

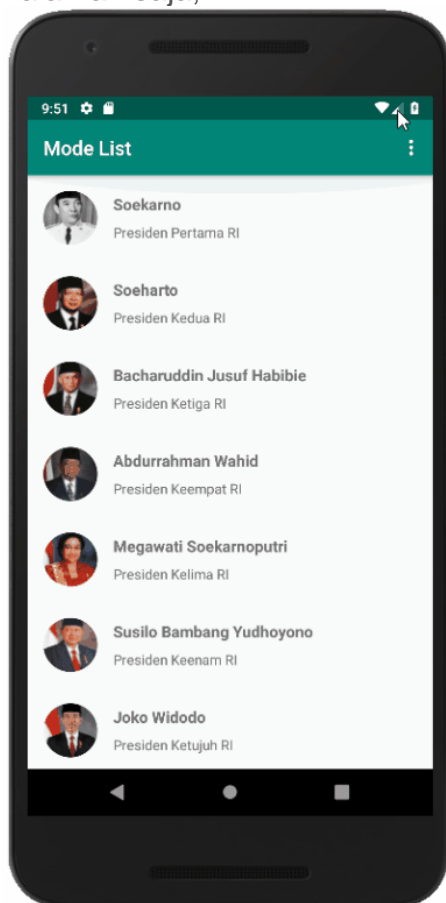
Modul 11.1 – RecyclerView: ListView

Tujuan

Pada codelab kali ini kalian akan mempelajari cara menampilkan data presiden Republik Indonesia ke dalam sebuah RecyclerView. Beberapa poin yang akan diulas dalam materi ini adalah :

1. Bagaimana menggunakan RecyclerView?
2. Berbagai macam RecyclerView seperti *list*, *grid*, dan *card*.
3. Membuat **CustomItemClickListener** pada RecyclerView.
4. Menambahkan menu pada Action Bar.

Contoh dari RecyclerView yang akan kita buat dalam bentuk *list*, *grid* dan *list* dengan bentuk kartu menggunakan cardview dimana semuanya berada dalam satu halaman saja,



Logika Dasar

Melakukan klik ke button → memanggil fragment atau activity dengan atau tanpa data → menampilkan activity atau fragment yang dituju

Codelab

Buat project dengan kriteria sebagai berikut :

Project Name : **MyRecyclerView**

Minimum Level Api : **21**

Default Activity : **Empty Activity**

Activity Name : **MainActivity**

1. Setelah terbentuk, tambahkan beberapa dependensi yang akan kita gunakan pada file **build.gradle (module: app)** di bagian *dependencies* seperti berikut :

```
1. implementation 'com.android.support:appcompat-v7:28.0.0'
2. implementation 'com.android.support:recyclerview-v7:28.0.0'
3. implementation 'com.android.support:cardview-v7:28.0.0'
4. implementation 'de.hdodenhof:circleimageview:2.2.0'
5. implementation 'com.github.bumptech.glide:glide:4.8.0'
```

Sehingga secara keseluruhan berkas **build.gradle(module: app)** Anda akan seperti ini :

```
6. apply plugin: 'com.android.application'
7. android {
8.     compileSdkVersion 28
9.     buildToolsVersion "28.0.3"
10.    defaultConfig {
11.        applicationId "com.dicoding.picodiploma.myrecyclerview"
12.        minSdkVersion 21
13.        targetSdkVersion 28
14.        versionCode 1
15.        versionName "1.0"
16.        testInstrumentationRunner "android.support.test.runner.AndroidJUnit4"
17.    }
18.    buildTypes {
19.        release {
20.            minifyEnabled false
21.            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
22.        }
23.    }
```

```
24. }
25.
26. dependencies {
27.     implementation fileTree(include: ['*.jar'], dir: 'libs')
28.     androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {
29.         exclude group: 'com.android.support', module: 'support-annotations'
30.     })
31.
32.     implementation 'com.android.support:cardview-v7:28.0.0'
33.     implementation 'com.android.support:appcompat-v7:28.0.0'
34.     implementation 'com.android.support:recyclerview-v7:28.0.0'
35.     implementation 'de.hdodenhof:circleimageview:2.2.0'
36.     implementation 'com.github.bumptech.glide:glide:4.8.0'
37.
38.     testImplementation 'junit:junit:4.12'
39. }
```

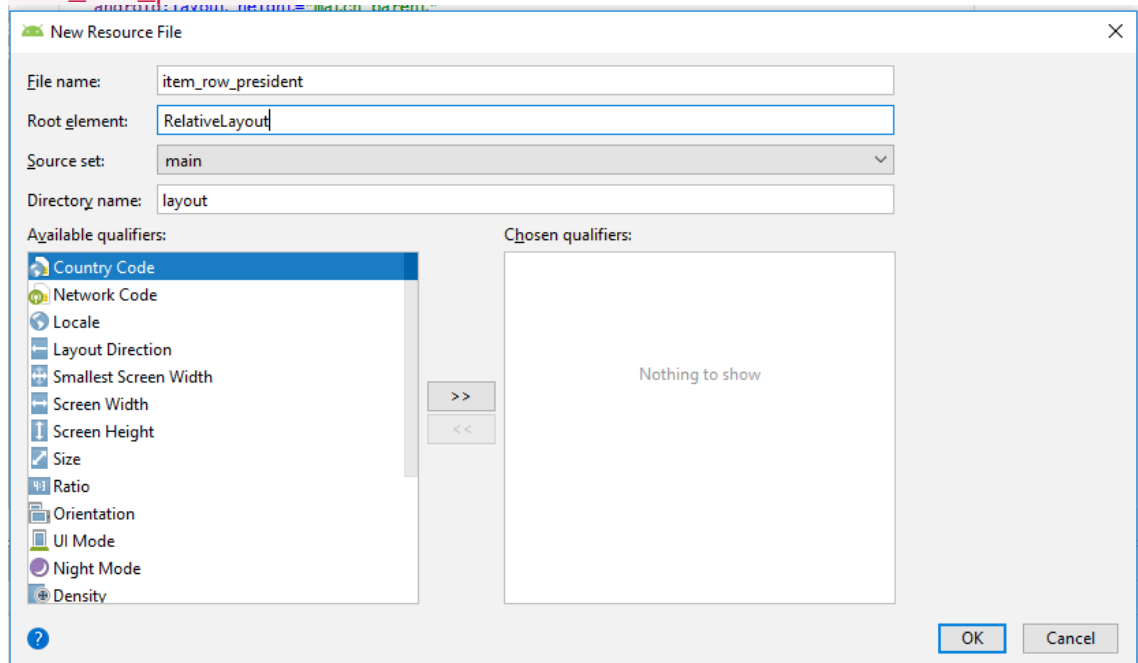
2. Selanjutnya pada `activity_main.xml` lengkapi kodenya menjadi seperti berikut :

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:id="@+id/activity_main"
5.     android:layout_width="match_parent"
6.     android:layout_height="match_parent"
7.     tools:context="com.dicoding.picodiploma.myrecyclerview.MainActivity"
8.     >
9.     <android.support.v7.widget.RecyclerView
10.         android:id="@+id/rv_category"
11.         android:layout_width="match_parent"
12.         android:layout_height="match_parent"
13.         tools:listitem="@layout/item_row_president"/>
14. </RelativeLayout>
```

Akan ada tanda merah di `@layout/item_row_president`. Ini karena layout `item_row_president` belum ditambahkan.

3. Saatnya kita membuat sebuah item tampilan dalam bentuk berkas layout xml yang akan ditampilkan di `RecyclerView`. Karena data pertama kali akan ditampilkan dalam bentuk list, maka kita buat layout dengan cara klik kanan pada direktori **layout** → **new** → **layout resource file** dan kemudian beri nama

item_row_president.



4. Setelah terbentuk kita lengkapi tampilan tersebut menjadi seperti ini :

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:orientation="horizontal"
5.     android:layout_width="match_parent"
6.     android:layout_height="wrap_content"
7.     android:padding="@dimen/activity_vertical_margin">
8.     <de.hdodenhof.circleimageview.CircleImageView
9.         android:id="@+id/img_item_photo"
10.        android:layout_width="55dp"
11.        android:layout_height="55dp"
12.        android:layout_marginRight="@dimen/activity_horizontal_margin"
13.        android:layout_marginEnd="@dimen/activity_horizontal_margin"
14.        tools:src="@android:color/darker_gray"/>
15.     <LinearLayout
16.         android:layout_width="match_parent"
17.         android:layout_height="wrap_content"
18.         android:layout_toRightOf="@id/img_item_photo"
19.         android:layout_toEndOf="@id/img_item_photo"
20.         android:layout_centerVertical="true"
21.         android:orientation="vertical">
22.         <TextView
23.             android:id="@+id/tv_item_name"
24.             android:layout_width="match_parent"
25.             android:layout_height="wrap_content"
26.             tools:text="Soekarno"
27.             android:textStyle="bold"
28.             android:textSize="16sp"
29.             android:layout_marginBottom="8dp"/>
30.         <TextView
31.             android:id="@+id/tv_item_remarks"
32.             android:layout_width="match_parent"

```

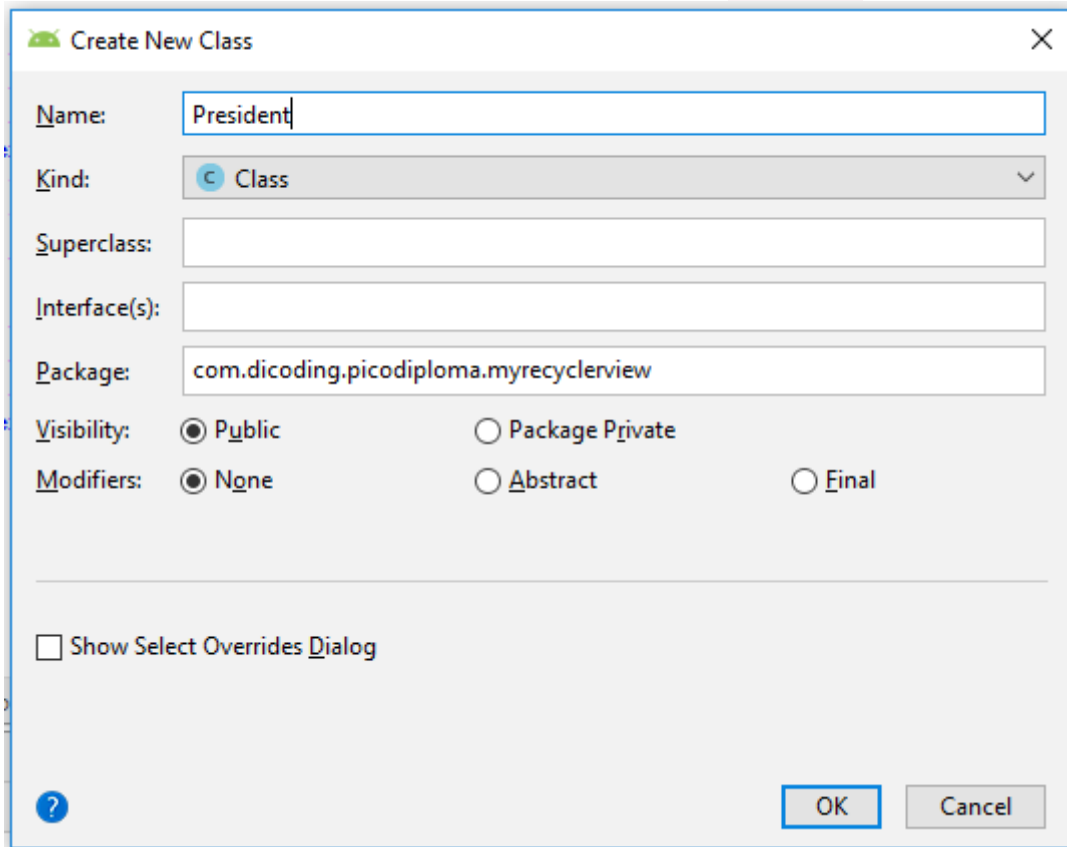
```
33.         android:layout_height="wrap_content"
34.         tools:text="Presiden Pertama RI"/>
35.     </LinearLayout>
36.</RelativeLayout>
```

`tools:text` bisa kita gunakan untuk placeholder di dalam editor layout. Atribut ini tidak akan terbawa saat run-time dan hanya akan nampak di dalam preview editor saja.

Akan ada yang error pada bagian `@dimen/activity_horizontal_margin`, seperti pada modul sebelumnya kita perlu menambahkan berkas `dimens.xml` secara manual di dalam **res** → **values**. Dan isikan berkas `dimens.xml` seperti berikut.

```
1. <resources>
2. <!-- Default screen margins, per the Android Design guidelines. -->
3.     <dimen name="activity_horizontal_margin">16dp</dimen>
4.     <dimen name="activity_vertical_margin">16dp</dimen>
5. </resources>
```

5. Kemudian buat sebuah kelas model data bernama **President**.

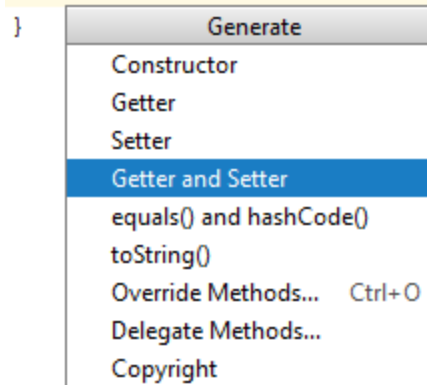


Setelah itu kita bisa menambahkan kode menjadi seperti berikut :

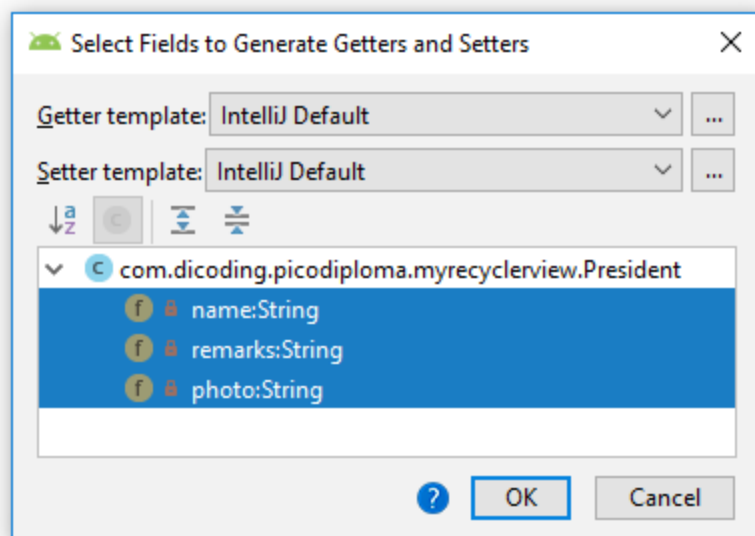
```
1. public class President {
2.     private String name, remarks, photo;
3. }
```

Setelah itu, kita akan buat setter getter, seperti sebelumnya. Bisa kita lakukan dengan **Alt+Insert** atau **Klik Kanan Generate** → **Getter and Setter**

```
public class President {  
    private String name, remarks, photo;
```



Kemudian pilih semua variabel yang sudah dibuat.

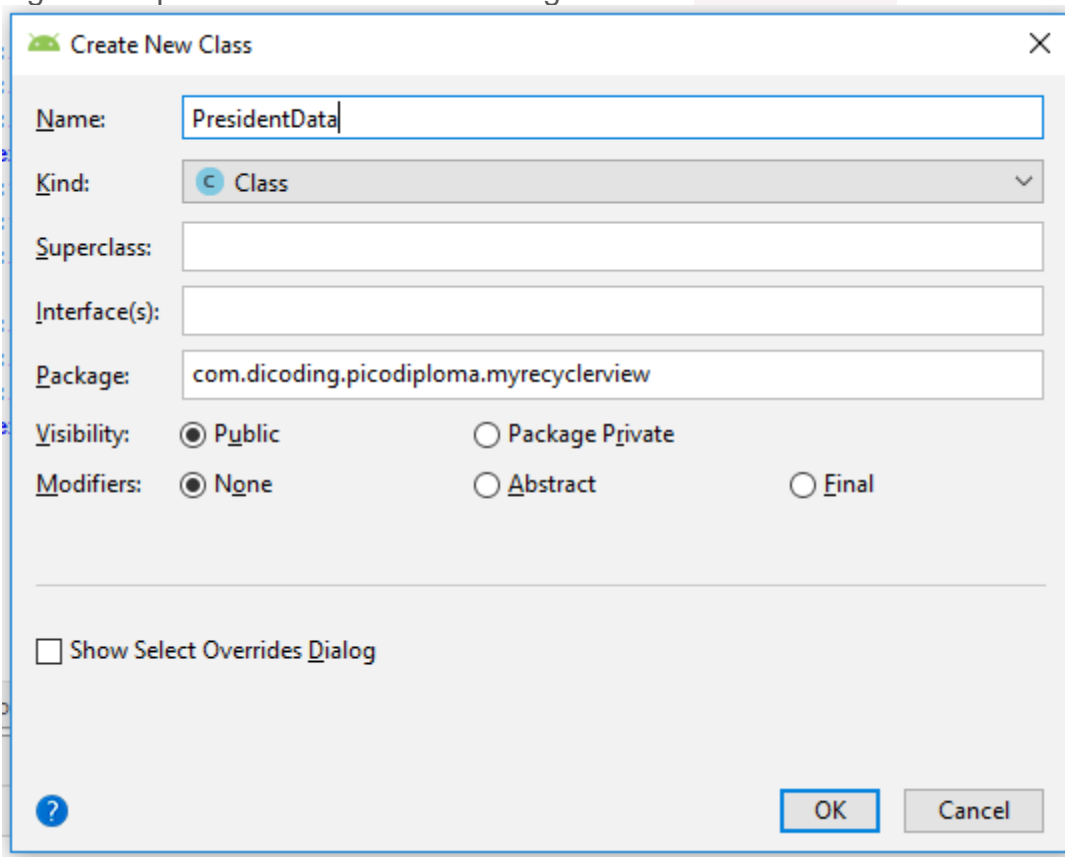


Maka kode `President` akan menjadi seperti ini :

```
4. public class President {  
5.     private String name, remarks, photo;  
6.  
7.     public String getName() {  
8.         return name;  
9.     }  
10.  
11.    public void setName(String name) {  
12.        this.name = name;  
13.    }  
14.  
15.    public String getRemarks() {  
16.        return remarks;  
17.    }  
18.  
19.    public void setRemarks(String remarks) {
```

```
20.     this.remarks = remarks;
21. }
22.
23. public String getPhoto() {
24.     return photo;
25. }
26.
27. public void setPhoto(String photo) {
28.     this.photo = photo;
29. }
30. }
```

6. Setelah selesai kita akan membuat beberapa kelas terkait koleksi data yang ingin ditampilkan. Buat kelas baru dengan nama **PresidentData**.



Setelah itu kita bisa menambahkan kode menjadi seperti berikut :

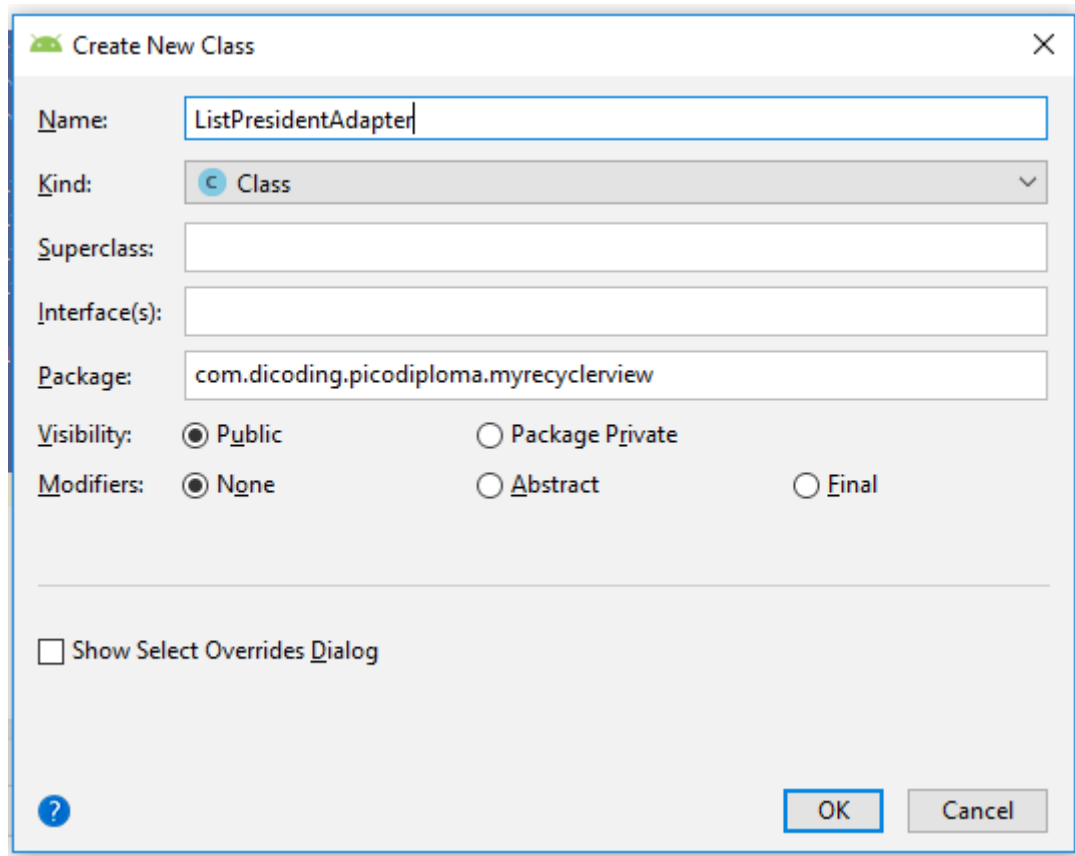
```
1. public class PresidentData {
2.     public static String[][] data = new String[][]{
3.         {"Soekarno", "Presiden Pertama RI", "https://upload.wikimedia.org/wikipedia/commons/thumb/0/01/Presiden_Sukarno.jpg/418px-Presiden_Sukarno.jpg"},
4.         {"Soeharto", "Presiden Kedua RI", "https://upload.wikimedia.org/wikipedia/commons/thumb/5/59/President_Suharto%2C_1993.jpg/468px-President_Suharto%2C_1993.jpg"},
5.         {"Bacharuddin Jusuf Habibie", "Presiden Ketiga RI", "https://upload.wikimedia.org/wikipedia/commons/thumb/f/f1/Bacharuddin_Jusuf_H
```

```

        abiebie_official_portrait.jpg/520px-Bacharuddin_Jusuf_Habibie_official_p
        ortrait.jpg"}},
6.        {"Abdurrahman Wahid", "Presiden Keempat RI", "https://uploa
        d.wikimedia.org/wikipedia/commons/thumb/3/35/President_Abdurrahman_Wahid_-_Indonesia.jpg/486px-President_Abdurrahman_Wahid_-_Indonesia.jpg"},
7.        {"Megawati Soekarnoputri", "Presiden Kelima RI", "https://u
        pload.wikimedia.org/wikipedia/commons/thumb/8/88/President_Megawati_Sukarnoputri_-_Indonesia.jpg/540px-President_Megawati_Sukarnoputri_-_Indon
        esia.jpg"},
8.        {"Susilo Bambang Yudhoyono", "Presiden Keenam RI", "https:/
        /upload.wikimedia.org/wikipedia/commons/5/58/Presiden_Susilo_Bambang_Yu
        dhoyono.png"},
9.        {"Joko Widodo", "Presiden Ketujuh RI", "https://upload.wiki
        media.org/wikipedia/commons/1/1c/Joko_Widodo_2014_official_portrait.jpg
        "}
10.    };
11.
12.    public static ArrayList<President> getListData(){
13.        President president = null;
14.        ArrayList<President> list = new ArrayList<>();
15.        for (String[] aData : data) {
16.            president = new President();
17.            president.setName(aData[0]);
18.            president.setRemarks(aData[1]);
19.            president.setPhoto(aData[2]);
20.
21.            list.add(president);
22.        }
23.
24.        return list;
25.    }
26. }

```

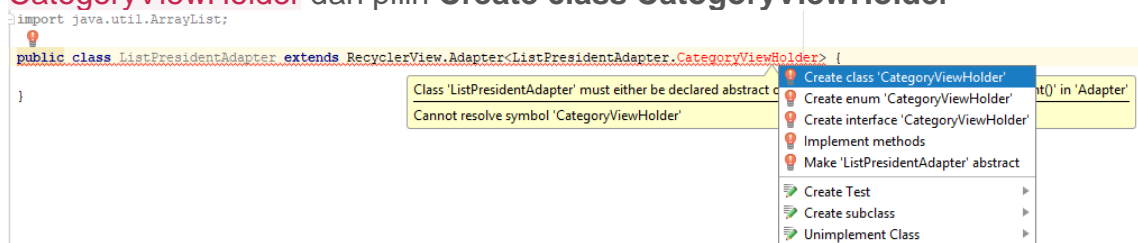
7. Sekarang kita akan membuat sebuah adapter yang akan memformat bagaimana tiap elemen dari koleksi data ditampilkan. Buat kelas adapter secara manual dengan klik kanan pada **package utama** → **new** → **Java Class** dan beri nama **ListPresidentAdapter**.



Setelah tercipta, lengkapi kodenya menjadi seperti berikut :

```
1. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPresidentAdapter.CategoryViewHolder> {
2.
3. }
```

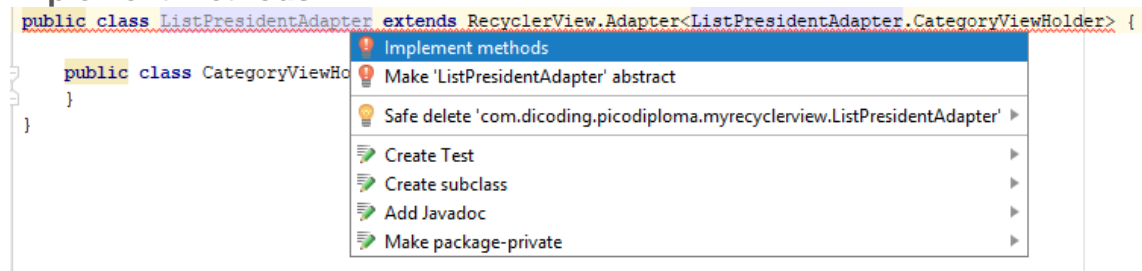
Maka akan ada garis merah dan **CategoryViewHolder** juga akan merah. Kita resolve satu persatu. Pertama adalah kita **alt+enter** / klik tombol merah pada **CategoryViewHolder** dan pilih **Create class CategoryViewHolder**



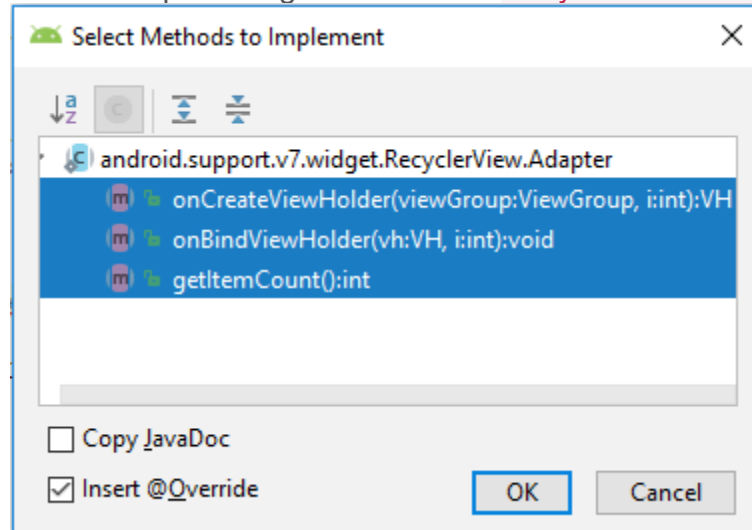
Maka kode saat ini menjadi seperti ini :

```
4. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPresidentAdapter.CategoryViewHolder> {
5.
6.     public class CategoryViewHolder {
7.     }
8. }
```

Masih ada garis merah kita lakukan lagi **alt+enter** / **klik tombol merah** dan pilih **implement methods**.



Kemudian pilih ketiga metode dari **RecyclerView.Adapter**.

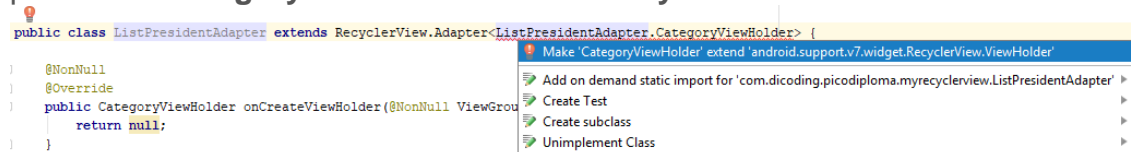


Maka kode saat ini menjadi seperti ini :

```

9. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPres
   identAdapter.CategoryViewHolder> {
10.
11.     @NonNull
12.     @Override
13.     public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup vie
   wGroup, int i) {
14.         return null;
15.     }
16.
17.     @Override
18.     public void onBindViewHolder(@NonNull CategoryViewHolder categoryVi
   ewHolder, int i) {
19.
20.     }
21.
22.     @Override
23.     public int getItemCount() {
24.         return 0;
25.     }
26.
27.     public class CategoryViewHolder {
28.     }
29. }
  
```

Masih ada garis merah kita lakukan lagi **alt+enter** / **klik tombol merah** dan pilih **Make CategoryViewHolder extend RecyclerView.ViewHolder**.

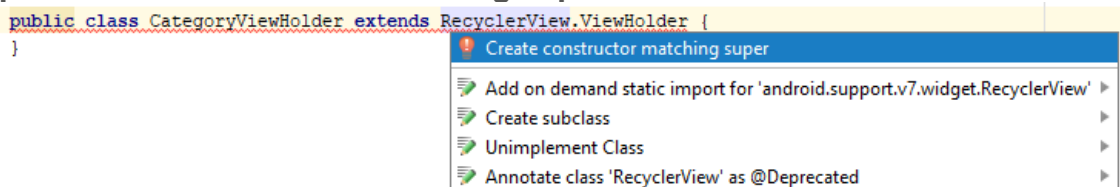


Maka kode saat ini menjadi seperti ini :

```

30. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPresidentAdapter.CategoryViewHolder> {
31.
32.     @NonNull
33.     @Override
34.     public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
35.         return null;
36.     }
37.
38.     @Override
39.     public void onBindViewHolder(@NonNull CategoryViewHolder categoryViewHolder, int i) {
40.
41.     }
42.
43.     @Override
44.     public int getItemCount() {
45.         return 0;
46.     }
47.
48.     public class CategoryViewHolder extends RecyclerView.ViewHolder {
49.     }
50. }
  
```

Masih ada garis merah kita lakukan lagi **alt+enter** / **klik tombol merah** dan pilih **Create constructor matching super**.



Maka kode saat ini menjadi seperti ini :

```

51. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPresidentAdapter.CategoryViewHolder> {
52.
53.     @NonNull
54.     @Override
  
```

```

55.     public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup vie
wGroup, int i) {
56.         return null;
57.     }
58.
59.     @Override
60.     public void onBindViewHolder(@NonNull CategoryViewHolder categoryVi
ewHolder, int i) {
61.
62.     }
63.
64.     @Override
65.     public int getItemCount() {
66.         return 0;
67.     }
68.
69.     public class CategoryViewHolder extends RecyclerView.ViewHolder {
70.         public CategoryViewHolder(@NonNull View itemView) {
71.             super(itemView);
72.         }
73.     }
74. }

```

Selanjutnya adalah kita buat variable untuk *context* dan *list* didalam **ListPresidentAdapter**.

```

75. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPres
identAdapter.CategoryViewHolder> {
76.     private Context context;
77.     private ArrayList<President> listPresident;
78.
79.     ...
80. }

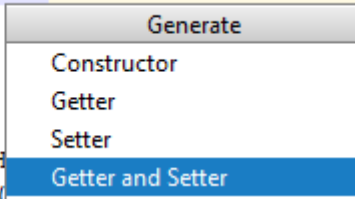
```

Kemudian kita akan membuat **getter** dan **setter**nya untuk **listPresident** dan **Constructor** untuk **Context**. Bisa kita lakukan dengan **Alt+Insert** atau **Klik Kanan Generate → Getter and Setter**.

```

private ArrayList<President> listPresident;

```

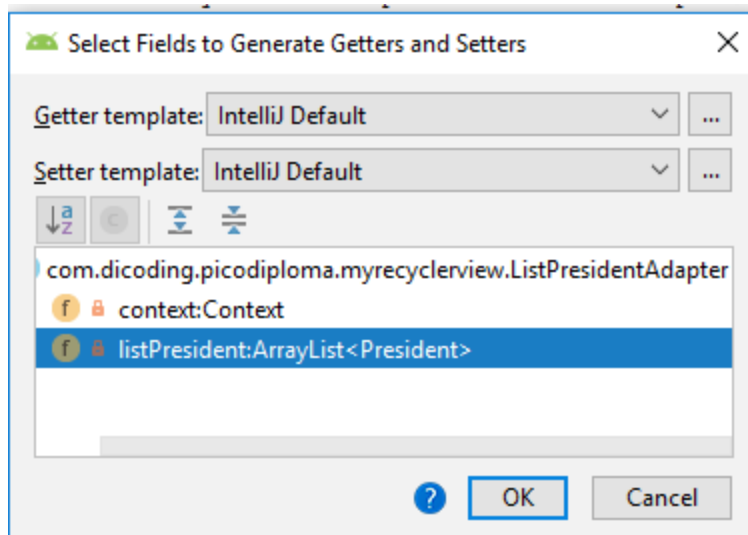


```

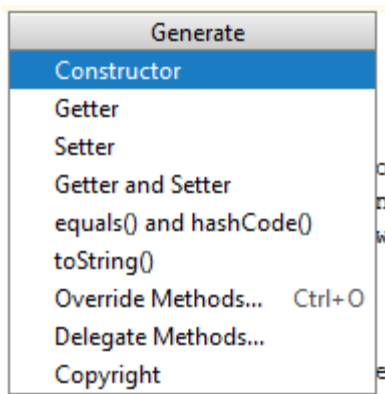
@NonNull
@Override
public CategoryViewHolder onCreateViewHolder
    View itemRow = LayoutInflater.from

```

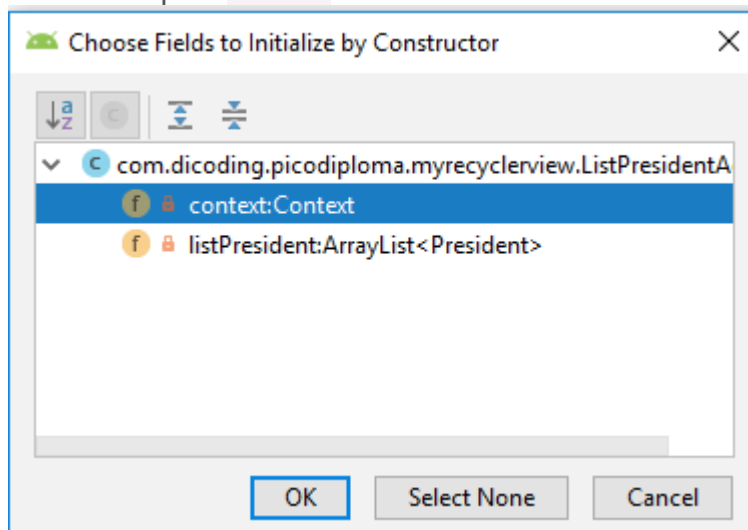
Kemudian pilih **listPresident**.



Kemudian **Constructor** untuk Contextnya.



Kemudian pilih **context**.



menjadi seperti berikut ini :

```

81. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPres
    identAdapter.CategoryViewHolder> {
82.     private Context context;
83.     private ArrayList<President> listPresident;
84.

```

Maka kode saat ini

```
85.     public void setContext(Context context) {
86.         this.context = context;
87.     }
88.
89.     public ArrayList<President> getListPresident() {
90.         return listPresident;
91.     }
92.
93.     public void setListPresident(ArrayList<President> listPresident) {
94.         this.listPresident = listPresident;
95.     }
96.
97.     ...
98. }
```

Setelah membuat **setter** dan **getter**, kita akan mengisi metode yang sudah di implementasikan dari kelas **RecyclerView.Adapter**.

```
99. public class ListPresidentAdapter extends RecyclerView.Adapter<ListPres
    identAdapter.CategoryViewHolder> {
100.     ...
101.
102.     @NonNull
103.     @Override
104.     public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup
        up viewGroup, int i) {
105.         View itemRow = LayoutInflater.from(viewGroup.getContext())
            .inflate(R.layout.item_row_president, viewGroup, false);
106.         return new CategoryViewHolder(itemRow);
107.     }
108.
109.     @Override
110.     public void onBindViewHolder(@NonNull CategoryViewHolder cate
        goryViewHolder, int position) {
111.         categoryViewHolder.tvName.setText(getListPresident().get(
            position).getName());
112.         categoryViewHolder.tvRemarks.setText(getListPresident().g
            et(position).getRemarks());
113.
114.         Glide.with(context)
115.             .load(getListPresident().get(position).getPhoto())
116.             .apply(new RequestOptions().override(55, 55))
117.             .into(categoryViewHolder.imgPhoto);
118.     }
119.
120.     @Override
121.     public int getItemCount() {
122.         return getListPresident().size();
123.     }
124.
125.     class CategoryViewHolder extends RecyclerView.ViewHolder {
126.         TextView tvName;
127.         TextView tvRemarks;
128.         ImageView imgPhoto;
```

```

129.
130.         CategoryViewHolder(@NonNull View itemView) {
131.             super(itemView);
132.             tvName = itemView.findViewById(R.id.tv_item_name);
133.             tvRemarks = itemView.findViewById(R.id.tv_item_remark
134.             s);
135.             imgPhoto = itemView.findViewById(R.id.img_item_photo)
136.             ;
137.         }
138.     }
139. }
140. }
141. }

```

Maka kode `ListPresidentAdapter` menjadi seperti ini :

```

138.     public class ListPresidentAdapter extends RecyclerView.Adapter<Li
139.         stPresidentAdapter.CategoryViewHolder> {
140.         private Context context;
141.         private ArrayList<President> listPresident;
142.
143.         private ArrayList<President> getListPresident() {
144.             return listPresident;
145.         }
146.
147.         void setListPresident(ArrayList<President> listPresident) {
148.             this.listPresident = listPresident;
149.         }
150.
151.         ListPresidentAdapter(Context context) {
152.             this.context = context;
153.         }
154.
155.         @NonNull
156.         @Override
157.         public CategoryViewHolder onCreateViewHolder(@NonNull ViewGro
158.             up viewGroup, int i) {
159.             View itemRow = LayoutInflater.from(viewGroup.getContext())
160.                 .inflate(R.layout.item_row_president, viewGroup, false);
161.             return new CategoryViewHolder(itemRow);
162.         }
163.
164.         @Override
165.         public void onBindViewHolder(@NonNull CategoryViewHolder cate
166.             goryViewHolder, int position) {
167.             categoryViewHolder.tvName.setText(getListPresident().get(
168.                 position).getName());
169.             categoryViewHolder.tvRemarks.setText(getListPresident().g
170.                 et(position).getRemarks());
171.
172.             Glide.with(context)
173.                 .load(getListPresident().get(position).getPhoto())
174.                 .apply(new RequestOptions().override(55, 55))
175.                 .into(categoryViewHolder.imgPhoto);
176.         }
177.
178.         @Override

```

```
173.         public int getItemCount() {
174.             return getListPresident().size();
175.         }
176.
177.         class CategoryViewHolder extends RecyclerView.ViewHolder {
178.             TextView tvName;
179.             TextView tvRemarks;
180.             ImageView imgPhoto;
181.
182.             CategoryViewHolder(@NonNull View itemView) {
183.                 super(itemView);
184.                 tvName = itemView.findViewById(R.id.tv_item_name);
185.                 tvRemarks = itemView.findViewById(R.id.tv_item_remark
186. s);
186.                 imgPhoto = itemView.findViewById(R.id.img_item_photo)
187. ;
187.             }
188.         }
189.     }
```

8. Kemudian kita modifikasi berkas kelas **MainActivity** kita kenalkan **RecycleView** yang sudah dibuat di **activity_main**.

```
1. public class MainActivity extends AppCompatActivity {
2.     private RecyclerView rvCategory;
3.
4.     @Override
5.     protected void onCreate(Bundle savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_main);
8.
9.         rvCategory = findViewById(R.id.rv_category);
10.        rvCategory.setHasFixedSize(true);
11.    }
12. }
```

9. Selanjutnya setelah dikenalkan, kita akan masukkan data yang sudah kita buat di **PresidentData**.

```
1. public class MainActivity extends AppCompatActivity {
2.     private RecyclerView rvCategory;
3.     private ArrayList<President> list = new ArrayList<>();
4.
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.
10.        rvCategory = findViewById(R.id.rv_category);
11.        rvCategory.setHasFixedSize(true);
12.
13.        list.addAll(PresidentData.getListData());
```



```

14.     showRecyclerList();
15. }
16.
17. private void showRecyclerList(){
18.     rvCategory.setLayoutManager(new LinearLayoutManager(this));
19.     ListPresidentAdapter listPresidentAdapter = new ListPresidentAd
    apter(this);
20.     listPresidentAdapter.setListPresident(list);
21.     rvCategory.setAdapter(listPresidentAdapter);
22. }
23. }

```

10. Karena data gambar yang kita miliki berasal dari internet maka kita harus menambahkan sebuah permission ke dalam file **AndroidManifest.xml** yang kita miliki. Tambahkan satu baru berikut di atas `tag <application>`.

```

1. <uses-permission android:name="android.permission.INTERNET" />

```

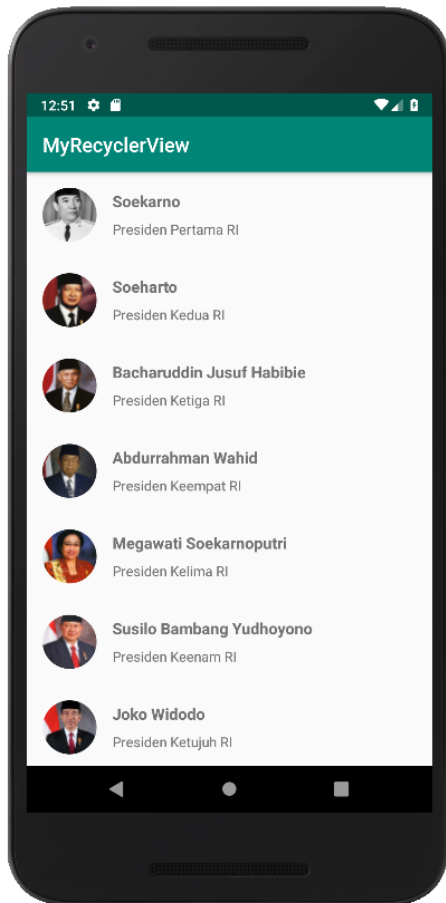
Sehingga secara keseluruhan berkas **AndroidManifest.xml** yang kita miliki saat ini menjadi seperti berikut :

```

2. <?xml version="1.0" encoding="utf-8"?>
3. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
4.     package="com.dicoding.picodiploma.myrecyclerview">
5.     <uses-permission android:name="android.permission.INTERNET" />
6.     <application
7.         android:allowBackup="true"
8.         android:icon="@mipmap/ic_launcher"
9.         android:label="@string/app_name"
10.        android:roundIcon="@mipmap/ic_launcher_round"
11.        android:supportRtl="true"
12.        android:theme="@style/AppTheme">
13.         <activity android:name=".MainActivity">
14.             <intent-filter>
15.                 <action android:name="android.intent.action.MAIN" />
16.
17.                 <category android:name="android.intent.category.LAUNCHE
    R" />
18.             </intent-filter>
19.         </activity>
20.     </application>
21.
22. </manifest>

```

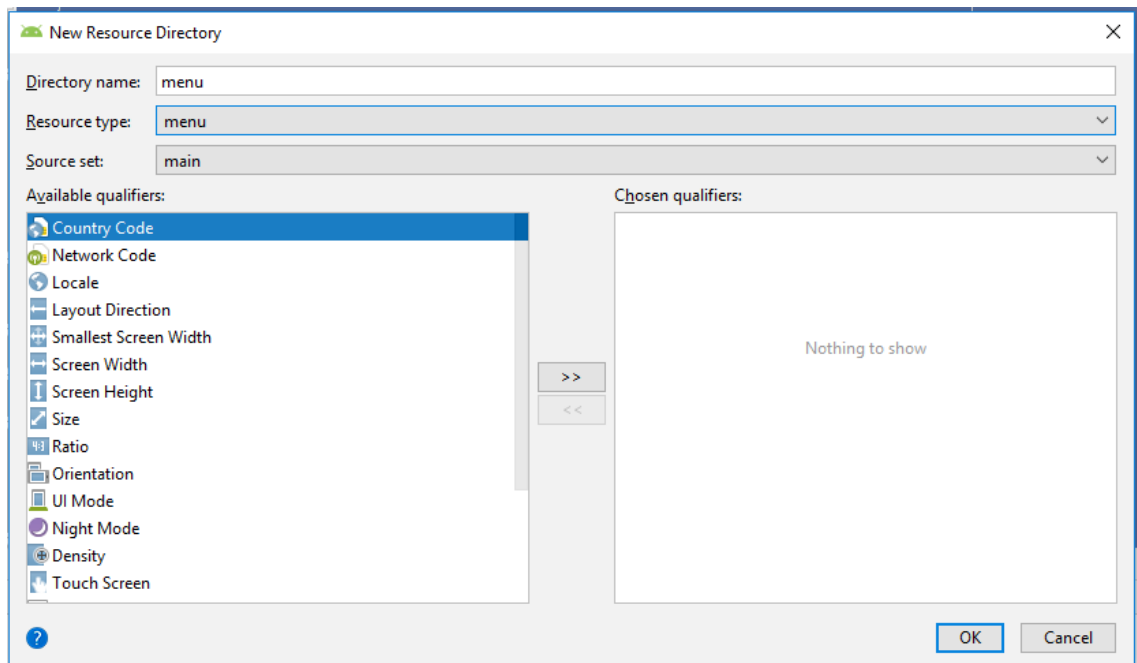
11. Jalankan aplikasi yang dibuat. Hasilnya kurang lebih akan seperti gambar di bawah ini :



Sekarang mari kita buat sebuah **RecyclerView** dengan memanfaatkan fasilitas menu.

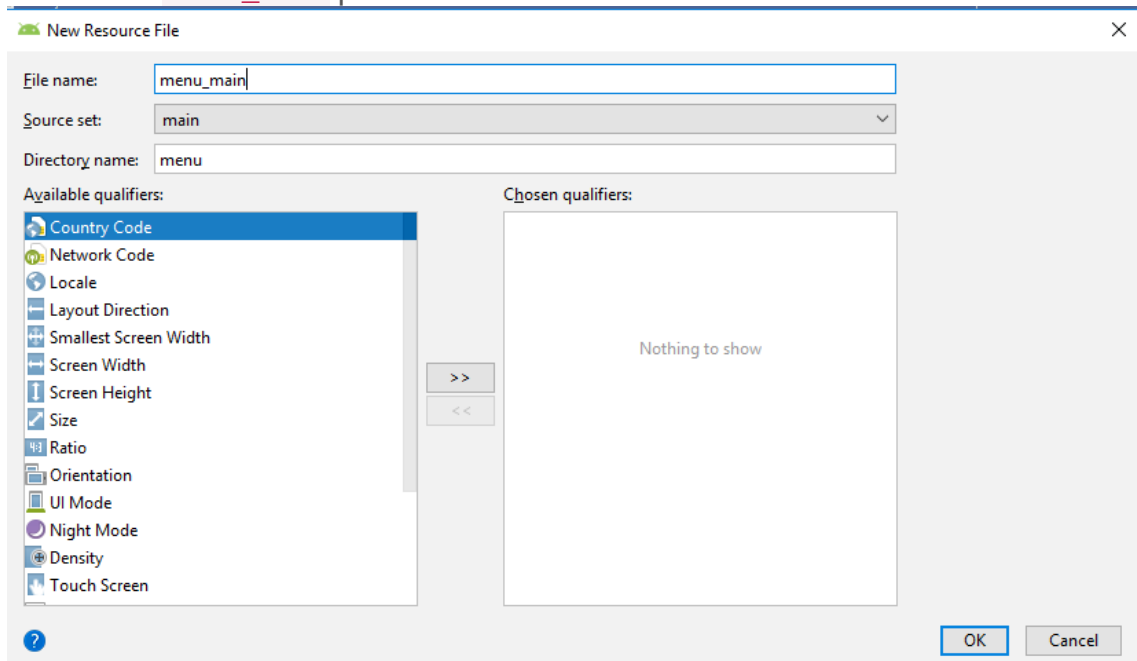
- Langkah pertama adalah dengan membuat *resource directory* terkait. *Resource directory* menu secara bawaan tidak disediakan. Kita harus membuatnya terlebih dahulu.

Klik kanan pada direktori **res** → **new** → **android resource directory**. Setelah muncul dialog box seperti di bawah ini, isikan menu pada *field directory name*. Klik **OK** untuk menyelesaikannya.



Sebuah **RecyclerView** untuk menampilkan data Presiden Republik Indonesia sudah tercipta.

13. Kemudian, **klik kanan pada direktori tersebut** → **new** → **menu resource file**.
 Beri nama **menu_main** pada **field file name**.



Setelah itu, lengkapi kodenya menjadi seperti berikut :

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <menu xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto">
4.     <item
5.         android:id="@+id/action_list"
6.         android:title="List"
7.         app:showAsAction="never"/>
  
```

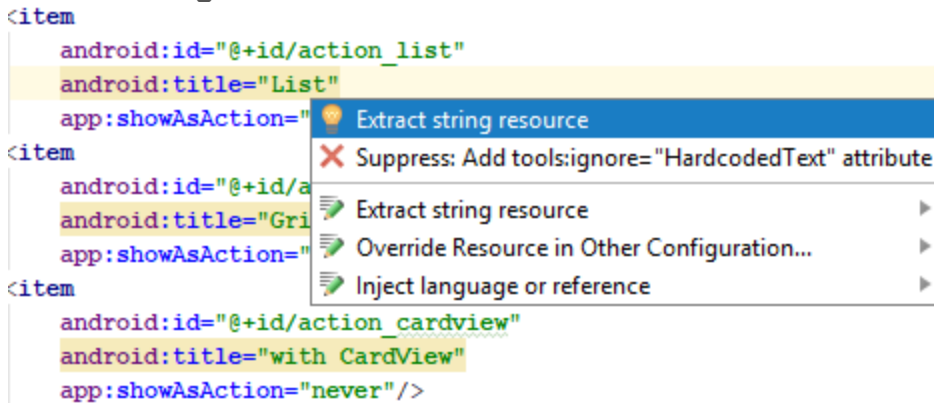
```

8.      <item
9.          android:id="@+id/action_grid"
10.         android:title="Grid"
11.         app:showAsAction="never"/>
12.      <item
13.          android:id="@+id/action_cardview"
14.          android:title="with CardView"
15.          app:showAsAction="never"/>
16. </menu>

```

Jangan lupa untuk memindahkan text ke **string.xml**. Caranya **alt+enter** →

Extract string resource.



```

<item
    android:id="@+id/action_list"
    android:title="List"
    app:showAsAction="
<item
    android:id="@+id/a
    android:title="Gri
    app:showAsAction="
<item
    android:id="@+id/action_cardview"
    android:title="with CardView"
    app:showAsAction="never"/>

```

Maka menu_main menjadi seperti berikut :

```

17. <?xml version="1.0" encoding="utf-8"?>
18. <menu xmlns:android="http://schemas.android.com/apk/res/android"
19.     xmlns:app="http://schemas.android.com/apk/res-auto">
20.     <item
21.         android:id="@+id/action_list"
22.         android:title="@string/list"
23.         app:showAsAction="never" />
24.     <item
25.         android:id="@+id/action_grid"
26.         android:title="@string/grid"
27.         app:showAsAction="never" />
28.     <item
29.         android:id="@+id/action_cardview"
30.         android:title="@string/with_cardview"
31.         app:showAsAction="never" />
32. </menu>

```

Dan pada bagian **res** → **values** → **strings.xml**.

```

33. <resources>
34.     <string name="app_name">MyRecyclerView</string>
35.     <string name="list">List</string>
36.     <string name="grid">Grid</string>
37.     <string name="with_cardview">with CardView</string>

```

```
38. </resources>
```

14. Setelah selesai, saatnya kita pasang menu tersebut di **MainActivity** dengan menambahkan dua metode berikut :

```
1. public class MainActivity extends AppCompatActivity {
2.     ...
3.
4.     @Override
5.     public boolean onCreateOptionsMenu(Menu menu) {
6.         getMenuInflater().inflate(R.menu.menu_main, menu);
7.         return super.onCreateOptionsMenu(menu);
8.     }
9.
10.    @Override
11.    public boolean onOptionsItemSelected(MenuItem item) {
12.        switch (item.getItemId()){
13.            case R.id.action_list:
14.                break;
15.
16.            case R.id.action_grid:
17.                break;
18.
19.            case R.id.action_cardview:
20.                break;
21.        }
22.        return super.onOptionsItemSelected(item);
23.    }
24. }
```

Untuk dua metode di atas, Anda bisa menggunakan **Ctrl (tahan) + Spasi** untuk menampilkan *code assistant* pada Android Studio.

```
    showRecyclerList();
}

private void showRecyclerList(){
    rvCategory.setLayoutManager(new LinearLayoutManager( context: this));
    ListPresidentAdapter listPresidentAdapter = new ListPresidentAdapter( context: this);
    listPresidentAdapter.setListPresident(list);
    rvCategory.setAdapter(listPresidentAdapter);
}

}
```

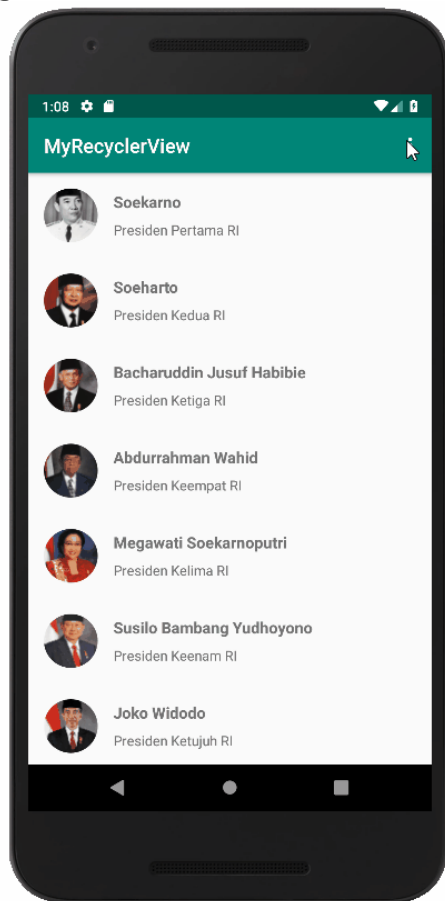
Kode di kelas **MainActivity** akan berubah menjadi :

```
25. public class MainActivity extends AppCompatActivity {
26.     private RecyclerView rvCategory;
27.     private ArrayList<President> list = new ArrayList<>();
28. }
```

```
29.     @Override
30.     protected void onCreate(Bundle savedInstanceState) {
31.         super.onCreate(savedInstanceState);
32.         setContentView(R.layout.activity_main);
33.
34.         rvCategory = findViewById(R.id.rv_category);
35.         rvCategory.setHasFixedSize(true);
36.
37.         list.addAll(PresidentData.getListData());
38.         showRecyclerList();
39.     }
40.
41.     private void showRecyclerList(){
42.         rvCategory.setLayoutManager(new LinearLayoutManager(this));
43.         ListPresidentAdapter listPresidentAdapter = new ListPresidentAd
44. apter(this);
45.         listPresidentAdapter.setListPresident(list);
46.         rvCategory.setAdapter(listPresidentAdapter);
47.
48.     @Override
49.     public boolean onCreateOptionsMenu(Menu menu) {
50.         getMenuInflater().inflate(R.menu.menu_main, menu);
51.         return super.onCreateOptionsMenu(menu);
52.     }
53.
54.     @Override
55.     public boolean onOptionsItemSelected(MenuItem item) {
56.         switch (item.getItemId()){
57.             case R.id.action_list:
58.                 break;
59.
60.             case R.id.action_grid:
61.                 break;
62.
63.             case R.id.action_cardview:
64.                 break;
65.         }
66.         return super.onOptionsItemSelected(item);
67.     }
68. }
```

Jalankan kembali aplikasi Anda. Tampilannya kurang lebih akan menjadi seperti

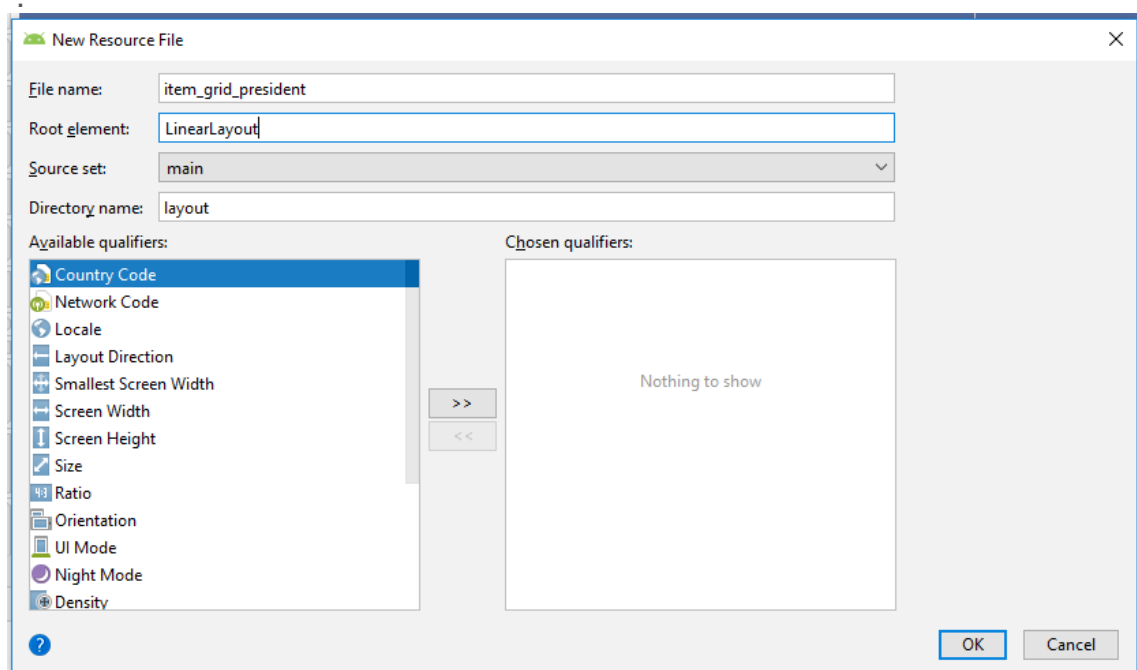
gambar di bawah ini :



Modul 11.2 – RecyclerView: GridView

Pada modul sebelumnya, kita telah membuat menu pada aplikasi daftar presiden yang sedang kita buat. Mari kita lanjutkan pembuatan bentuk dari **RecyclerView** dalam bentuk *grid*.

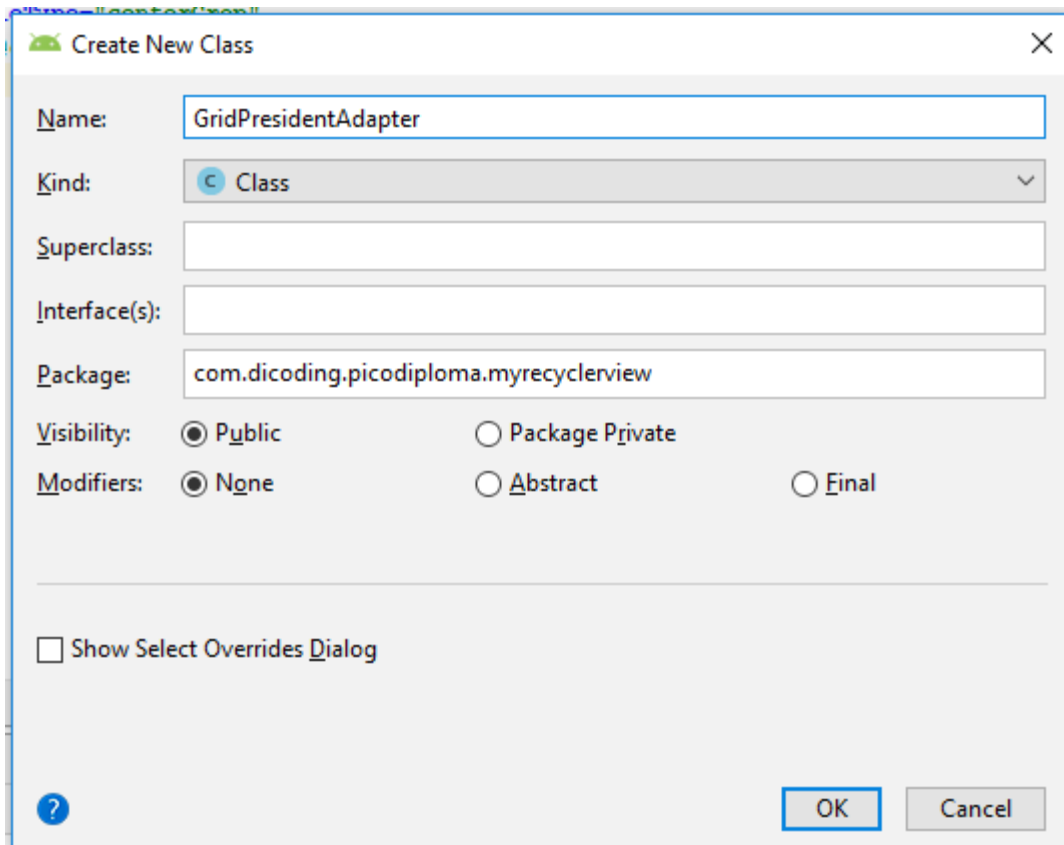
1. Buat sebuah berkas layout xml baru dengan nama **item_grid_president**



Setelah itu lengkapi kodenya menjadi seperti berikut :

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="wrap_content"
6.     android:orientation="vertical">
7.
8.     <ImageView
9.         android:id="@+id/img_item_photo"
10.        android:layout_width="match_parent"
11.        android:layout_height="250dp"
12.        android:layout_margin="1dp"
13.        android:scaleType="centerCrop"
14.        tools:src="@color/colorAccent" />
15. </LinearLayout>
```


- Setelah selesai, lanjut ke pembuatan Adapter untuk file layout xml tersebut. Buat sebuah kelas baru dengan nama **GridPresidentAdapter**.



Kemudian kita lakukan hal yang sama dengan pembuatan **ListPresidentAdapter**, lengkapi kodenya menjadi seperti berikut :

```
1. public class GridPresidentAdapter extends RecyclerView.Adapter<GridPresidentAdapter.GridViewHolder> {  
2. }
```

Setelah itu kita akan memperbaiki error dengan cara **alt+enter** atau **klik lampu merahnya**. Setelah memperbaiki maka akan menjadi seperti ini :

```
3. public class GridPresidentAdapter extends RecyclerView.Adapter<GridPresidentAdapter.GridViewHolder> {  
4.     @NonNull  
5.     @Override  
6.     public GridViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {  
7.         return null;  
8.     }  
9.  
10.    @Override  
11.    public void onBindViewHolder(@NonNull GridViewHolder gridViewHolder, int i) {
```

```
12.
13.     }
14.
15.     @Override
16.     public int getItemCount() {
17.         return 0;
18.     }
19.
20.     public class GridViewHolder extends RecyclerView.ViewHolder {
21.         public GridViewHolder(@NonNull View itemView) {
22.             super(itemView);
23.         }
24.     }
25. }
```

Kemudian kita bisa bikin **setter** dan **getter** untuk *list* dan **constructor** untuk *context* seperti ini :

```
26. public class GridPresidentAdapter extends RecyclerView.Adapter<GridPres
    identAdapter.GridViewHolder> {
27.     private Context context;
28.     private ArrayList<President> listPresident;
29.
30.     private ArrayList<President> getListPresident() {
31.         return listPresident;
32.     }
33.
34.     public void setListPresident(ArrayList<President> listPresident) {
35.         this.listPresident = listPresident;
36.     }
37.
38.     public GridPresidentAdapter(Context context) {
39.         this.context = context;
40.     }
41.
42.     ...
43. }
```

Setelah itu kita bisa melengkapi kode kode hasil dari turunan **RecyclerView.Adapter**. Maka kode lengkap dari kelas **GridPresidentAdapter** sebagai berikut :

```
44. public class GridPresidentAdapter extends RecyclerView.Adapter<GridPres
    identAdapter.GridViewHolder> {
45.     private Context context;
46.     private ArrayList<President> listPresident;
47.
48.     private ArrayList<President> getListPresident() {
49.         return listPresident;
50.     }
51. }
```

```
52.     public void setListPresident(ArrayList<President> listPresident) {
53.         this.listPresident = listPresident;
54.     }
55.
56.     public GridPresidentAdapter(Context context) {
57.         this.context = context;
58.     }
59.
60.     @NonNull
61.     @Override
62.     public GridViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
        int viewType) {
63.         View view = LayoutInflater.from(parent.getContext()).inflate(R.
        layout.item_grid_president, parent, false);
64.         return new GridViewHolder(view);
65.     }
66.
67.     @Override
68.     public void onBindViewHolder(@NonNull GridViewHolder holder, int po
        sition) {
69.         Glide.with(context)
70.             .load(getListPresident().get(position).getPhoto())
71.             .apply(new RequestOptions().override(350, 550))
72.             .into(holder.imgPhoto);
73.     }
74.
75.     @Override
76.     public int getItemCount() {
77.         return getListPresident().size();
78.     }
79.
80.     class GridViewHolder extends RecyclerView.ViewHolder {
81.         ImageView imgPhoto;
82.
83.         GridViewHolder(View itemView) {
84.             super(itemView);
85.             imgPhoto = itemView.findViewById(R.id.img_item_photo);
86.         }
87.     }
88. }
```

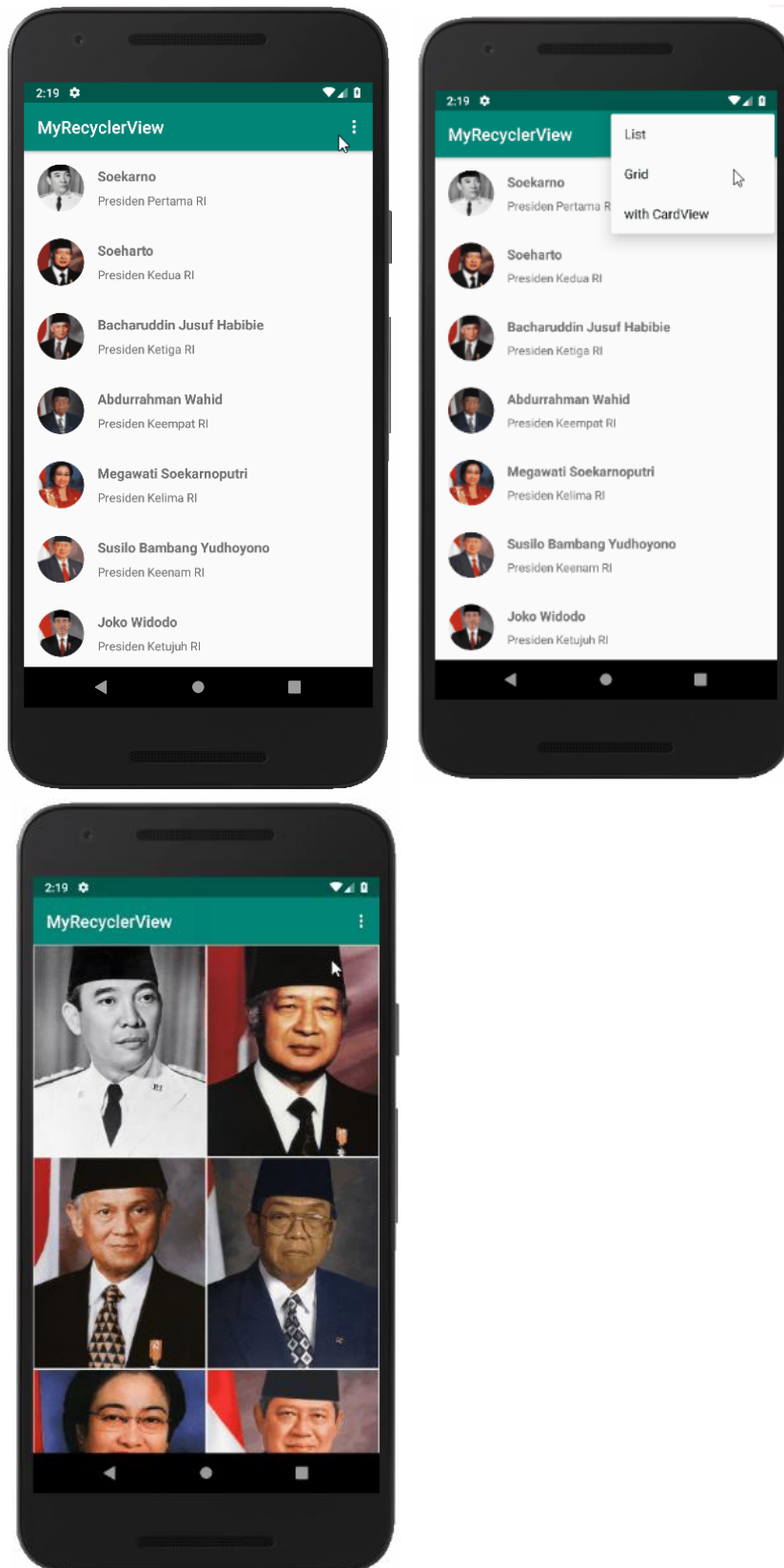
3. Setelah semuanya selesai, mari kita pasang lagi di **MainActivity** dengan menambahkan satu metode berikut :

```
1. private void showRecyclerGrid(){
2.     rvCategory.setLayoutManager(new GridLayoutManager(this, 2));
3.     GridPresidentAdapter gridPresidentAdapter = new GridPresidentAdapte
        r(this);
4.     gridPresidentAdapter.setListPresident(list);
5.     rvCategory.setAdapter(gridPresidentAdapter);
6. }
```

Dan lengkapi metode **onOptionsItemSelected()** menjadi seperti berikut :

```
7. @Override
8. public boolean onOptionsItemSelected(MenuItem item) {
9.     switch (item.getItemId()){
10.         case R.id.action_list:
11.             showRecyclerList();
12.             break;
13.         case R.id.action_grid:
14.             showRecyclerGrid();
15.             break;
16.         case R.id.action_cardview:
17.             break;
18.     }
19.     return super.onOptionsItemSelected(item);
20. }
```

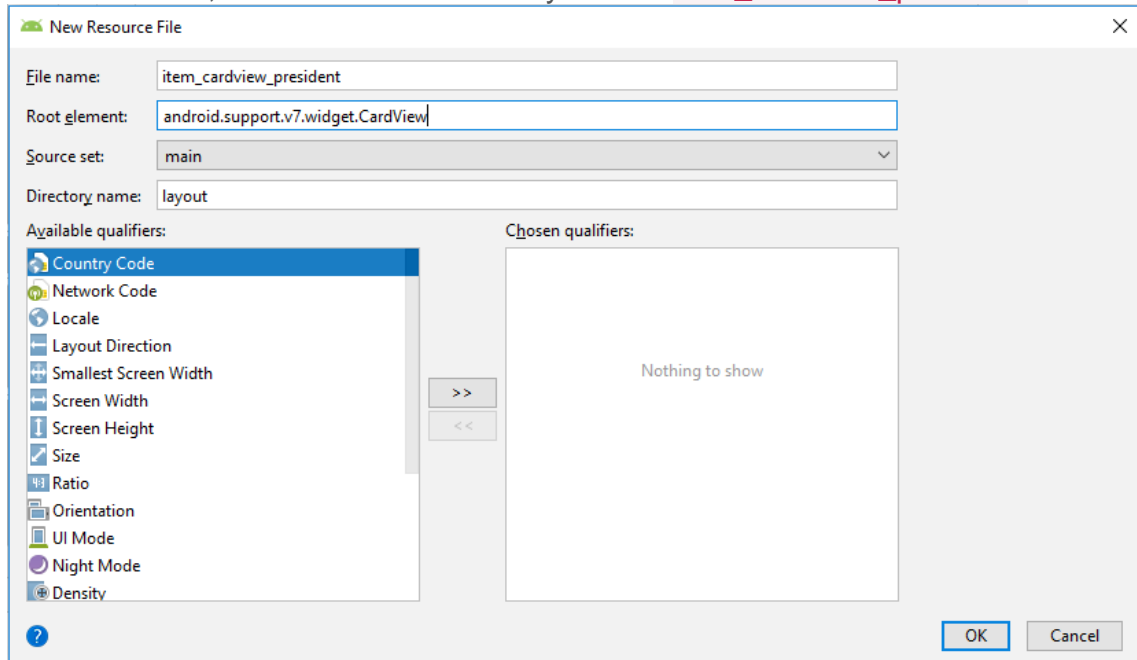
4. Ini akan mengubah tampilan ketika pengguna memilih salah satu bentuk layout. Sekarang jalankan kembali aplikasinya. Akan ada dua bentuk dari **RecyclerView** yaitu *list* dan *grid*.



Modul 11.3 – RecyclerView: CardView

Pada modul ini, kita akan membuat bentuk **RecyclerView** dengan komponen **CardView**. Komponen ini akan menampilkan data selayaknya sebuah kartu. Pendekatan ini hanya diperuntukkan bila item list memiliki lebih dari satu *action*.

1. Baik. Pertama, buat kembali berkas layout xml **item_cardview_president**.



Setelah itu lengkapi kodenya menjadi seperti berikut :

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:card_view="http://schemas.android.com/apk/res-auto"
4.     xmlns:tools="http://schemas.android.com/tools"
5.     android:id="@+id/card_view"
6.     android:layout_gravity="center"
7.     android:layout_width="match_parent"
8.     android:layout_height="wrap_content"
9.     android:layout_marginTop="4dp"
10.    android:layout_marginBottom="4dp"
11.    android:layout_marginLeft="8dp"
12.    android:layout_marginRight="8dp"
13.    card_view:cardCornerRadius="4dp">
14.    <RelativeLayout
15.        android:layout_width="match_parent"
16.        android:layout_height="200dp"
17.        android:padding="8dp">
18.        <ImageView
19.            android:layout_width="150dp"
20.            android:layout_height="220dp"
21.            android:scaleType="centerCrop"
22.            android:id="@+id/img_item_photo"

```

```

23.         android:layout_marginBottom="4dp"
24.         tools:src="@color/colorAccent"/>
25.     <TextView
26.         android:layout_toRightOf="@id/img_item_photo"
27.         android:layout_toEndOf="@id/img_item_photo"
28.         android:id="@+id/tv_item_name"
29.         android:layout_width="match_parent"
30.         android:layout_height="wrap_content"
31.         android:textSize="16sp"
32.         android:textStyle="bold"
33.         tools:text="Soekarno"
34.         android:layout_marginTop="@dimen/activity_vertical_margin"
35.         android:layout_marginBottom="8dp"
36.         android:layout_marginRight="@dimen/activity_vertical_margin"
37.         android:layout_marginLeft="@dimen/activity_vertical_margin"
38.     />
39.     <TextView
40.         android:layout_below="@id/tv_item_name"
41.         android:layout_toRightOf="@id/img_item_photo"
42.         android:layout_toEndOf="@id/img_item_photo"
43.         android:id="@+id/tv_item_remarks"
44.         android:layout_width="match_parent"
45.         android:layout_height="wrap_content"
46.         tools:text="Remarks"
47.         android:layout_marginBottom="8dp"
48.         android:layout_marginRight="@dimen/activity_vertical_margin"
49.         android:layout_marginLeft="@dimen/activity_vertical_margin"
50.     />
51.     <LinearLayout
52.         android:layout_width="match_parent"
53.         android:layout_height="wrap_content"
54.         android:layout_alignParentBottom="true"
55.         android:layout_toRightOf="@id/img_item_photo"
56.         android:layout_toEndOf="@id/img_item_photo"
57.         android:layout_marginLeft="@dimen/activity_horizontal_margi
58.     n"
59.         android:layout_marginStart="@dimen/activity_horizontal_marg
60.     in"
61.         android:orientation="horizontal">
62.         <Button
63.             android:id="@+id/btn_set_favorite"
64.             android:layout_width="match_parent"
65.             android:layout_height="wrap_content"
66.             android:textSize="12sp"
67.             style="@style/Widget.AppCompat.Button.Colored"
68.             android:text="@string/favorite"
69.             android:layout_weight="1"/>
70.         <Button
71.             android:id="@+id/btn_set_share"
72.             android:layout_width="match_parent"
73.             android:layout_height="wrap_content"
74.             android:textSize="12sp"
75.             style="@style/Widget.AppCompat.Button.Colored"
76.             android:text="@string/share"
77.             android:layout_weight="1"/>
78.     </LinearLayout>
79. </RelativeLayout>

```

```
76. </android.support.v7.widget.CardView>
```

Setelah memasukkan view kedalam `item_cardview_president`. Kita pindahkan teks ke `strings.xml`.

```
<Button
    android:id="@+id/btn_set_favorite"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="12sp"
    style="@style/Widget.AppCompat.Button.Colored"
    android:text="Favorite"
    android:layout_weight="1"/>
```

Context menu options:

- Extract string resource
- Suppress: Add tools:ignore="HardcodedText" attribute
- Extract string resource
- Override Resource in Other Configuration...
- Inject language or reference

```
<Button
    android:id="@+id/btn_set_favorite"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="12sp"
    style="@style/Widget.AppCompat.Button.Colored"
    android:text="Share"
    android:layout_weight="1"/>
```

Maka file `strings.xml` akan menjadi seperti ini :

```
77. <resources>
78.     <string name="app_name">MyRecyclerView</string>
79.     <string name="list">List</string>
80.     <string name="grid">Grid</string>
81.     <string name="with_cardview">with CardView</string>
82.     <string name="favorite">Favorite</string>
83.     <string name="share">Share</string>
84. </resources>
```

Dan kode keseluruhan dari `item_cardview_president` akan menjadi seperti ini :

```
85. <?xml version="1.0" encoding="utf-8"?>
86. <android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
87.     xmlns:card_view="http://schemas.android.com/apk/res-auto"
88.     xmlns:tools="http://schemas.android.com/tools"
89.     android:id="@+id/card_view"
90.     android:layout_width="match_parent"
91.     android:layout_height="wrap_content"
92.     android:layout_gravity="center"
93.     android:layout_marginLeft="8dp"
94.     android:layout_marginTop="4dp"
95.     android:layout_marginRight="8dp"
96.     android:layout_marginBottom="4dp"
97.     card_view:cardCornerRadius="4dp">
98.
99.     <RelativeLayout
100.         android:layout_width="match_parent"
101.         android:layout_height="200dp">
```



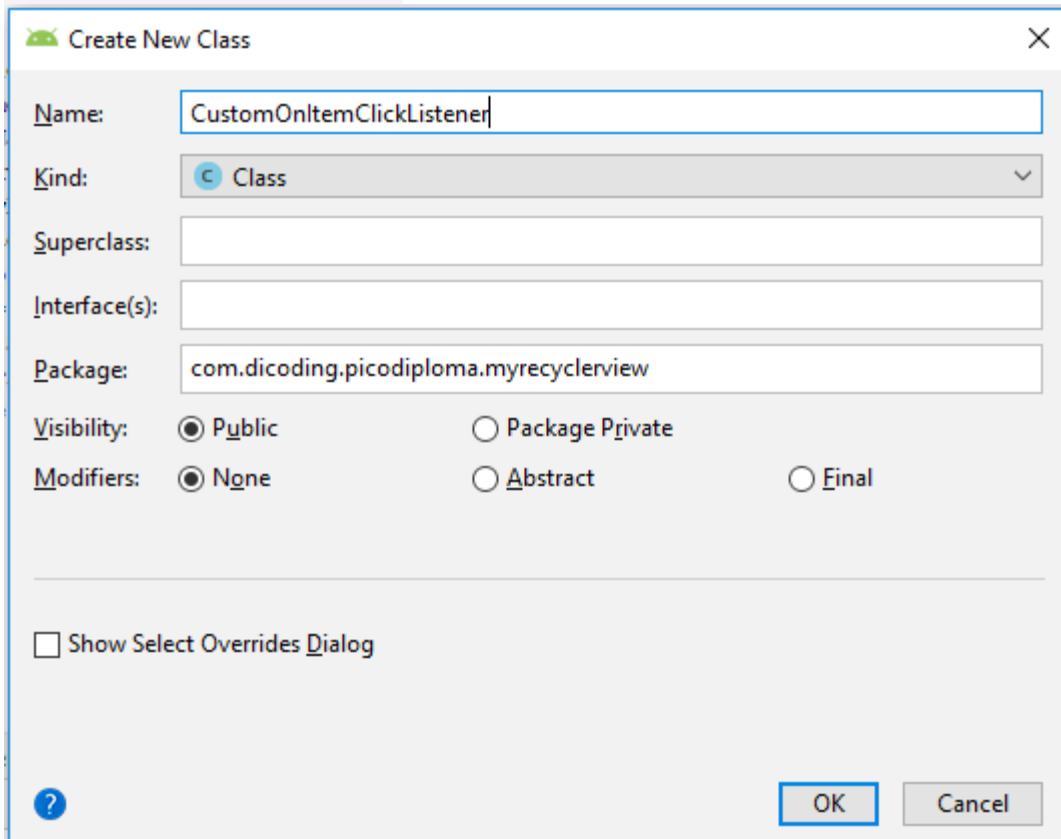
```

102.         android:padding="8dp">
103.
104.         <ImageView
105.             android:id="@+id/img_item_photo"
106.             android:layout_width="150dp"
107.             android:layout_height="220dp"
108.             android:layout_marginBottom="4dp"
109.             android:scaleType="centerCrop"
110.             tools:src="@color/colorAccent" />
111.
112.         <TextView
113.             android:id="@+id/tv_item_name"
114.             android:layout_width="match_parent"
115.             android:layout_height="wrap_content"
116.             android:layout_marginLeft="@dimen/activity_vertical_m
argin"
117.             android:layout_marginTop="@dimen/activity_vertical_ma
rgin"
118.             android:layout_marginRight="@dimen/activity_vertical_
margin"
119.             android:layout_marginBottom="8dp"
120.             android:layout_toEndOf="@id/img_item_photo"
121.             android:layout_toRightOf="@id/img_item_photo"
122.             android:textSize="16sp"
123.             android:textStyle="bold"
124.             tools:text="Soekarno" />
125.
126.         <TextView
127.             android:id="@+id/tv_item_remarks"
128.             android:layout_width="match_parent"
129.             android:layout_height="wrap_content"
130.             android:layout_below="@id/tv_item_name"
131.             android:layout_marginLeft="@dimen/activity_vertical_m
argin"
132.             android:layout_marginRight="@dimen/activity_vertical_
margin"
133.             android:layout_marginBottom="8dp"
134.             android:layout_toEndOf="@id/img_item_photo"
135.             android:layout_toRightOf="@id/img_item_photo"
136.             tools:text="Remarks" />
137.
138.         <LinearLayout
139.             android:layout_width="match_parent"
140.             android:layout_height="wrap_content"
141.             android:layout_alignParentBottom="true"
142.             android:layout_marginStart="@dimen/activity_hizonta
l_margin"
143.             android:layout_marginLeft="@dimen/activity_horizontal
_margin"
144.             android:layout_toEndOf="@id/img_item_photo"
145.             android:layout_toRightOf="@id/img_item_photo"
146.             android:orientation="horizontal">
147.
148.         <Button
149.             android:id="@+id/btn_set_favorite"
150.             style="@style/Widget.AppCompat.Button.Colored"
151.             android:layout_width="match_parent"
152.             android:layout_height="wrap_content"
153.             android:layout_weight="1"

```

```
154.         android:text="@string/favorite"
155.         android:textSize="12sp" />
156.
157.     <Button
158.         android:id="@+id/btn_set_share"
159.         style="@style/Widget.AppCompat.Button.Colored"
160.         android:layout_width="match_parent"
161.         android:layout_height="wrap_content"
162.         android:layout_weight="1"
163.         android:text="@string/share"
164.         android:textSize="12sp" />
165.     </LinearLayout>
166. </RelativeLayout>
167. </android.support.v7.widget.CardView>
```

2. Selanjutnya kita buat terlebih dahulu kelas baru dengan nama **CustomOnItemClickListener**.



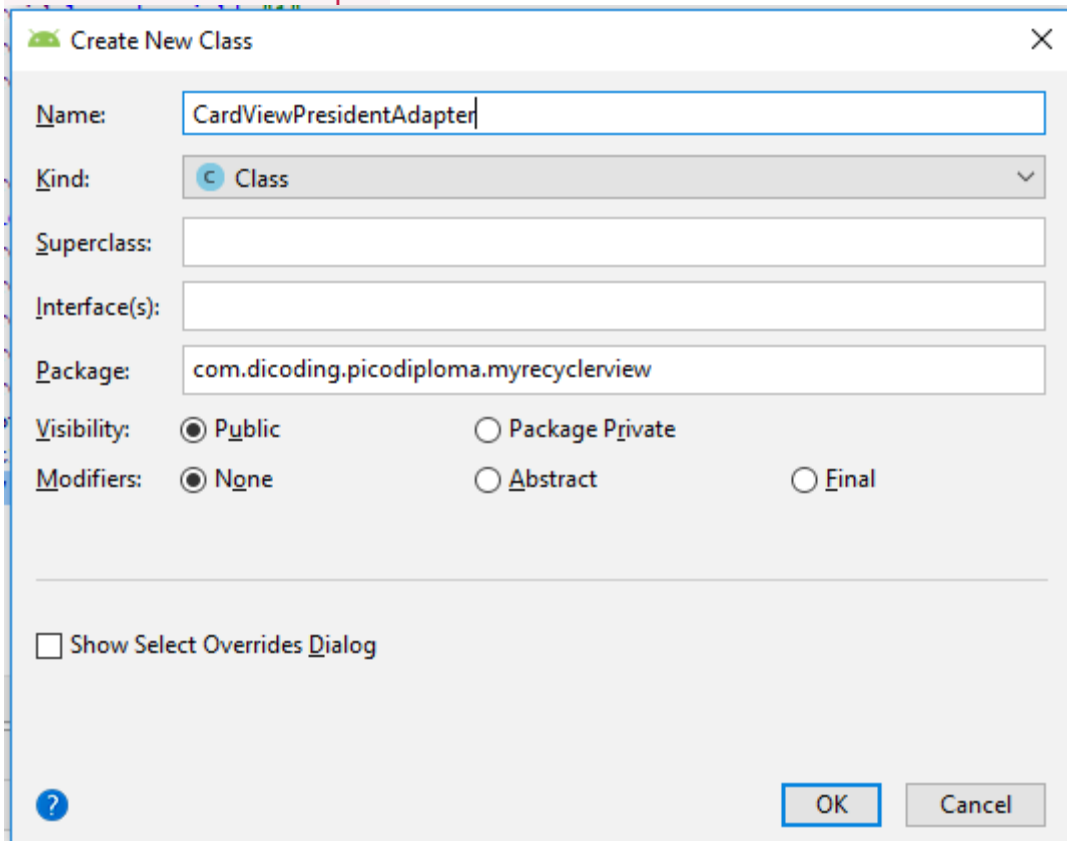
The screenshot shows the 'Create New Class' dialog box. The 'Name' field contains 'CustomOnItemClickListener'. The 'Kind' dropdown is set to 'Class'. The 'Package' field contains 'com.dicoding.picodiploma.myrecyclerview'. The 'Visibility' section has 'Public' selected. The 'Modifiers' section has 'None' selected. There are 'OK' and 'Cancel' buttons at the bottom right.

Kelas ini nantinya yang berfungsi untuk menangani aksi ketika itemnya di klik. Kodenya menjadi seperti berikut :

```
1. public class CustomOnItemClickListener implements View.OnClickListener
2. {
3.     private int position;
4.     private OnItemClickListener onItemClickListener;
5.     CustomOnItemClickListener(int position, OnItemClickListener onItemClickListener) {
6.         this.position = position;
```

```
6.         this.onItemClickListener = onItemClickListener;
7.     }
8.
9.     @Override
10.    public void onClick(View view) {
11.        onItemClickListener.onItemClicked(view, position);
12.    }
13.    public interface OnItemClickListener {
14.        void onItemClicked(View view, int position);
15.    }
16. }
```

3. Setelah selesai, buat kembali kelas Adapter dengan nama **CardViewPresidentAdapter**.



Kemudian kita lakukan hal yang sama dengan pembuatan **ListPresidentAdapter**, lengkapi kodenya menjadi seperti berikut :

```
1. public class CardViewPresidentAdapter extends RecyclerView.Adapter<Card
  ViewPresidentAdapter.CardViewHolder>{
2. }
```

Setelah itu kita akan memperbaiki error dengan cara **alt+enter** atau **klik lampu merahnya**. Setelah memperbaiki maka akan menjadi seperti ini :

```

3. public class CardViewPresidentAdapter extends RecyclerView.Adapter<Card
   ViewPresidentAdapter.CardViewViewHolder>{
4.     @NonNull
5.     @Override
6.     public CardViewViewHolder onCreateViewHolder(@NonNull ViewGroup vie
   wGroup, int i) {
7.         return null;
8.     }
9.
10.    @Override
11.    public void onBindViewHolder(@NonNull CardViewViewHolder cardViewVi
   ewHolder, int i) {
12.
13.    }
14.
15.    @Override
16.    public int getItemCount() {
17.        return 0;
18.    }
19.
20.    public class CardViewViewHolder extends RecyclerView.ViewHolder {
21.        public CardViewViewHolder(@NonNull View itemView) {
22.            super(itemView);
23.        }
24.    }
25. }

```

Kemudian kita bisa bikin *setter* dan *getter* untuk *list* dan *constructor* untuk *context* seperti ini :

```

26. public class CardViewPresidentAdapter extends RecyclerView.Adapter<Card
   ViewPresidentAdapter.CardViewViewHolder>{
27.     private Context context;
28.     private ArrayList<President> listPresident;
29.     private ArrayList<President> getListPresident() {
30.         return listPresident;
31.     }
32.     public void setListPresident(ArrayList<President> listPresident) {
33.         this.listPresident = listPresident;
34.     }
35.     public CardViewPresidentAdapter(Context context) {
36.         this.context = context;
37.     }
38.
39.     ...
40. }

```

Setelah itu kita bisa melengkapi kode kode hasil dari turunan `RecyclerView.Adapter`. Maka kode lengkap dari kelas `CardPresidentAdapter` sebagai berikut

```

41. public class CardViewPresidentAdapter extends RecyclerView.Adapter<Card
    ViewPresidentAdapter.CardViewViewHolder>{
42.     private Context context;
43.     private ArrayList<President> listPresident;
44.     private ArrayList<President> getListPresident() {
45.         return listPresident;
46.     }
47.     public void setListPresident(ArrayList<President> listPresident) {
48.         this.listPresident = listPresident;
49.     }
50.     public CardViewPresidentAdapter(Context context) {
51.         this.context = context;
52.     }
53.
54.     @NonNull
55.     @Override
56.     public CardViewViewHolder onCreateViewHolder(@NonNull ViewGroup vie
        wGroup, int i) {
57.         View view = LayoutInflater.from(viewGroup.getContext()).inflate
            (R.layout.item_cardview_president, viewGroup, false);
58.         return new CardViewViewHolder(view);
59.     }
60.
61.     @Override
62.     public void onBindViewHolder(@NonNull CardViewViewHolder cardViewVi
        ewHolder, int i) {
63.         President p = getListPresident().get(i);
64.
65.         Glide.with(context)
66.             .load(p.getPhoto())
67.             .apply(new RequestOptions().override(350, 550))
68.             .into(cardViewViewHolder.imgPhoto);
69.
70.         cardViewViewHolder.tvName.setText(p.getName());
71.         cardViewViewHolder.tvRemarks.setText(p.getRemarks());
72.
73.         cardViewViewHolder.btnFavorite.setOnClickListener(new CustomOnI
            temClickListener(i, new CustomOnItemClickListener.OnItemClickCallback()
            {
74.                 @Override
75.                 public void onItemClickClicked(View view, int position) {
76.                     Toast.makeText(context, "Favorite "+getListPresident().
                        get(position).getName(), Toast.LENGTH_SHORT).show();
77.                 }
78.             }));
79.
80.         cardViewViewHolder.btnShare.setOnClickListener(new CustomOnItem
            ClickListener(i, new CustomOnItemClickListener.OnItemClickCallback() {
81.                 @Override
82.                 public void onItemClickClicked(View view, int position) {
83.                     Toast.makeText(context, "Share "+getListPresident().get
                        (position).getName(), Toast.LENGTH_SHORT).show();
84.                 }
85.             }));
86.     }
87.
88.     @Override
89.     public int getItemCount() {
90.         return getListPresident().size();

```

```
91.     }
92.
93.     class CardViewViewHolder extends RecyclerView.ViewHolder{
94.         ImageView imgPhoto;
95.         TextView tvName, tvRemarks;
96.         Button btnFavorite, btnShare;
97.         CardViewViewHolder(View itemView) {
98.             super(itemView);
99.             imgPhoto = itemView.findViewById(R.id.img_item_photo);
100.            tvName = itemView.findViewById(R.id.tv_item_name);
101.            tvRemarks = itemView.findViewById(R.id.tv_item_remark
102.            s);
103.            btnFavorite = itemView.findViewById(R.id.btn_set_favo
104.            rite);
105.            btnShare = itemView.findViewById(R.id.btn_set_share);
106.        }
107.    }
```

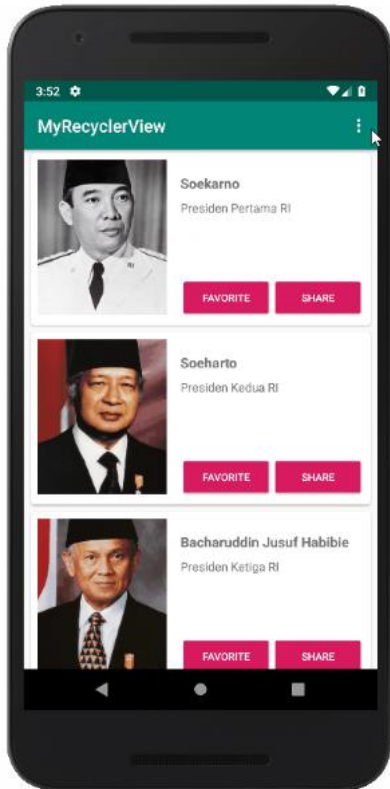
4. Setelah selesai, kita perlu menambahkan satu metode lagi untuk menampilkan bentuk **RecyclerView** ini. Caranya lengkapi kode pada **MainActivity** menjadi seperti berikut:

```
1. private void showRecyclerCardView(){
2.     rvCategory.setLayoutManager(new LinearLayoutManager(this));
3.     CardViewPresidentAdapter cardViewPresidentAdapter = new CardViewPre
4.     sidentAdapter(this);
5.     cardViewPresidentAdapter.setListPresident(list);
6.     rvCategory.setAdapter(cardViewPresidentAdapter);
7. }
```

Dan update metode **onOptionsItemSelected()** menjadi seperti berikut :

```
7. @Override
8. public boolean onOptionsItemSelected(MenuItem item) {
9.
10.     switch (item.getItemId()){
11.         case R.id.action_list:
12.             showRecyclerList();
13.             break;
14.
15.         case R.id.action_grid:
16.             showRecyclerGrid();
17.             break;
18.
19.         case R.id.action_cardview:
20.             showRecyclerCardView();
21.             break;
22.     }
23.     return super.onOptionsItemSelected(item);
24. }
```

5. Jalankan kembali aplikasinya. Seharusnya Anda sudah bisa menampilkan tiga bentuk **RecyclerView** yang telah kita rencanakan sebelumnya.



6. Untuk mempercantik tampilan dan memenuhi target di awal, kita akan melakukan penambahan metode untuk mengubah judul halaman setiap kita melakukan perubahan bentuk dari **RecyclerView**.

```
1. private void setActionBarTitle(String title){
2.     getSupportActionBar().setTitle(title);
3. }
```

Kemudian tambahkan *variable* baru diatas metode **onCreate()** seperti berikut :

```
4. private RecyclerView rvCategory;
5. private ArrayList<President> list = new ArrayList<>();
6. private String title = "Mode List";
7.
8. @Override
9. protected void onCreate(Bundle savedInstanceState) {
10.     super.onCreate(savedInstanceState);
11.     setContentView(R.layout.activity_main);
12.     ....
13. }
```

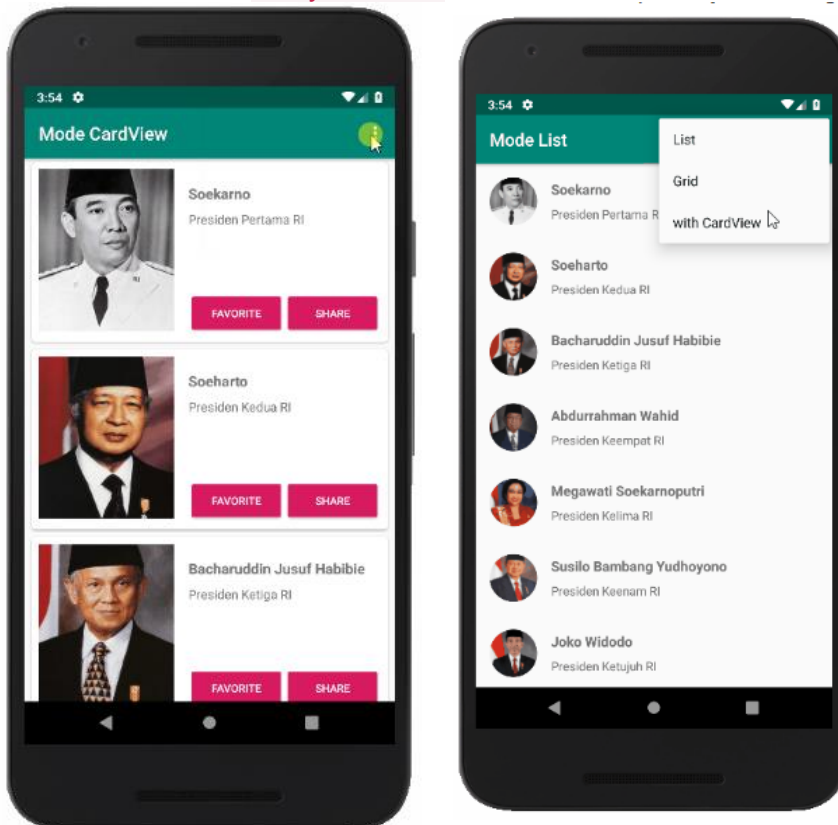
```
14. }
```

Dan kita *update* kembali metode `onOptionsItemSelected()` sehingga menjadi seperti berikut :

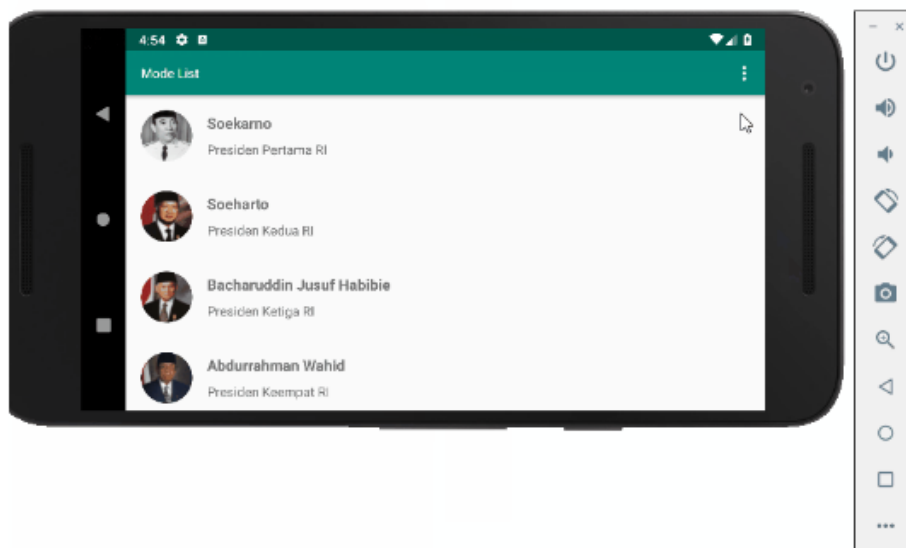
```
15. @Override
16. public boolean onOptionsItemSelected(MenuItem item) {
17.     switch (item.getItemId()){
18.         case R.id.action_list:
19.             setActionBarTitle("Mode List");
20.             showRecyclerList();
21.             break;
22.
23.         case R.id.action_grid:
24.             setActionBarTitle("Mode Grid");
25.             showRecyclerGrid();
26.             break;
27.
28.         case R.id.action_cardview:
29.             setActionBarTitle("Mode CardView");
30.             showRecyclerCardView();
31.             break;
32.     }
33.     return super.onOptionsItemSelected(item);
34. }
```

Dan pada metode `onCreate()` tambahkan baris ini `setActionBarTitle("Mode List");` sebelum kita memanggil metode `showRecyclerList();`

7. Jalankan kembali aplikasinya. Sekarang judul halaman akan berubah seiring perubahan bentuk **RecyclerView**.



8. Selanjutnya jika kita rotate dari posisi *grid* atau dari posisi *cardview*, pasti akan kembali ke tampilan *list*.



Ini terjadi karena data belum bisa terjaga. Kita tambahkan kode berikut untuk menyimpan state tersebut.

```
1. public class MainActivity extends AppCompatActivity {
2.     RecyclerView rvCategory;
3.     private ArrayList<President> list;
4.     final String STATE_TITLE = "state_string";
5.     final String STATE_LIST = "state_list";
6.     final String STATE_MODE = "state_mode";
7.     int mode;
8.
9.     @Override
10.    protected void onCreate(Bundle savedInstanceState) {
11.        super.onCreate(savedInstanceState);
12.        setContentView(R.layout.activity_main);
13.        ...
14.
15.        if (savedInstanceState == null) {
16.            setActionBarTitle("Mode List");
17.            list.addAll(PresidentData.getListData());
18.            showRecyclerList();
19.            mode = R.id.action_list;
20.        } else {
21.            String stateTitle = savedInstanceState.getString(STATE_TITL
E);
22.            ArrayList<President> stateList= savedInstanceState.getParc
eableArrayList(STATE_LIST);
23.            int stateMode = savedInstanceState.getInt(STATE_MODE);
24.            setActionBarTitle(stateTitle);
25.            list.addAll(stateList);
26.            setMode(stateMode);
27.        }
28.    }
29.
30.    ...
31.
32.    @Override
33.    protected void onSaveInstanceState(Bundle outState) {
34.        super.onSaveInstanceState(outState);
35.        outState.putString(STATE_TITLE, getSupportActionBar().getTitle(
).toString());
36.        outState.putParcelableArrayList(STATE_LIST, list);
37.        outState.putInt(STATE_MODE, mode);
38.    }
39. }
```

Maka kode di MainActivity akan menjadi seperti berikut ini :

```
40. public class MainActivity extends AppCompatActivity {
41.     RecyclerView rvCategory;
42.     private ArrayList<President> list;
43.     final String STATE_TITLE = "state_string";
44.     final String STATE_LIST = "state_list";
45.     final String STATE_MODE = "state_mode";
46.     int mode;
47.
48.     @Override
49.    protected void onCreate(Bundle savedInstanceState) {
50.        super.onCreate(savedInstanceState);
```

```

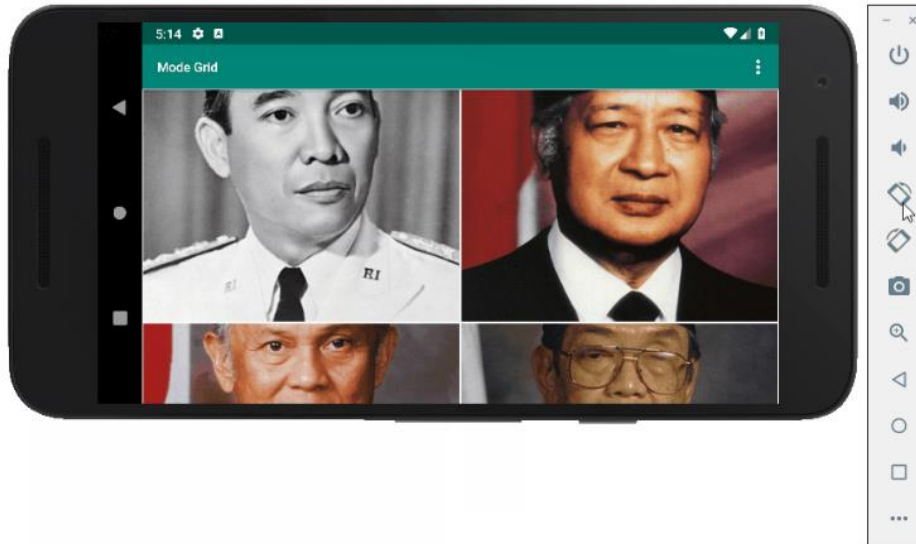
51.         setContentView(R.layout.activity_main);
52.
53.         rvCategory = findViewById(R.id.rv_category);
54.         rvCategory.setHasFixedSize(true);
55.
56.         list = new ArrayList<>();
57.
58.         if (savedInstanceState == null) {
59.             setActionBarTitle("Mode List");
60.             list.addAll(PresidentData.getListData());
61.             showRecyclerList();
62.             mode = R.id.action_list;
63.
64.         } else {
65.             String stateTitle = savedInstanceState.getString(STATE_TITL
E);
66.             ArrayList<President> stateList = savedInstanceState.getParc
eableArrayList(STATE_LIST);
67.             int stateMode = savedInstanceState.getInt(STATE_MODE);
68.             setActionBarTitle(stateTitle);
69.             list.addAll(stateList);
70.             setMode(stateMode);
71.         }
72.     }
73.
74.     private void showSelectedPresident(President president) {
75.         Toast.makeText(this, "Kamu memilih " + president.getName(), Toa
st.LENGTH_SHORT).show();
76.     }
77.
78.     private void showRecyclerList() {
79.         rvCategory.setLayoutManager(new LinearLayoutManager(this));
80.         ListPresidentAdapter listPresidentAdapter = new ListPresidentAd
apter(this);
81.         listPresidentAdapter.setListPresident(list);
82.         rvCategory.setAdapter(listPresidentAdapter);
83.
84.         ItemClickSupport.addTo(rvCategory).setOnItemClickListener(new I
temClickSupport.OnItemClickListener() {
85.             @Override
86.             public void onItemClick(RecyclerView recyclerView, int po
sition, View v) {
87.                 showSelectedPresident(list.get(position));
88.             }
89.         });
90.     }
91.
92.     private void showRecyclerGrid() {
93.         rvCategory.setLayoutManager(new GridLayoutManager(this, 2));
94.         GridPresidentAdapter gridPresidentAdapter = new GridPresidentAd
apter(this);
95.         gridPresidentAdapter.setListPresident(list);
96.         rvCategory.setAdapter(gridPresidentAdapter);
97.
98.         ItemClickSupport.addTo(rvCategory).setOnItemClickListener(new I
temClickSupport.OnItemClickListener() {
99.             @Override
100.             public void onItemClick(RecyclerView recyclerView,
int position, View v) {

```

```
101.         showSelectedPresident(list.get(position));
102.     }
103. });
104. }
105.
106.     private void showRecyclerCardView() {
107.         rvCategory.setLayoutManager(new LinearLayoutManager(this)
108. );
109.         CardViewPresidentAdapter cardViewPresidentAdapter = new C
ardViewPresidentAdapter(this);
110.         cardViewPresidentAdapter.setListPresident(list);
111.         rvCategory.setAdapter(cardViewPresidentAdapter);
112.     }
113.     @Override
114.     public boolean onCreateOptionsMenu(Menu menu) {
115.         getMenuInflater().inflate(R.menu.menu_main, menu);
116.         return super.onCreateOptionsMenu(menu);
117.     }
118.
119.     private void setActionBarTitle(String title) {
120.         getSupportActionBar().setTitle(title);
121.     }
122.
123.     @Override
124.     public boolean onOptionsItemSelected(MenuItem item) {
125.
126.         setMode(item.getItemId());
127.
128.         return super.onOptionsItemSelected(item);
129.     }
130.
131.     public void setMode(int selectedMode) {
132.         String title = null;
133.         switch (selectedMode) {
134.             case R.id.action_list:
135.                 title = "Mode List";
136.                 showRecyclerList();
137.                 break;
138.
139.             case R.id.action_grid:
140.                 title = "Mode Grid";
141.                 showRecyclerGrid();
142.                 break;
143.
144.             case R.id.action_cardview:
145.                 title = "Mode CardView";
146.                 showRecyclerCardView();
147.                 break;
148.         }
149.         mode = selectedMode;
150.         setActionBarTitle(title);
151.     }
152.
153.     @Override
154.     protected void onSaveInstanceState(Bundle outState) {
155.         super.onSaveInstanceState(outState);
156.         outState.putString(STATE_TITLE, getSupportActionBar().get
Title().toString());
```

```
157.         outState.putParcelableArrayList(STATE_LIST, list);
158.         outState.putInt(STATE_MODE, mode);
159.     }
160. }
```

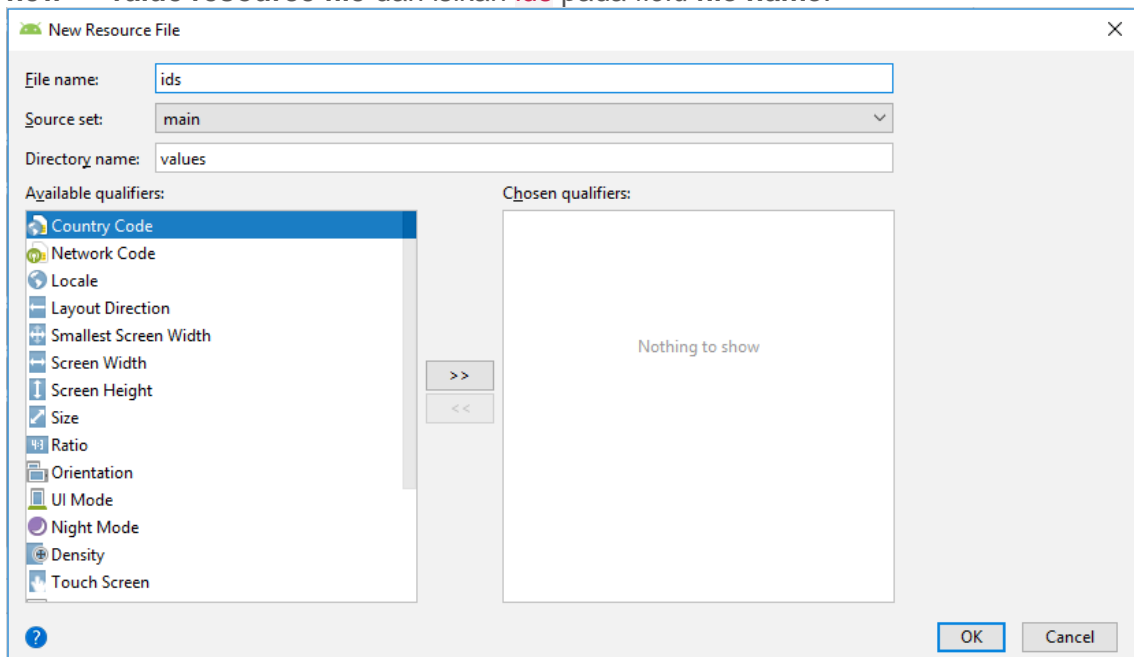
Maka setelah dijalankan akan jadi seperti ini :



Modul 11.4 – RecyclerView: ItemClickListener

Listview memiliki listener untuk melakukan sebuah aksi ketika salah satu item pada list dipilih. Sementara itu, pada recyclerview kita harus membuatnya secara manual. Pada modul ini, kita akan mengupas bagaimana caranya menangani kejadian ketika salah satu item pada list di RecyclerView dipilih.

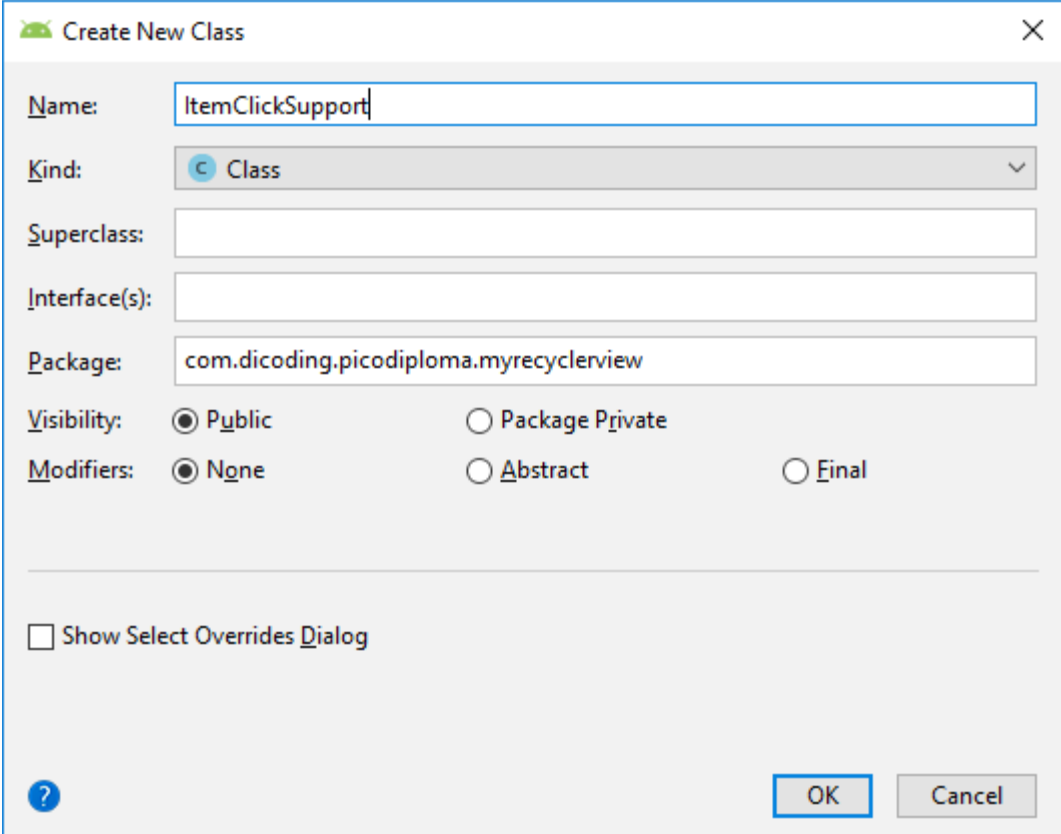
1. Pertama buat berkas bernama **ids** pada direktori values. Klik kanan pada **values** → **new** → **value resource file** dan isikan **ids** pada **field file name**.



Setelah terbentuk, lengkapi kodenya menjadi seperti berikut :

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <resources>
3.     <item name="item_click_support" type="id" />
4. </resources>
```

2. Selanjutnya buat kelas baru dengan nama **ItemClickSupport**.



Create New Class

Name: ItemClickSupport

Kind: Class

Superclass:

Interface(s):

Package: com.dicoding.picodiploma.myrecyclerview

Visibility: ☒ Public ☐ Package Private

Modifiers: ☒ None ☐ Abstract ☐ Final

☐ Show Select Overrides Dialog

OK Cancel

Setelah terbentuk, lengkapi kodenya menjadi seperti berikut :

```
1. public class ItemClickSupport {
2.     private final RecyclerView mRecyclerView;
3.     private OnItemClickListener mOnItemClickListener;
4.     private OnItemLongClickListener mOnItemLongClickListener;
5.     private View.OnClickListener mOnClickListener = new View.On
ClickListener() {
6.         @Override
7.         public void onClick(View v) {
8.             if (mOnItemClickListener != null) {
9.                 RecyclerView.ViewHolder holder = mRecyclerView.
getChildViewHolder(v);
10.                 mOnItemClickListener.onItemClicked(mRecycle
rView, holder.getAdapterPosition(), v);
11.             }
12.         }
13.     };
14.     private View.OnLongClickListener mOnLongClickListener =
new View.OnLongClickListener() {
15.         @Override
16.         public boolean onLongClick(View v) {
17.             if (mOnItemLongClickListener != null) {
18.                 RecyclerView.ViewHolder holder = mRecycleV
iew.getChildViewHolder(v);
19.                 return mOnItemLongClickListener.onItemLongC
licked(mRecyclerView, holder.getAdapterPosition(), v);
20.             }
21.             return false;
22.         }
23.     };
```

```
24.         private RecyclerView.OnChildAttachStateChangeListener m
AttachListener
25.             = new RecyclerView.OnChildAttachStateChangeList
ener() {
26.                 @Override
27.                 public void onChildViewAttachedToWindow(@NonNull Vi
ew view) {
28.                     if (mOnItemClickListener != null) {
29.                         view.setOnClickListener(mOnClickListener);
30.                     }
31.                     if (mOnItemLongClickListener != null) {
32.                         view.setOnLongClickListener(mOnLongClickLis
tener);
33.                     }
34.                 }
35.
36.                 @Override
37.                 public void onChildViewDetachedFromWindow(@NonNull
View view) {
38.                 }
39.             };
40.
41.         private ItemClickSupport(RecyclerView recyclerView) {
42.             mRecyclerView = recyclerView;
43.             mRecyclerView.setTag(R.id.item_click_support, this)
;
44.             mRecyclerView.addOnChildAttachStateChangeListener(m
AttachListener);
45.         }
46.
47.         public static ItemClickSupport addTo(RecyclerView view)
{
48.             ItemClickSupport support = (ItemClickSupport) view.
getTag(R.id.item_click_support);
49.             if (support == null) {
50.                 support = new ItemClickSupport(view);
51.             }
52.             return support;
53.         }
54.
55.         public static ItemClickSupport removeFrom(RecyclerView
view) {
56.             ItemClickSupport support = (ItemClickSupport) view.
getTag(R.id.item_click_support);
57.             if (support != null) {
58.                 support.detach(view);
59.             }
60.             return support;
61.         }
62.
63.         public void setOnItemClickListener(OnItemClickListener
listener) {
64.             mOnItemClickListener = listener;
65.         }
66.
67.
68.         public void setOnItemLongClickListener(OnItemLongClickL
istener listener) {
69.             mOnItemLongClickListener = listener;
70.         }
71.
```



```

72.     private void detach(RecyclerView view) {
73.         view.removeOnChildAttachStateChangeListener(mAttach
74.             Listener);
75.         view.setTag(R.id.item_click_support, null);
76.     }
77.     public interface OnItemClickListener {
78.         void onItemClick(RecyclerView recyclerView, int p
79.             osition, View v);
80.     }
81.     public interface OnItemLongClickListener {
82.         boolean onItemLongClicked(RecyclerView recyclerView
83.             , int position, View v);
84.     }

```

Pada prinsipnya, kelas `ItemClickSupport` akan menangani kejadian ketika pengguna menekan (on clicked) atau melakukan klik yang lama (*long clicked*) pada salah satu item pada `RecyclerView`.

- Setelah selesai, mari kita pasang pada `MainActivity`. Tambahkan satu metode ini untuk menandakan item mana yang dipilih.

```

1. private void showSelectedPresident(President president){
2.     Toast.makeText(this, "Kamu memilih "+president.getName(), T
3.         oast.LENGTH_SHORT).show();

```

Kemudian tambahkan pada masing-masing metode di *list* dan *grid* seperti berikut :

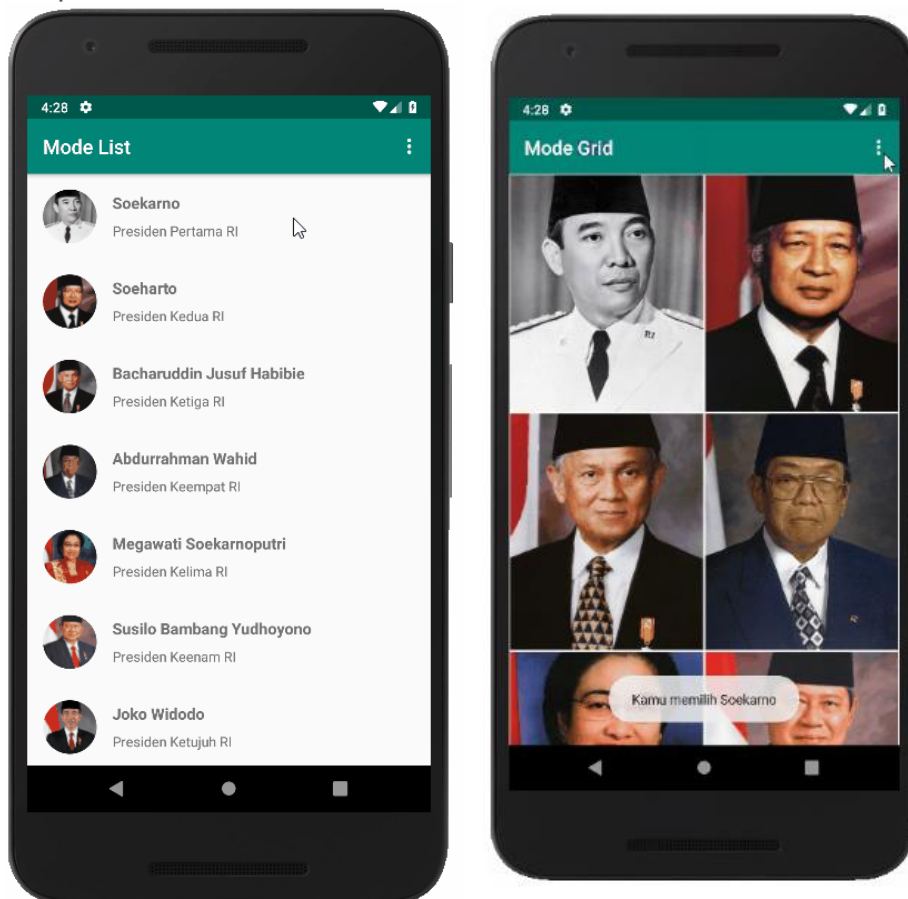
```

4. private void showRecyclerViewList(){
5.     rvCategory.setLayoutManager(new LinearLayoutManager(this));
6.     ListPresidentAdapter listPresidentAdapter = new ListPreside
7.         ntAdapter(this);
8.     listPresidentAdapter.setListPresident(list);
9.     rvCategory.setAdapter(listPresidentAdapter);
10.    ItemClickSupport.addTo(rvCategory).setOnItemClickListener(new ItemClickSupport.OnItemClickListener() {
11.        @Override
12.        public void onItemClick(RecyclerView recyclerView
13.            , int position, View v) {
14.            showSelectedPresident(list.get(position));
15.        }
16.    });
17.
18.    private void showRecyclerViewGrid(){
19.        rvCategory.setLayoutManager(new GridLayoutManager(this,
20.            2));

```

```
20.     GridPresidentAdapter gridPresidentAdapter = new GridPre
    sidentAdapter(this);
21.     gridPresidentAdapter.setListPresident(list);
22.     rvCategory.setAdapter(gridPresidentAdapter);
23.
24.     ItemClickSupport.addTo(rvCategory).setOnItemClickListener(new ItemClickSupport.OnItemClickListener() {
25.         @Override
26.         public void onItemClick(RecyclerView recyclerView
    , int position, View v) {
27.             showSelectedPresident(list.get(position));
28.         }
29.     });
30. }
```

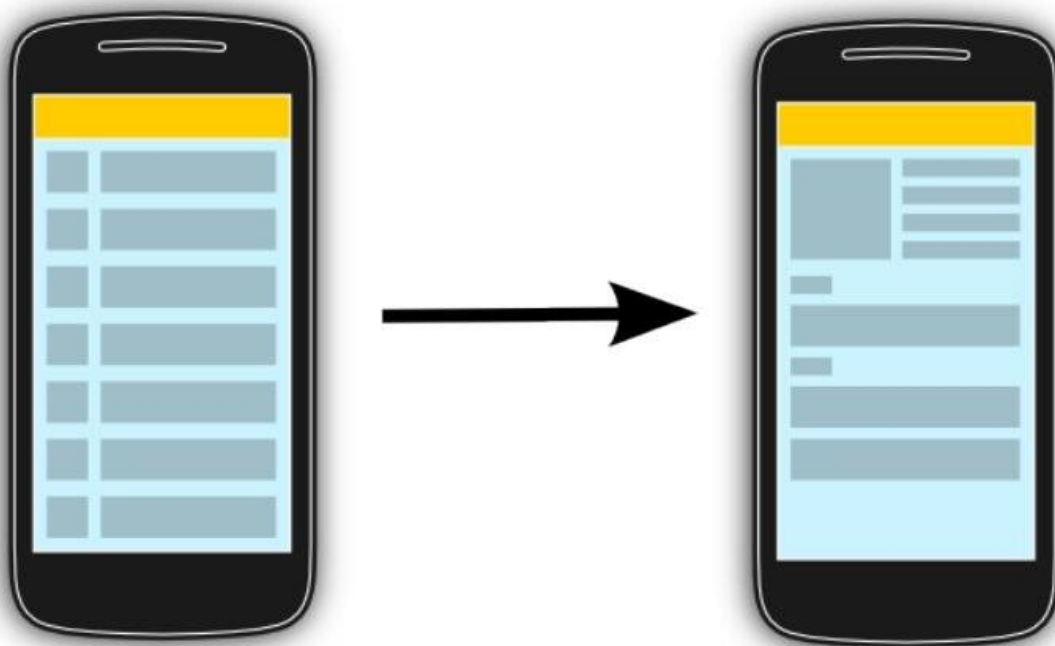
4. Sekarang jalankan kembali aplikasinya. Klik salah satu item pada **RecyclerView** bentuk *list* dan *grid*. Seharusnya sebuah obyek **Toast** akan tampil ketika salah satu item ditekan.



Selesai! Sejauh ini Anda telah berhasil menerapkan beragam bentuk dari **RecyclerView**. Pendekatan ini sangat berguna karena ke depannya Anda akan berhadapan dengan koleksi data yang harus ditampilkan ke dalam bentuk *list*.

Bedah Kode

Pendekatan umum dalam mengembangkan aplikasi *mobile* terletak pada bentuk *list-to-detail*. Pendekatan ini menampilkan sejumlah informasi dalam bentuk *list*. Ketika pengguna menekan salah satu *item* untuk mendapatkan detail informasi, maka aplikasi akan menampilkannya secara detail.



Pendekatan tersebut sangat lumrah dikarenakan keterbatasan ukuran layar. Keterbatasan ini justru memudahkan penggunaan dua jempol sebagai pointer untuk melakukan sebuah aksi pada aplikasi.

Android sendiri memiliki beberapa mekanisme menampilkan data dalam jumlah yang banyak, yaitu dalam bentuk **ListView**, **GridView**, maupun **RecyclerView**. Namun semenjak adanya material design, para developer dituntut untuk memahami suksesor dari komponen **ListView** ini.

Kelebihan utama dari **RecyclerView** terletak pada performa dan fleksibilitasnya.

Dari sisi performa, secara default adapter dari **RecyclerView** haruslah mengimplementasikan **ViewHolder** pattern. Tujuan dari pendekatan ini adalah agar **RecyclerView** tetap responsif dengan mempertahankan performa terbaik ketika menampilkan koleksi data dalam jumlah yang banyak.

Ini jelas berbeda dengan **ListView** yang tidak mewajibkan pattern tersebut secara *default*. Pada salah satu contoh adapter yang kita tuliskan di atas, pattern **ViewHolder** diimplementasikan pada bagian berikut :

```
1. @NonNull
2. @Override
3. public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
    int viewType) {
4.     View itemRow = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_row_president, parent, false);
5.     return new CategoryViewHolder(itemRow);
6. }
7.
8. @Override
9. public void onBindViewHolder(@NonNull CategoryViewHolder holder, int position) {
10.    holder.tvName.setText(getListPresident().get(position).getName());
11.    holder.tvRemarks.setText(getListPresident().get(position).getRemarks());
12.
13.    Glide.with(context)
14.        .load(getListPresident().get(position).getPhoto())
15.        .apply(new RequestOptions().override(55, 55))
16.        .into(holder.imgPhoto);
17. }
18.
19. class CategoryViewHolder extends RecyclerView.ViewHolder{
20.     TextView tvName;
21.     TextView tvRemarks;
22.     ImageView imgPhoto;
23.
24.     public CategoryViewHolder(View itemView) {
25.         super(itemView);
26.         tvName = itemView.findViewById(R.id.tv_item_name);
27.         tvRemarks = itemView.findViewById(R.id.tv_item_remarks);
28.         imgPhoto = itemView.findViewById(R.id.img_item_photo);
29.     }
30. }
```

Setiap kali kita melakukan *scroll* pada **RecyclerView**, ia akan memeriksa memori apakah *item* view yang hendak ditampilkan tertentu sudah berada di memori atau belum. Jika belum, maka akan dijalankan sebuah proses yang cukup mahal dari segi memori, yaitu dijalanannya **onCreateViewHolder()**. Di dalam metode ini, terjadi sebuah *casting view* (**findViewById**) yang akan menampilkan koleksi data dalam format tampilan yang ditentukan, baris per baris jika pada bentuk *list* atau baris dan kolom pada bentuk *grid*.

Hubungan antara satu adapter dengan `ViewHolder` adalah satu ke banyak. Artinya, satu kelas adapter bisa memiliki lebih dari satu `ViewHolder`.

```
1. rvCategory.setHasFixedSize(true);
```

Baris di atas menjelaskan bahwa bila *fixed size* bernilai true, maka `RecyclerView` dapat melakukan optimasi ukuran lebar dan tinggi secara otomatis. Nilai lebar dan tinggi `RecyclerView` menjadi konstan. Terlebih jika kita memiliki koleksi data yang dinamis untuk proses penambahan, perpindahan, dan pengurangan *item* dari koleksi data.

Pada contoh kode di atas, kita menggunakan library bernama Glide untuk menampilkan foto secara *asynchronous* dari URL yang diberikan.

Pada sisi fleksibilitas, `RecyclerView` memiliki beragam bentuk yang disesuaikan dengan design yang diinginkan. Kita hanya perlu menentukan nilai pada metode `setLayoutManager()` saja untuk menentukan bagaimana `RecyclerView` ditampilkan.

```
1. rvCategory.setLayoutManager(new LinearLayoutManager(this));
```

atau

```
1. rvCategory.setLayoutManager(new GridLayoutManager(this, 2));
```

Lembar Kerja Praktikum

NPM:	Nama Asisten:
Nama:	Nilai:
Kelas:	

Form ini digunakan untuk menilai kerja mahasiswa selama proses praktikum berlangsung. Selama praktikum mahasiswa diminta untuk membuat Aplikasi Modul 11.1 – 11.4. Semua tombol harus dapat berfungsi sesuai dengan materi yang telah disampaikan.