

BAB VII

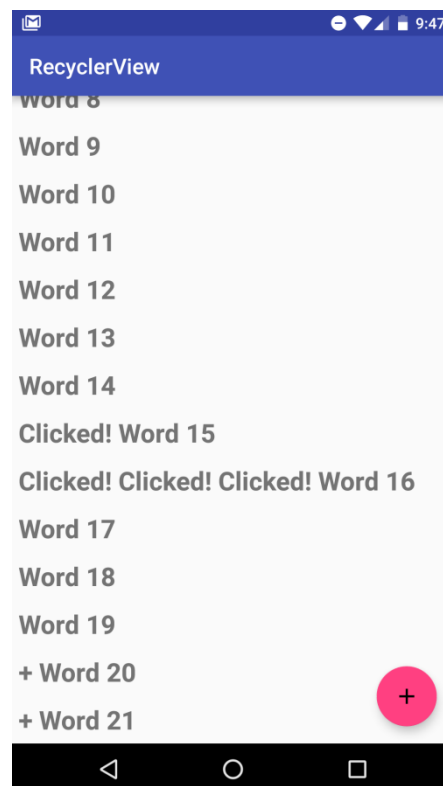
RecyclerView

A. TINJAUAN TEORI

7.1. Apa itu *RecyclerView* ?

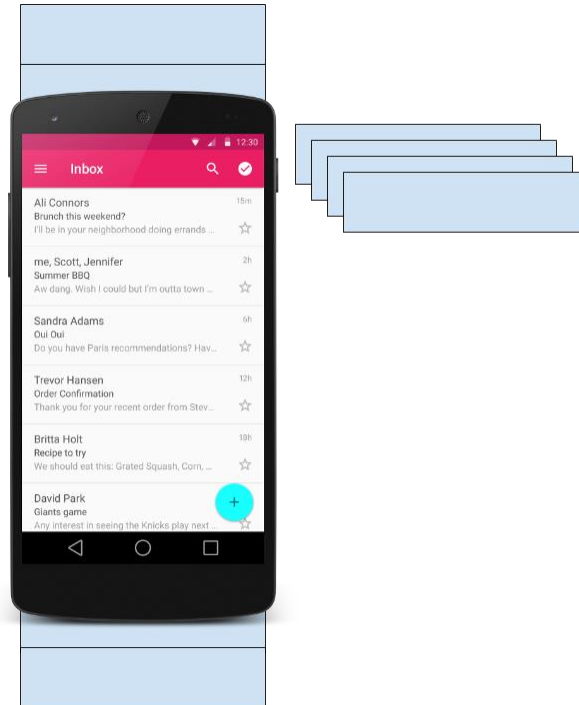
Terkadang dalam sebuah aplikasi kita ingin menampilkan sebuah set data yang berjumlah besar (ratusan atau mungkin sampai jutaan). Nah disini kita tentu perlu sebuah *view* yang mampu handle itu. Adapun sebelum *RecyclerView* ada namanya *ListView*. Namun ada beberapa kekurangan yang ada pada *ListView*. Disini muncullah *RecyclerView* dengan kemampuan yang lebih baik dari *ListView* (lebih cepat dan lebih efisien, terutama dalam menangani data berjumlah besar). Adapun contoh penggunaan *RecyclerView* ada pada GMail.

Saat Anda menampilkan banyak item dalam daftar yang bisa digulir, sebagian besar item tidak terlihat. Misalnya, dalam daftar kata yang panjang atau banyak judul berita, pengguna hanya melihat sedikit item daftar untuk setiap kalinya.



Atau, Anda bisa memiliki kumpulan data yang akan berubah saat pengguna berinteraksi dengannya. Jika Anda membuat tampilan baru setiap kali data berubah, itu juga membuat banyak tampilan, bahkan untuk kumpulan data yang kecil. Dari perspektif kinerja, Anda bisa

meminimalkan jumlah tampilan yang disimpan pada titik tertentu (Memori), dan jumlah tampilan yang harus Anda buat (Waktu). Kedua tujuan ini bisa dicapai dengan membuat agak lebih banyak tampilan daripada yang bisa dilihat pengguna pada layar, dan buat cache serta gunakan kembali tampilan yang dibuat sebelumnya dengan data berbeda saat pengguna menggulir ke dalam dan ke luar tampilan.



Kelas *RecyclerView* adalah versi *ListView* yang lebih canggih dan fleksibel. Widget ini adalah kontainer untuk menampilkan rangkaian data besar yang bisa digulir secara sangat efisien dengan mempertahankan tampilan dalam jumlah terbatas. Gunakan widget *RecyclerView* bila Anda perlu menampilkan banyak data yang bisa digulir, atau kumpulan data dengan elemen yang berubah pada waktu proses berdasarkan aksi pengguna atau kejadian jaringan.

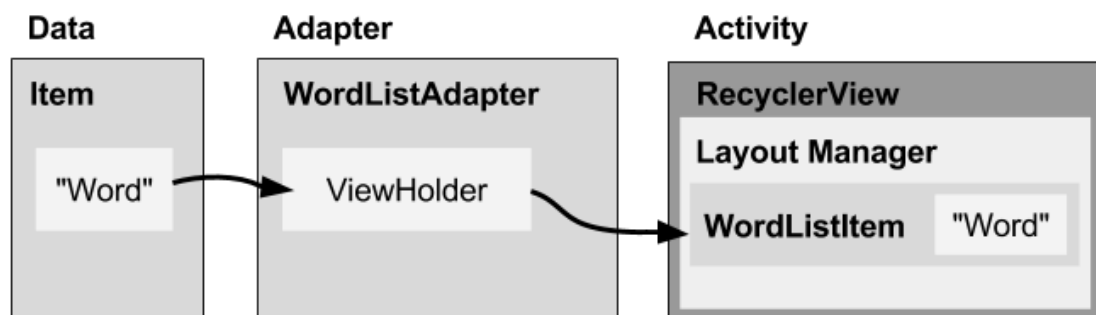
7.2. Komponen *RecyclerView*

Untuk menampilkan data dalam *RecyclerView*, Anda memerlukan bagian berikut:

- **Data.** Tidak penting dari mana asal data. Anda bisa membuat data secara lokal, seperti yang Anda lakukan dalam latihan, mendapatkannya dari database perangkat seperti yang akan Anda lakukan dalam praktik nanti, atau menariknya dari awan.
- **RecyclerView.** Daftar gulir yang berisi item daftar.
Instance *RecyclerView* sebagaimana didefinisikan dalam *file layout* aktivitas Anda akan bertindak sebagai kontainer tampilan.

- **Layout untuk satu item data.** Semua item daftar tampak sama, sehingga Anda bisa menggunakan layout yang sama untuk semuanya. *Layout item* harus dibuat secara terpisah dari *layout* aktivitas, sehingga satu per satu tampilan item bisa dibuat dan diisi data.
- **Pengelola layout.** Pengelola layout menangani penyusunan (*layout*) komponen antarmuka pengguna dalam suatu tampilan. Semua grup tampilan memiliki pengelola *layout*. Untuk *LinearLayout*, sistem Android menangani layout untuk Anda. *RecyclerView* memerlukan pengelola layout eksplisit untuk mengelola susunan item daftar yang terdapat di dalamnya. Layout ini bisa vertikal, horizontal, atau berupa petak.
Pengelola layout adalah *instance* dari ***RecyclerView.LayoutManager*** untuk menyusun layout item dalam *RecyclerView*.
- **Adapter.** Adapter menghubungkan data Anda dengan *RecyclerView*. Adapter menyiapkan data dan cara menampilkan data dalam *view holder*. Bila data berubah, adapter akan memperbarui materi tampilan item daftar terkait dalam *RecyclerView*.
Adapter juga merupakan ekstensi dari ***RecyclerView.Adapter***. Adapter menggunakan *ViewHolder* untuk menampung tampilan yang menyusun setiap item dalam *RecyclerView*, dan mengikat data untuk ditampilkan dalam tampilan yang menampilkannya.
- **View holder.** View holder memperluas kelas *ViewHolder*. View holder berisi tampilan informasi untuk menampilkan satu item dari layout item.
View holder digunakan oleh adapter untuk menyediakan data, yang merupakan ekstensi dari ***RecyclerView.ViewHolder***.

Diagram di bawah ini menampilkan hubungan antara komponen-komponen ini.



7.3. Implementasi RecyclerView

Langkah-langkah mengimplementasikan recyclerview sebagai berikut:

1. Tambahkan dependencies komponen *recyclerview* pada file ***build.gradle (modul: app)***.
2. Tambahkan obyek *RecyclerView* di berkas layout xml dari activity / fragment.
3. Definisikan kelas model yang akan digunakan sebagai data *source*.
4. Buat berkas layout xml untuk 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***.

B. TUJUAN

Pada pertemuan ini, tujuan yang harus dicapai antara lain

1. Mahasiswa memahami cara menggunakan RecyclerView
2. Mahasiswa mampu membuat tampilan RecyclerView seperti list, grid, dan card.
3. Mahasiswa mampu membuat onClick pada RecyclerView
4. Mahasiswa mampu menambahkan menu pada ActionBar

C. PERALATAN YANG DIGUNAKAN

Pengembangan aplikasi android membutuhkan peralatan baik perangkat keras maupun perangkat lunak. Adapun perangkat keras yang digunakan untuk membangun aplikasi android adalah sebagai berikut.

1. Laptop / Personal Komputer dengan spesifikasi minimal seperti berikut.
 - ✓ Prosesor minimal Intel Core i3 Generasi kelima, atau prosesor AMD A8
 - ✓ RAM minimum 3 GB, RAM yang disarankan 8 GB,
 - ✓ Ruang disk minimum yang tersedia 2 GB,
 - ✓ Resolusi layar minimum 1280 x 800
2. *Smartphone* android yang memiliki versi minimal Ice Cream Sandwich (Android 4.0)
3. Kabel data

Sedangkan perangkat lunak yang harus terpasang di Laptop/PC antara lain

1. Sistem operasi windows (disarankan windows 10), Linux (disarankan Linux Mint), Macintosh (disarankan versi 10, yosemite)
2. Java Development Kit (JDK) 8 atau versi terbaru
3. Android Studio

D. LANGKAH PRAKTIKUM

7.4. Membuat project RecyclerView di android

1. Buat sebuah project baru di android studi, seperti berikut

Nama Project	MyRecyclerView
--------------	-----------------------

Target & Minimum Target SDK	Phone and Table, API level 21
Tipe Activity	Empty Activity
Activity Name	MainActivity
Use AndroidX Artifacts	True

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

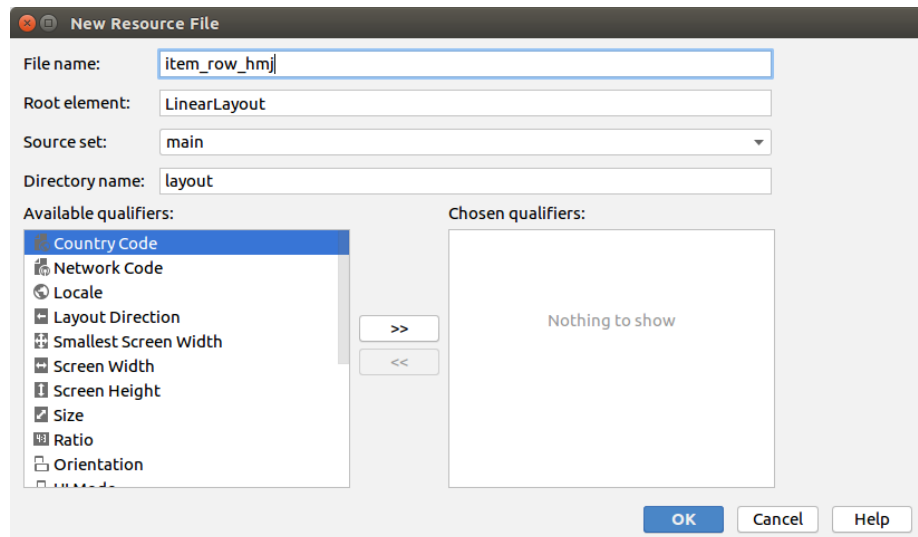
- implementation 'androidx.recyclerview:recyclerview:1.0.0'
- implementation 'de.hdodenhof:circleimageview:3.0.0'
- implementation 'com.github.bumptech.glide:glide:4.9.0'

- Selanjutnya pada **activity_main** tambahkan kode berikut.

- <?xml version="1.0" encoding="utf-8"?>
- <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
- xmlns:app="http://schemas.android.com/apk/res-auto"
- android:id="@+id/activity_main"
- xmlns:tools="http://schemas.android.com/tools"
- android:layout_width="match_parent"
- android:layout_height="match_parent"
- tools:context=".MainActivity">
-
- <androidx.recyclerview.widget.RecyclerView
- android:id="@+id/rv_hmj"
- android:layout_width="match_parent"
- android:layout_height="match_parent"
- tools:listitem="@layout/item_row_hmj">
- </androidx.recyclerview.widget.RecyclerView>
-
- </RelativeLayout>

Akan ada tanda merah di **@layout/item_row_hmj**. Hal ini dikarenakan layout **item_row_hmj** belum ditambahkan.

- 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_hmj**.



5. 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/vertical_activity">
8.
9.     <de.hdodenhof.circleimageview.CircleImageView
10.         android:id="@+id/img_item_photo"
11.         android:layout_width="55dp"
12.         android:layout_height="55dp"
13.         android:layout_marginEnd="@dimen/vertical_activity"
14.         android:layout_marginRight="@dimen/horizontal_ativity"
15.         tools:src="@android:color/darker_gray"/>
16.
17.     <LinearLayout
18.         android:layout_width="match_parent"
19.         android:layout_height="wrap_content"
20.         android:layout_centerVertical="true"
21.         android:layout_toEndOf="@+id/img_item_photo"
22.         android:layout_toRightOf="@+id/img_item_photo"
23.         android:orientation="vertical">
24.
25.         <TextView
26.             android:id="@+id/tv_item_name"
27.             android:layout_width="match_parent"
28.             android:layout_height="wrap_content"
29.             android:layout_marginBottom="8dp"
30.             android:textSize="16sp"
31.             android:textStyle="bold"
32.             tools:text="@string/hmj_name"/>
33.
34.         <TextView
35.             android:id="@+id/tv_item_desc"
36.             android:layout_width="match_parent"
37.             android:layout_height="wrap_content"

```

```

38.         android:ellipsize="end"
39.         android:maxLines="2"
40.         tools:text="@string/description" />
41.
42.     </LinearLayout>
43.
44. </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.

6. Terdapat error pada `item_row_hmj.xml`, hal itu dikarenakan kita belum menambahkan data string. Oleh karena itu silahkan buka `stings.xml` dan lengkapi dengan kode berikut ini.

```

1. <resources>
2.     <string name="app_name">MyRecyclerView</string>
3.     <string name="hmj_name">HMJ Polines</string>
4.     <string name="description">Deskripsi</string>
5.
6.     <string-array name="data_name">
7.         <item>HMJ Elektro</item>
8.         <item>HMJ Sipil</item>
9.         <item>HMJ Mesin</item>
10.        <item>HMJ Akutansi</item>
11.        <item>HMJ Administrasi Bisnis</item>
12.        <item>UKM PCC</item>
13.        <item>UKM Konsep</item>
14.    </string-array>
15.
16.    <string-array name="data_description">
17.        <item>Himpunan Mahasiswa Jurusan Teknik Elektro adalah salah satu Himpunan Mahasiswa Jurusan yang ada di Politeknik Negeri Semarang yang merupakan organisasi kemahasiswaan di tingkat jurusan, yang dibentuk oleh mahasiswa Jurusan Teknik Elektro untuk melaksanakan kegiatan Tri Dharma Perguruan Tinggi.</item>
18.        <item>Himpunan Mahasiswa Jurusan Teknik Sipil adalah salah satu Himpunan Mahasiswa Jurusan yang ada di Politeknik Negeri Semarang yang merupakan organisasi kemahasiswaan di tingkat jurusan, yang dibentuk oleh mahasiswa Jurusan Teknik Sipil untuk melaksanakan kegiatan Tri Dharma Perguruan Tinggi.</item>
19.        <item>Himpunan Mahasiswa Jurusan Teknik Mesin adalah salah satu Himpunan Mahasiswa Jurusan yang ada di Politeknik Negeri Semarang yang merupakan organisasi kemahasiswaan di tingkat jurusan, yang dibentuk oleh mahasiswa Jurusan Teknik Mesin untuk melaksanakan kegiatan Tri Dharma Perguruan Tinggi.</item>
20.        <item>Himpunan Mahasiswa Jurusan Akutansi adalah salah satu Himpunan Mahasiswa Jurusan yang ada di Politeknik Negeri Semarang yang merupakan organisasi kemahasiswaan di tingkat jurusan, yang dibentuk oleh mahasiswa Jurusan akutansi untuk melaksanakan kegiatan Tri Dharma Perguruan Tinggi.</item>
21.        <item>Himpunan Mahasiswa Jurusan Administrasi Bisnis adalah salah satu Himpunan Mahasiswa Jurusan yang ada di Politeknik Negeri Semarang yang merupakan organisasi kemahasiswaan di tingkat jurusan, yang dibentuk oleh mahasiswa Jurusan Administrasi Bisnis untuk melaksanakan kegiatan Tri Dharma Perguruan Tinggi.</item>
22.        <item>Unit Kegiatan Mahasiswa Polytechnic Computer Club atau biasa disingkat UKM PCC, merupakan organisasi di Politeknik Negeri Semarang yang bergerak di bidang informasi dan teknologi. UKM Polytechnic Computer Club secara eksplisit terbagi menjadi dua bagian inti, yaitu Workshop dan Organisasi.</item>
23.        <item>UKM Komunitas Seni Polines merupakan satu-satunya unit kegiatan mahasiswa di Politeknik Negeri Semarang yang bergerak di bidang Kesenian. Nama UKM Komunitas Seni Polines diresmikan pada tanggal 21 Juni 2003. Fungsi UKM KoNSEP adalah sebagai satu-

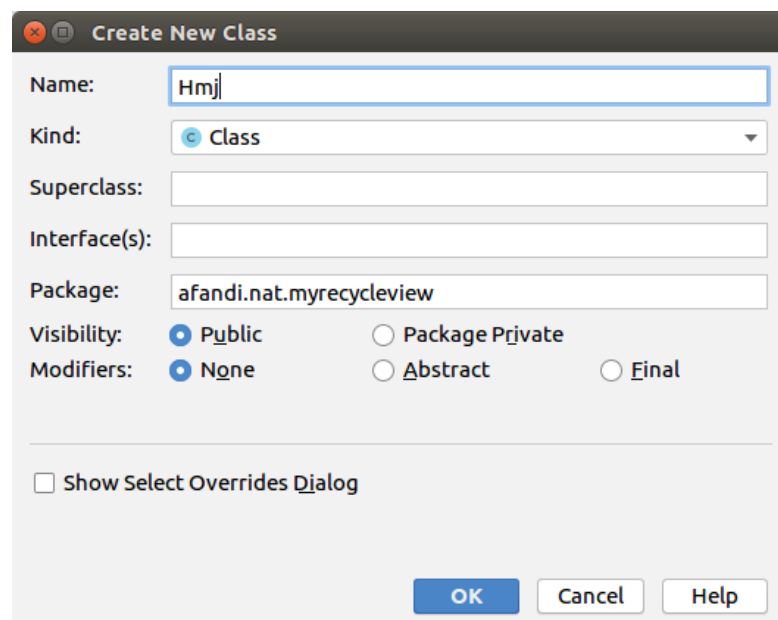
```

```

    satunya sarana pembangunan dan penyaluran minat dan bakat Mahasiswa Polines dalam
    m bidang seni dan berorganisasi.</item>
24. </string-array>
25.
26. <string-array name="photo">
27.     <item>https://i.postimg.cc/QxdcLRBk/hme.jpg</item>
28.     <item>https://i.postimg.cc/yxBXSxYw/hms.jpg</item>
29.     <item>https://i.postimg.cc/LsL1BS7R/hmm.jpg</item>
30.     <item>https://i.postimg.cc/3NS4Jxyj/hmak.jpg</item>
31.     <item>https://i.postimg.cc/pLYvD1Z4/hmab.jpg</item>
32.     <item>https://i.postimg.cc/8PGrdsgC/pcc.png</item>
33.     <item>https://i.postimg.cc/QC3VqYPz/konsep.png</item>
34. </string-array>
35. </resources>

```

7. Kemudian buat sebuah kelas model data dengan nama **Hmj**.



Setelah itu kita bisa menambahkan kode menjadi seperti berikut:

```

1. public class Hmj {
2.     private String name;
3.     private String description;
4.     private String photo;
5.
6.     public String getName() {
7.         return name;
8.     }
9.
10.    public void setName(String name) {
11.        this.name = name;
12.    }
13.
14.    public String getDescription() {
15.        return description;
16.    }
17.
18.    public void setDescription(String description) {

```



```

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

```

Untuk kelas Java, ada cara otomatis untuk membuat setter & getter. Caranya, klik kanan pada text editor lalu pilih Generate dan pilih Getter and Setter. Pilih semua variabel. Maka setter and getter akan ter-generate secara otomatis.

Jangan lupa untuk menambahkan implementasi parcelable ke kelas Hmj. Penggunaan parcelable akan diterangkan dosen minggu depan.

```

1. public class Hmj implements Parcelable {
2.     private String name;
3.     private String description;
4.     private String photo;
5.
6.     public String getName() {
7.         return name;
8.     }
9.
10.    public void setName(String name) {
11.        this.name = name;
12.    }
13.
14.    public String getDescription() {
15.        return description;
16.    }
17.
18.    public void setDescription(String description) {
19.        this.description = description;
20.    }
21.
22.    public String getPhoto() {
23.        return photo;
24.    }
25.
26.    public void setPhoto(String photo) {
27.        this.photo = photo;
28.    }
29.
30.    @Override
31.    public int describeContents() {
32.        return 0;
33.    }
34.
35.    @Override
36.    public void writeToParcel(Parcel dest, int flags) {
37.        dest.writeString(this.name);

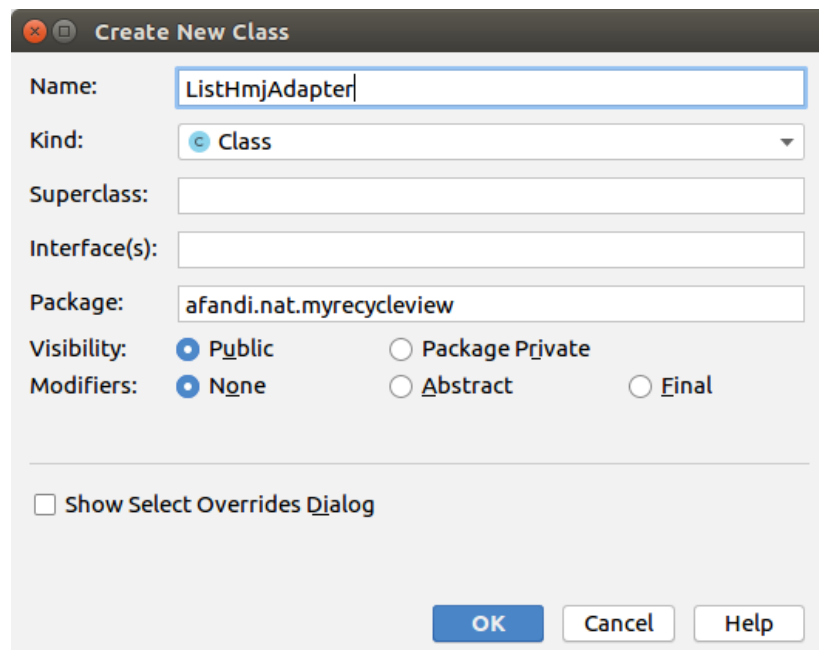
```

```

38.     dest.writeString(this.description);
39.     dest.writeString(this.photo);
40. }
41.
42. Hmj(){
43.
44. }
45.
46. private Hmj(Parcel in) {
47.     this.name = in.readString();
48.     this.description = in.readString();
49.     this.photo = in.readString();
50. }
51.
52. public static final Parcelable.Creator<Hmj> CREATOR = new Parcelable.Creator<Hmj>() {
53.     @Override
54.     public Hmj createFromParcel(Parcel source) {
55.         return new Hmj(source);
56.     }
57.
58.     @Override
59.     public Hmj[] newArray(int size) {
60.         return new Hmj[size];
61.     }
62. };
63. }

```

8. 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 **ListHmjAdapter**.



Setelah kelas adapter berhasil dibuat, lengkapi kodenya menjadi seperti berikut:

```

1.  public class ListHmjAdapter extends RecyclerView.Adapter<ListHmjAdapter.ListViewHolder>
    {
2.      private ArrayList<Hmj> listHmj;
3.
4.      public ListHmjAdapter(ArrayList<Hmj> list){
5.          this.listHmj = list;
6.      }
7.
8.      @NonNull
9.      @Override
10.     public ListViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType
    e) {
11.         View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.item_row_hmj,
    viewGroup, false);
12.         return new ListViewHolder(view);
13.     }
14.
15.     @Override
16.     public void onBindViewHolder(@NonNull ListViewHolder holder, int position) {
17.         Hmj hmj = listHmj.get(position);
18.
19.         Glide.with(holder.itemView.getContext())
20.             .load(hmj.getPhoto())
21.             .apply(new RequestOptions().override(55,55))
22.             .into(holder.imgPhoto);
23.
24.         holder.tvName.setText(hmj.getName());
25.         holder.tvDesc.setText(hmj.getDescription());
26.     }
27.
28.     @Override
29.     public int getItemCount() {
30.         return listHmj.size();
31.     }
32.
33.     class ListViewHolder extends RecyclerView.ViewHolder {
34.         ImageView imgPhoto;
35.         TextView tvName, tvDesc;
36.
37.         ListViewHolder(View itemView) {
38.             super(itemView);
39.             imgPhoto = itemView.findViewById(R.id.img_item_photo);
40.             tvName = itemView.findViewById(R.id.tv_item_name);
41.             tvDesc = itemView.findViewById(R.id.tv_item_desc);
42.         }
43.     }
44. }

```

Keterangan :

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 dijelankannya `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.

Glide, perhatikan kode berikut

```
1. Glide.with(itemView.context)
2.     .load(hero.photo)
3.     .apply(RequestOptions().override(55, 55))
4.     .into(img_item_photo)
```

Glide digunakan untuk memuat sebuah gambar, baik yang sudah Anda siapkan di drawable maupun berada di *server*. Glide terdiri dari banyak fungsi:

1. **with**: digunakan untuk memasukkan context.
 2. **load**: digunakan untuk memasukkan sumber gambar, contohnya menggunakan url.
 3. **apply**: digunakan untuk melakukan manipulasi gambar, contohnya menggunakan RequestOptions untuk me-override ukuran gambar.
 4. **into**: digunakan untuk memasukkan imageView sebagai obyek penampil gambar.
9. Kemudian modifikasi berkas kelas **MainActivity**, kita atur fixed size recyclerview yang sudah dibuat di **activity_main**

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

10. Selanjutnya setelah diinisiasikan, kita akan panggil data yang sudah kita buat di kelas **Hmj**.

```
1. public class MainActivity extends AppCompatActivity {
2.     private RecyclerView rvHmj;
3.     private ArrayList<Hmj> list = new ArrayList<>();
4.
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.
10.        if(getSupportActionBar() != null){
11.            getSupportActionBar().setTitle("HM & UKM Polines");
12.        }
13.    }
```

```

14.     list.addAll(getListHmj());
15.     showRecycleList();
16. }
17.
18. public ArrayList<Hmj> getListHmj() {
19.     String[] dataName = getResources().getStringArray(R.array.data_name);
20.     String[] dataDescription = getResources().getStringArray(R.array.data_description);
21.     String[] dataPhoto = getResources().getStringArray(R.array.photo);
22.
23.     ArrayList<Hmj> listHmj = new ArrayList<>();
24.     for (int i = 0; i < dataName.length; i++){
25.         Hmj hmj = new Hmj();
26.         hmj.setName(dataName[i]);
27.         hmj.setDescription(dataDescription[i]);
28.         hmj.setPhoto(dataPhoto[i]);
29.
30.         listHmj.add(hmj);
31.     }
32.     return listHmj;
33. }
34.
35. private void showRecycleList(){
36.     rvHmj.setLayoutManager(new LinearLayoutManager(this));
37.     ListHmjAdapter listHmjAdapter = new ListHmjAdapter(list);
38.     rvHmj.setAdapter(listHmjAdapter);
39. }
40. }

```

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

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

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

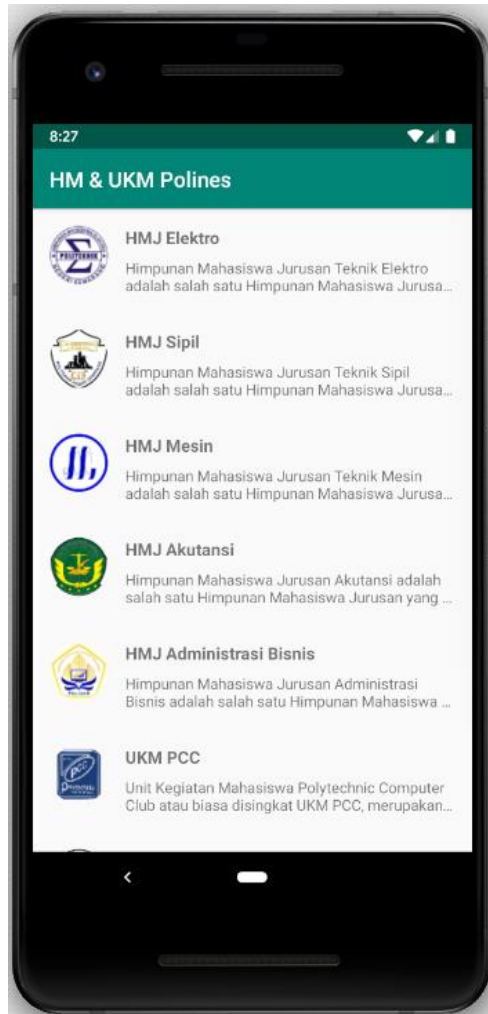
```

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

```

17. `</intent-filter>`
18. `</activity>`
19. `</application>`
- 20.
21. `</manifest>`

12. Jalankan aplikasi yang dibuat. Hasilnya kurang lebih akan seperti gambar dibawah ini



Challenge :

Buatlah RecyclerView seperti pada contoh sebelumnya, dan tambahkan data UKM yang ada di Polines. Upload project nya di elnino, pertemuan ke 8.