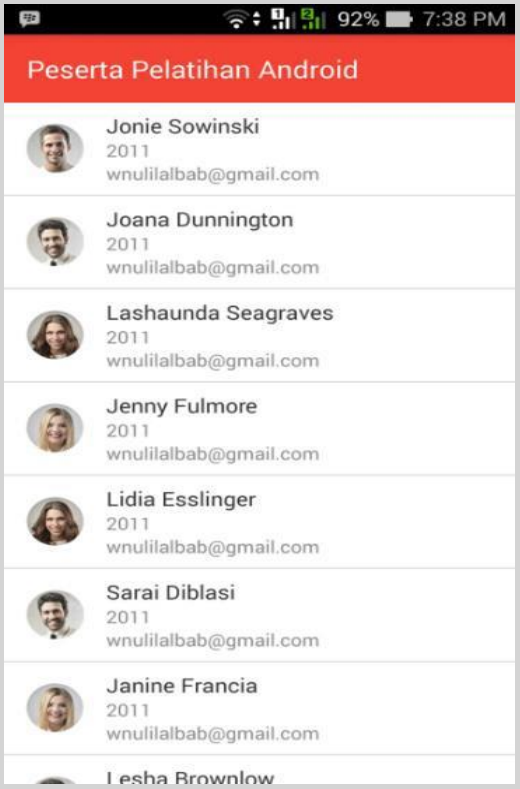
RecyclerView

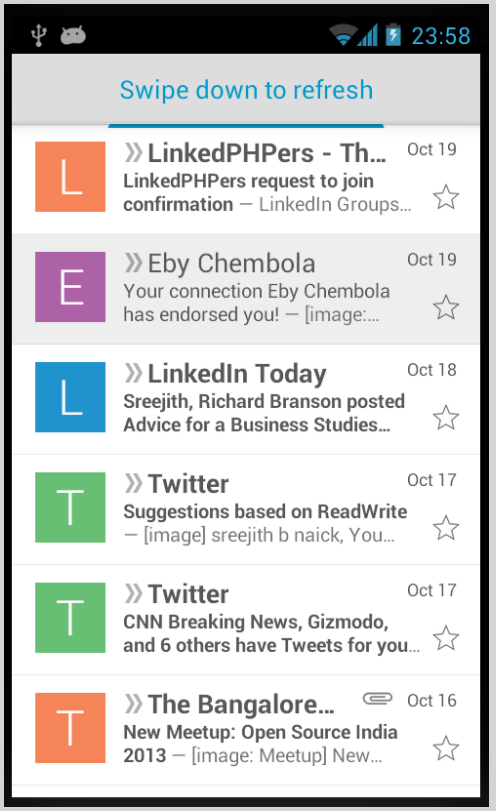
https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html

ListView

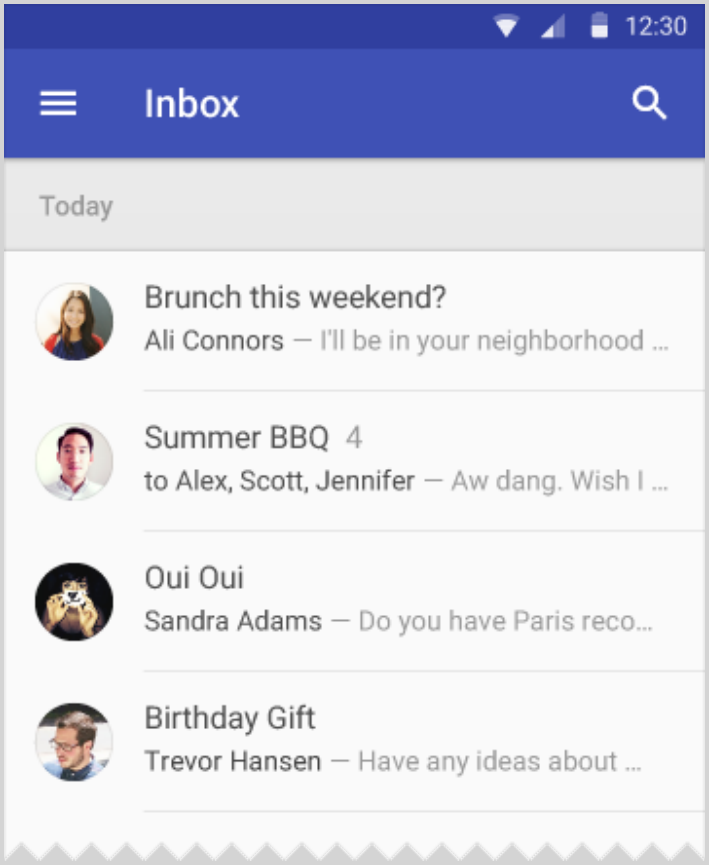
- A specific view that shows items in a vertically scrolling list.
- An ordered collection of selectable choices.
- Uses **adapter** to populate listView items with data.

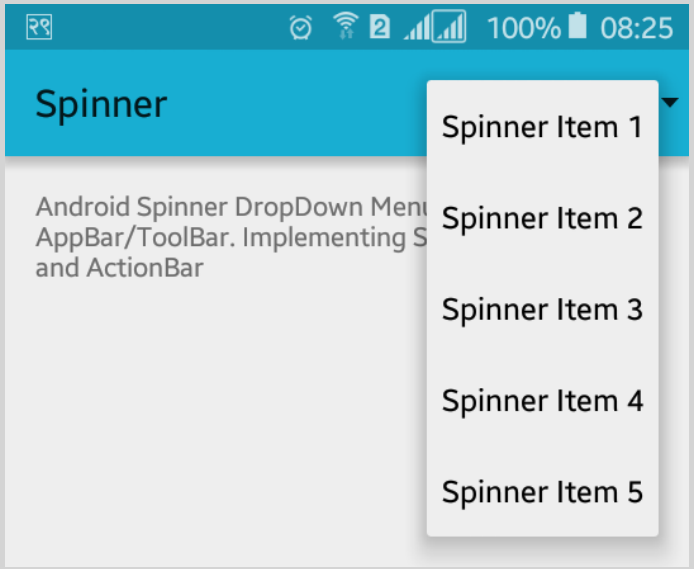


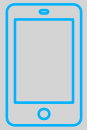




List-based View and DataSet Problem







ListView

- 2 kind of ListView :
  - **Static List.**
  - **Dynamic List.**

- key attributes in XML:

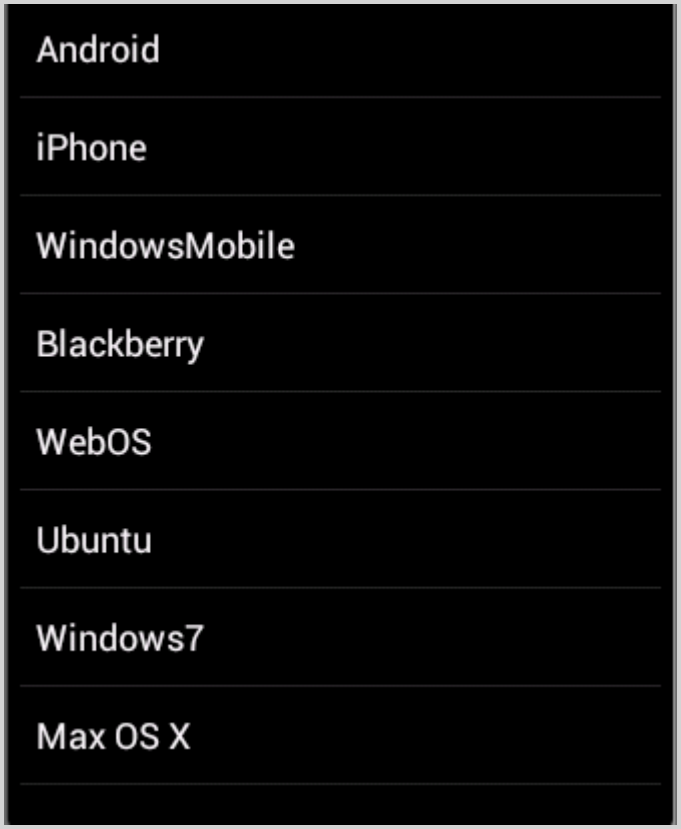
`android:clickable="`***`bool`***`"`	set to false to disable the list
`android:id="@+id/`***`theID`***`"`	unique ID for use in Java code
`android:entries="@array/`***`array`***`"`	set of options to appear in the list (must match an array in `strings.xml`)

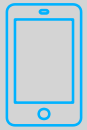
Static ListView

static list: Content is fixed and known before the app runs.
– Declare the list elements in the **strings.xml resource file.**

```
<!-- res/values/strings.xml -->
<resources>
<string-array name="oses">
<item>Android</item>
<item>iPhone</item>
...
<item>Max OS X</item>
</string-array>
</resources>
```

```
<!-- res/layout/activity_main.xml -->
<ListView ... android:id="@+id/mylist"
android:entries="@array/oses" />
```





Dynamic ListView

Dynamic list: Content is read or generated as the program runs.

- **Content comes from a data file, or from the internet (WS), etc.**
- **Must be set in the Java code.**

- Uses **adapter** to populate listView items with data.

App Data

- Many times application data comes in the form of lists of objects rather than lists of simple Strings
  - E.g. contacts data which consists of
    - Name, Phone Number, E-mail, etc.
- This is the type of data that is best served by **custom adapter** classes since you will want to present all of these detailed information to the user somehow

RecyclerView

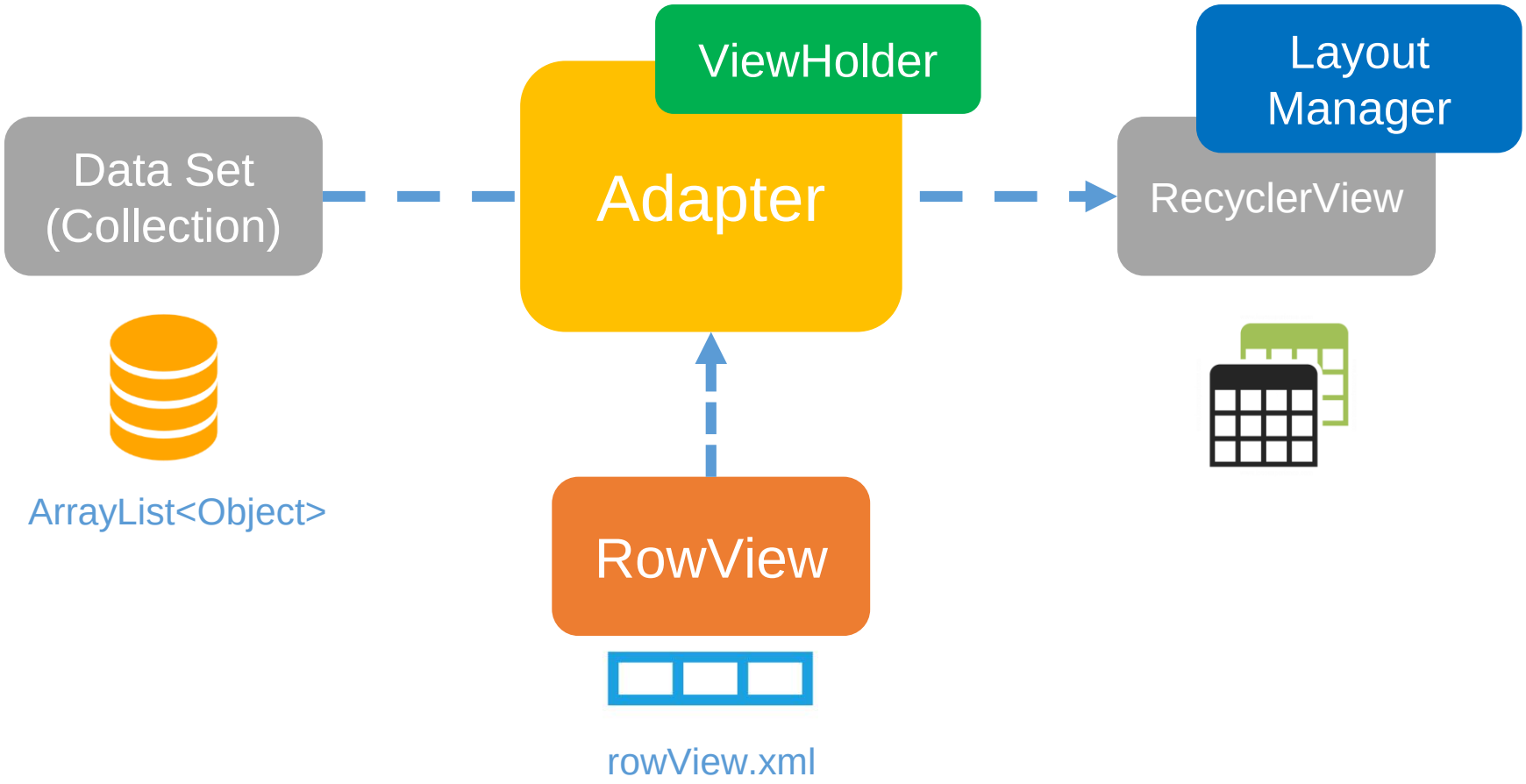
Secara definisi **RecyclerView** adalah sebuah komponen tampilan *(**widget**)* yang lebih canggih ketimbang pendahulunya **ListView** dan bersifat lebih fleksibel.

RecyclerView memiliki kemampuan untuk menampilkan data secara efisien dalam jumlah yang besar. Terlebih jika anda memiliki koleksi data yang tiap elemennya mampu berubah-ubah sewaktu dijalankan *(runtime)* karena interaksi pengguna atau karena adanya pengaruh dari jaringan internet.

RecyclerView

- Scrollable container for large data sets
- **Efficient**
  - uses and reuses limited number of views
  - Updates changing data fast

Dynamic List using Adapter



All Components overview

- **Data**
- **RecyclerView** scrolling list for list items
- **RowView** Layout for one item of data (XML file)
- **Layout manager** handles the organization of UI components in a view
  Recyclerview.LayoutManager

- **Adapter** connects data to the RecyclerView
  RecyclerView.Adapter

- **View holder** has view information for displaying one item
  RecyclerView.ViewHolder

Layout Manager

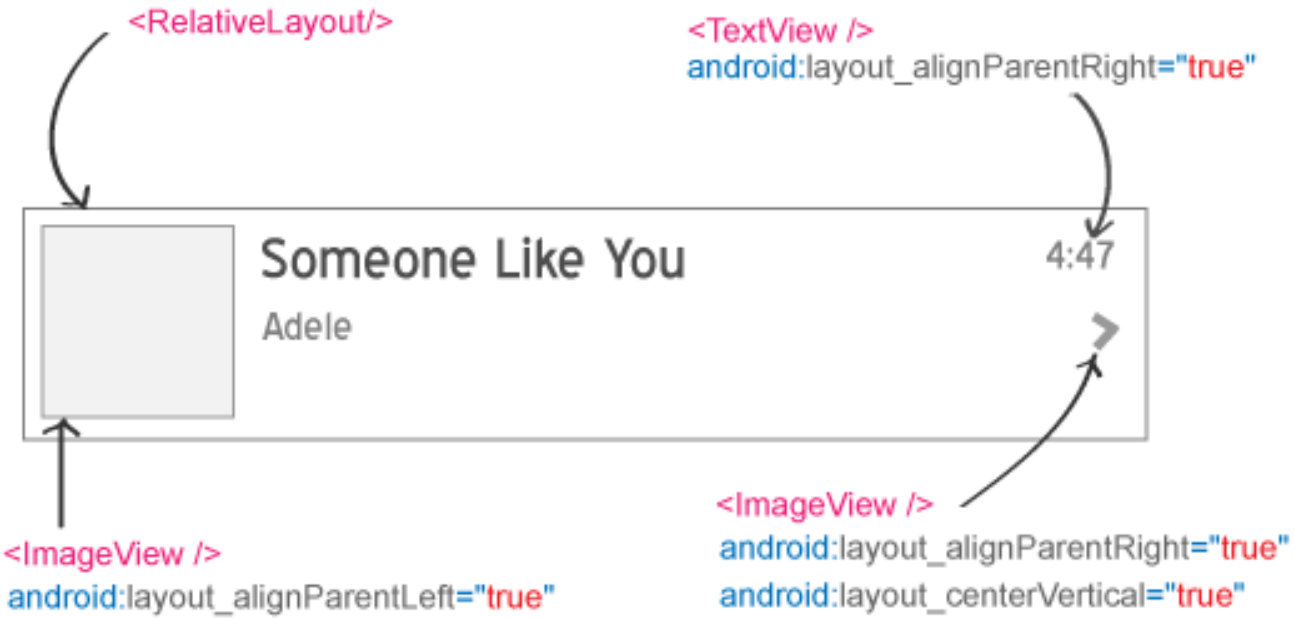
- All view groups have layout managers
- Positions item views inside a RecyclerView.
- Reuses item views that are no longer visible to the user
- Built-in layout managers:
  - LinearLayoutManager, GridLayoutManager, and StaggeredGridLayoutManager
- For RecyclerView, extend **RecyclerView.LayoutManager**

Adapter

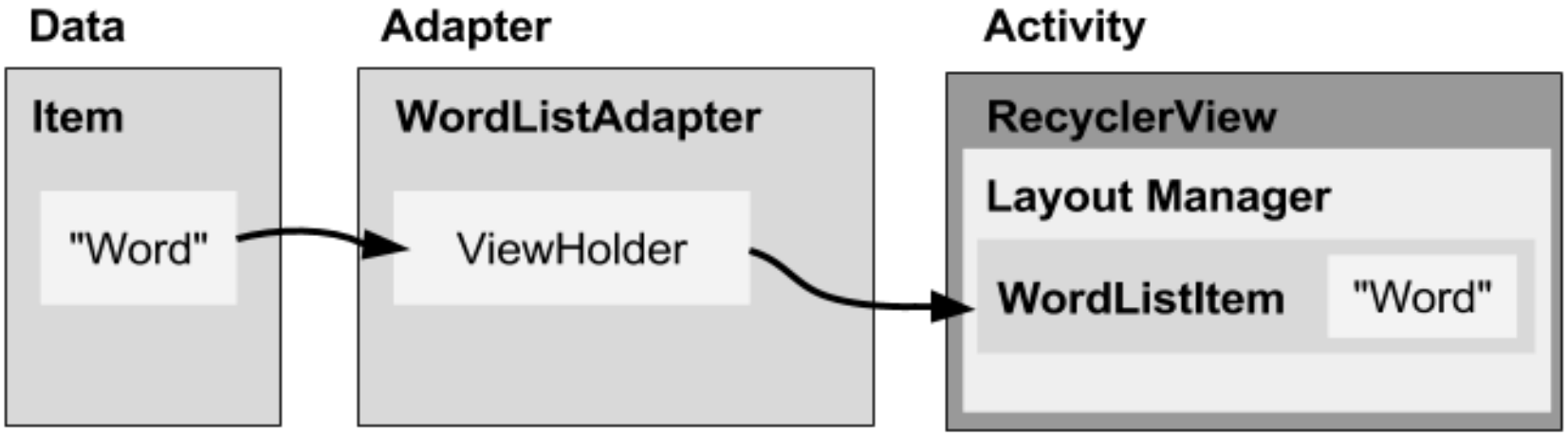
- Intermediary object between data and view
- The Adapter provides access to the data items (model/data set).
- Helps incompatible interfaces work together, for example, takes data from a database Cursor and puts them as strings into a view.
- Manages creating, updating, adding, deleting item views as the underlying data changes.
- An Adapter can be implemented by extending the **BaseAdapter** abstract class.
- Several subtypes of BaseAdapter classes that are built-in to Android
  ListAdapter, ArrayAdapter, SpinnerAdapter

View Holder

- An object that is used by the adapter to prepare **one view** with **data for one list item**.
- Layout specified in an XML resource file.
- Can have clickable elements
- Is placed by the layout manager



How components fit together overview



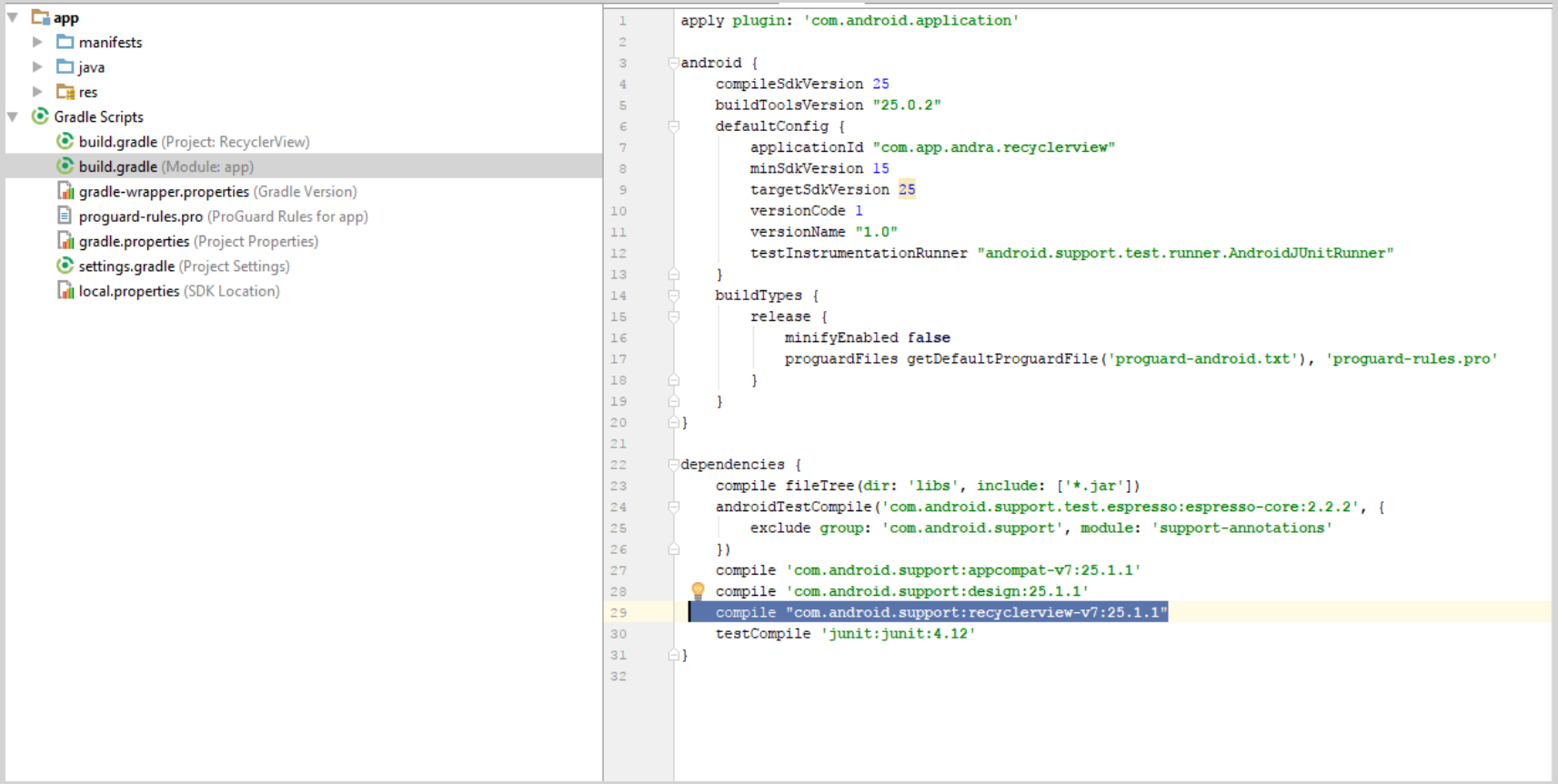
RecyclerView Implementation

Langkah-langkah mengimplementasikan **RecyclerView** sebagai berikut :

1. Tambahkan **dependencies** komponen **RecyclerView** pada file **build.gradle** (module: app) level modul.
2. Tambahkan obyek **RecyclerView** di file layout xml dari **Activity** / **Fragment.**
3. Definisikan collection, bisa array bisa model kelas (POJO) yang akan digunakan sebagai data source.
4. Buat file layout (.xml) untuk tiap baris item di **RecyclerView**.
5. Buat sebuah kelas **adapter** yang inherit ke **RecyclerView.Adapter** dan **ViewHolder** untuk menampilkan tiap elemen data.
6. Definisikan obyek **RecyclerView** berikut dengan bentuk yang diinginkan (bisa dalam bentuk list, grid, atau staggered) dan selanjutnya pasang obyek **adapter** (*binding*) agar bisa menampilkan koleksi data ke dalam **RecyclerView**.

Add dependency to app/build.gradle

```
dependencies {
   ...
   compile 'com.android.support:recyclerview-v7:25+'
   ...
}
```



Add RecyclerView to main XML Layout

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

    <TextView
        android:id="@+id/selection"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#F44333"
        android:gravity="center_horizontal"
        android:text="List Sitem Operasi"
        android:textSize="20dp"/>


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


</LinearLayout>
```

Create Layout for each list item (Row View) => **rowview.XML**

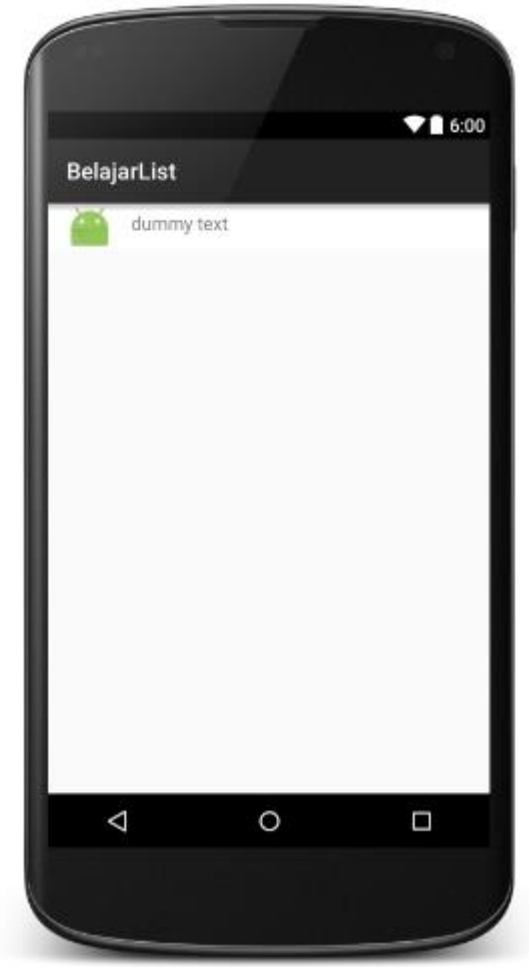
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/logo"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_gravity="center"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:src="@mipmap/ic_launcher" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center_vertical|left"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/namaos"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:paddingTop="8dp"
            android:text="dummy text"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:textSize="16dp" />

    </LinearLayout>
</LinearLayout>
```



Create Class Representing an Item from Collection => **Sisop.Java**

```
public class Sisop {

    public String nama;

    public Sisop (String nama) {
        this.nama = nama;
    }
}
```

Create the Custom Adapter

```
public class CustomAdapter extends RecyclerView.Adapter
  <CustomAdapter.customViewHolder> {


  LayoutInflater mInflater;
  ArrayList<Sisop> sisop;

  public CustomAdapter(Context context,
    ArrayList<Sisop> sisop) {
      this.mInflater = LayoutInflater.from(context);
      this.sisop = sisop;
  }
}
```

Create the View Holder Class **inside** the Adapter Class

```
class CustomViewHolder
    extends RecyclerView.ViewHolder {}
```

If you want to handle mouse clicks:

```
class CustomViewHolder
    extends RecyclerView.ViewHolder
    implements View.OnClickListener {}
```

View Holder Constructor

```
public CustomViewHolder(View itemView, CustomAdapter adapter) {
  super(itemView);

// Get the layout

  namaItemView = (TextView) itemView.findViewById(R.id.namaos);

// Associate with this adapter
  this.mAdapter = adapter;

  // Add click listener, if desired

  itemView.setOnClickListener(this);
}

// Implement onClick() if desired

onClick(){
}
```

3 Adapter Required Methods

- `onCreateViewHolder()`
- `onBindViewHolder()`
- `getItemCount()`

Let's take a look!

onCreateViewHolder()

```
@Override
public CustomViewHolder onCreateViewHolder(
  ViewGroup parent, int viewType) {

// Create view from layout
  View itemView = mInflater.inflate(
    R.layout.rowview, parent, false);
  return new CustomViewHolder(itemView, this);
}
```

onBindViewHolder()

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

  // Retrieve the data for that position
  Sisop current = sisop.get(position);
  // Add the data to the view
  holder.namaItemView.setText(current.nama);


// add the Listener to the view of that position if desired
  holder.namaItemView.setOnClickListener ();

}
```

getItemCount()

```
@Override
public int getItemCount() {

    // Return the number of
    // data items to display

    return sisop.size();
}
```

Create the RecyclerView in activity's onCreate()

```
ArrayList<Sisop> listOs = new ArrayList<Sisop>();

// populasikan nama Sisop di sini

 listOs.add(new Sisop("Android”));
 listOs.add(new Sisop(“iPhone”));
 ...


mRecyclerView =
  (RecyclerView) findViewById(R.id.recyclerview);
mAdapter = new CustomAdapter(this, listOs);
mRecyclerView.setAdapter(mAdapter);
mRecyclerView.setLayoutManager(new
  LinearLayoutManager(this));
```

Modifying Collection

Modifying Collection

Modifying collection item in an adapter
E.g., Inserting, Deleting, Updating item in the collection
Perform modification directly to the collection or to the collection item.
To reflect the changes in RecyclerView, call:

```
adapter.notifyDataSetChanged()
```

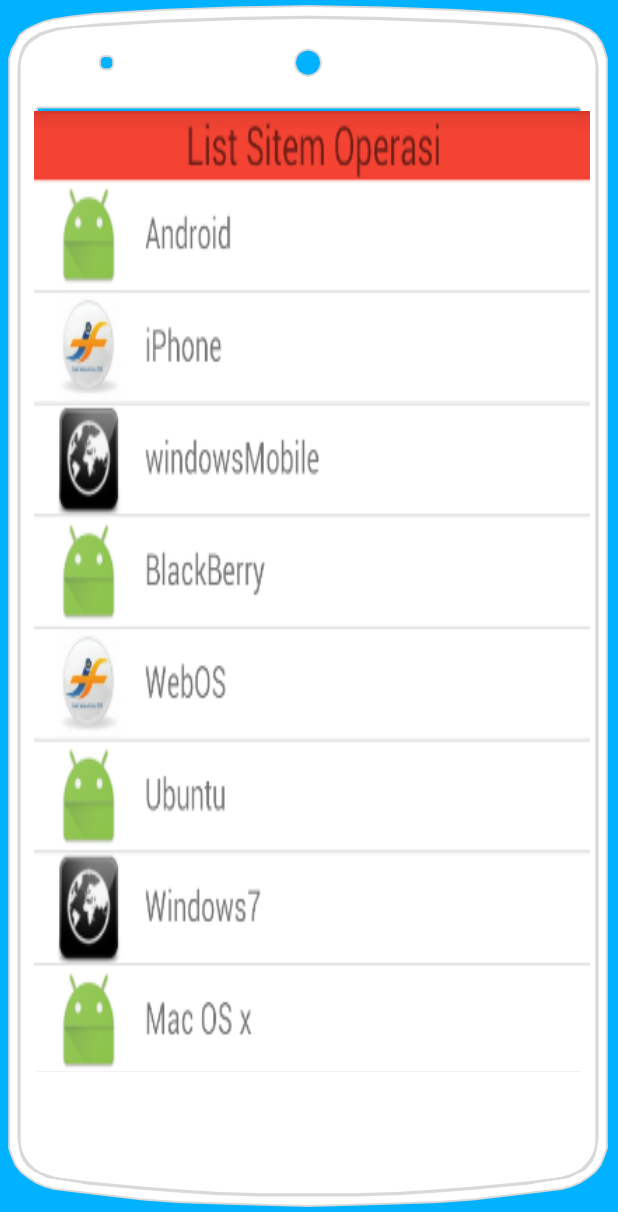
or

```
notifyDataSetChanged()
```

from inside the adapter itself

Tugas !!

- Realisasikan Potongan kode yang ada di Slide !
- Modifikasi Kelas Adapter agar dapat menerima masukan image saat instansiasi object !
- Tambahkan Listener pada gambar agar ketika setiap gambar item diclick, memunculkan Toast nama OS sesuai item yang diclick dan menghilangkan item yang diclick.



Thanks

We are moving..

http://j.gs/18164083/papb-tif