

**WORKSHOP MOBILE APPLICATIONS**  
**APLIKASI RETROVOLLEY**



**GOLONGAN B**

**Disusun oleh :**

Nama : HANI'ATUL KHOIRIYAH

Nim : E41201173

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI JEMBER**  
**2021**

## **KATA PENGANTAR**

Puji syukur penulis haturkan kehadiran Allah SWT atas rahmat-Nya penulis dapat menyelesaikan “Laporan Workshop Mobile Applications” tepat waktu. Penulis mengucapkan terima kasih kepada pihak yang telah membantu dan membimbing penulis dalam menyelesaikan laporan tersebut. Tugas yang telah diberikan ini dapat menambah pengetahuan dan wawasan terkait bidang yang ditekuni penulis.

Untuk mempermudah pemahaman juga sebagai bahan latihan, penulis menyertakan gambar dalam setiap langkah-langkah yang dapat dipraktikkan oleh para pembaca.

Penulis menyadari bahwa laporan tersebut masih jauh dari kata sempurna. Oleh karena itu penulis berharap akan kritik dan saran yang membangun dari segenap pembaca. Demikianlah semoga laporan yang telah dibuat dapat bermanfaat.

Jember, 17 November 2021

Penulis

Hani'Atul Khoiriyah

## DAFTAR ISI

KATA PENGANTAR .....	1
DAFTAR ISI.....	2
DAFTAR GAMBAR .....	3
BAB I .....	4
PENDAHULUAN .....	4
1.1.    Dasar Teori .....	4
1.2.    Tujuan dan Manfaat Praktikum .....	5
BAB II.....	6
PEMBAHASAN .....	6
2.1. Aplikasi Retrovolley.....	6
BAB III .....	19
PENUTUP.....	19
3.1. Kesimpulan.....	19
DAFTAR PUSTAKA .....	20

## DAFTAR GAMBAR

Gambar 1. Activity_main.xml .....	6
Gambar 2. activity_detail.xml.....	7
Gambar 3. activity_detail.xml.....	7
Gambar 4. Item.xml .....	8
Gambar 5. Item.xml .....	8
Gambar 6. Gradle .....	9
Gambar 7. Mainactivity.java.....	9
Gambar 8. Mainactivity.java.....	10
Gambar 9. DetailActivity.java .....	11
Gambar 10. movie.java .....	12
Gambar 11. MovieAdapter.java.....	13
Gambar 12. MovieAdapter.java.....	13
Gambar 13. movieholder.....	14
Gambar 14. VolleySingleton.java .....	15
Gambar 15. output.....	16
Gambar 16. output.....	17
Gambar 17. output.....	18

# **BAB I**

## **PENDAHULUAN**

### **1.1. Dasar Teori**

Retrofit dan Volley merupakan suatu library yang digunakan sebagai media komunikasi antara client (smartphone) dengan server berbasis web service. Perhatikan gambar berikut ini, kedua library seolah-olah menjadi penghubung antara smartphone dengan server. Smartphone bertugas melakukan request kepada server. Setelah itu, server akan memberikan response data sesuai yang diminta (request).

Retrofit merupakan library yang dibuat untuk memudahkan komunikasi berbasis web service pada Java dan Android. Library ini disebut juga sebagai REST (REpresentational State Transfer) Client pada Java dan Android. Komunikasi yang digunakan berupa pertukaran data berbentuk JSON (JavaScript Object Notation) atau XML (Extensible Markup Language). Retrofit menggunakan library OkHttp agar dapat melakukan request HTTP. Pada praktikum kali ini, format pertukaran data menggunakan JSON karena lebih cepat serta ringan. JSON juga lebih umum digunakan oleh para pengembang aplikasi baik berbasis web maupun mobile. Langkah pertama yaitu tambahkan permission INTERNET pada Manifest agar aplikasi dapat terhubung ke server meskipun server terletak pada localhost. Selanjutnya, tambahkan library retrofit dan Gson pada gradle nya. Gson digunakan untuk membaca data response dari server berupa API agar mudah digunakan. Selain itu, retrofit juga menggunakan JAVA 8 yang dapat diambil di official website android.

Volley adalah kerangka kerja http yang menawarkan penjadwalan permintaan, penanganan permintaan secara bersamaan, cache respons berbasis disk dan memori, prioritas permintaan, dan kemampuan pembatalan permintaan di luar kotak. Volley bisa get dan post jenis data apa pun menggunakan volley framework. Volley menawarkan penanganan respons bawaan untuk menangani string, gambar, dan respons JSON. Jika aplikasi

perlu menangani jenis respons selain string, gambar, dan JSON, maka harus dituliskan permintaan khusus dengan memperluas Request class dan menerapkan metode `parseNetworkResponse ()` dan metode `deliverResponse ()`. Volley bagus untuk mentransmisikan data aplikasi sederhana. Jika fitur aplikasi memerlukan unduhan besar, lebih baik menggunakan `DownloadManager` karena volley menyimpan cache untuk semua respons. Menambahkan library voli ke proyek aplikasi dengan cara memasukkan dependensi dalam file gradle seperti `"compile 'com.android.volley:volley:1.0.0'"`. Karena permintaan dan respons http dikirimkan melalui jaringan, aplikasi yang menggunakan volley memerlukan izin untuk mengakses internet. Anda perlu meminta izin `android.permission.INTERNET` dengan menambahkannya ke file manifest.

## **1.2. Tujuan dan Manfaat Praktikum**

- 1.2.1. Mahasiswa mampu memahami konsep retrofit.
- 1.2.2. Mahasiswa mampu menerapkan retrofit.
- 1.2.3. Mahasiswa mampu memahami konsep volley.
- 1.2.4. Mahasiswa mampu menerapkan volley.

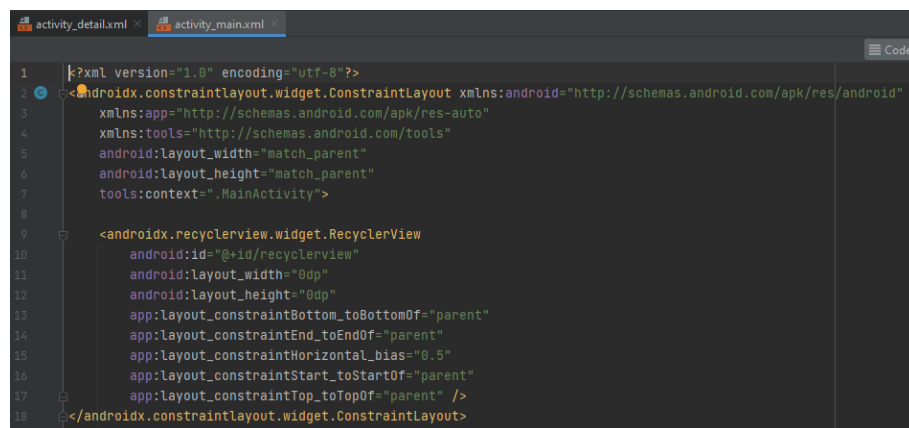
## BAB II

### PEMBAHASAN

#### 2.1. Aplikasi Retrovolley

Link GitHub : [https://github.com/haniatulkhoiriyah/HANI-ATUL-KHOIRIYAH\\_E41201173\\_B\\_SMT3](https://github.com/haniatulkhoiriyah/HANI-ATUL-KHOIRIYAH_E41201173_B_SMT3)

##### A. Activity\_main.xml

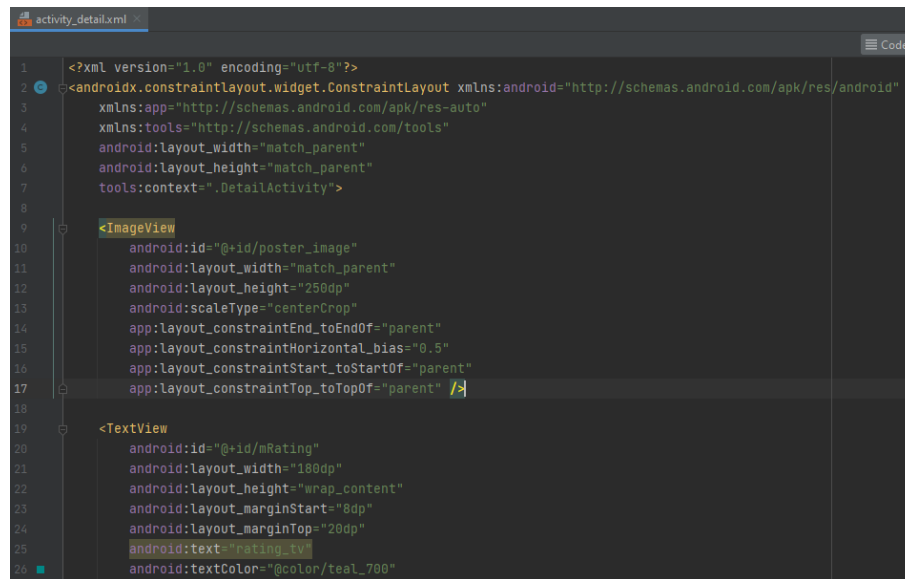


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <androidx.recyclerview.widget.RecyclerView
10         android:id="@+id/recyclerView"
11         android:layout_width="0dp"
12         android:layout_height="0dp"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintEnd_toEndOf="parent"
15         app:layout_constraintHorizontal_bias="0.5"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18 </androidx.constraintlayout.widget.ConstraintLayout>
```

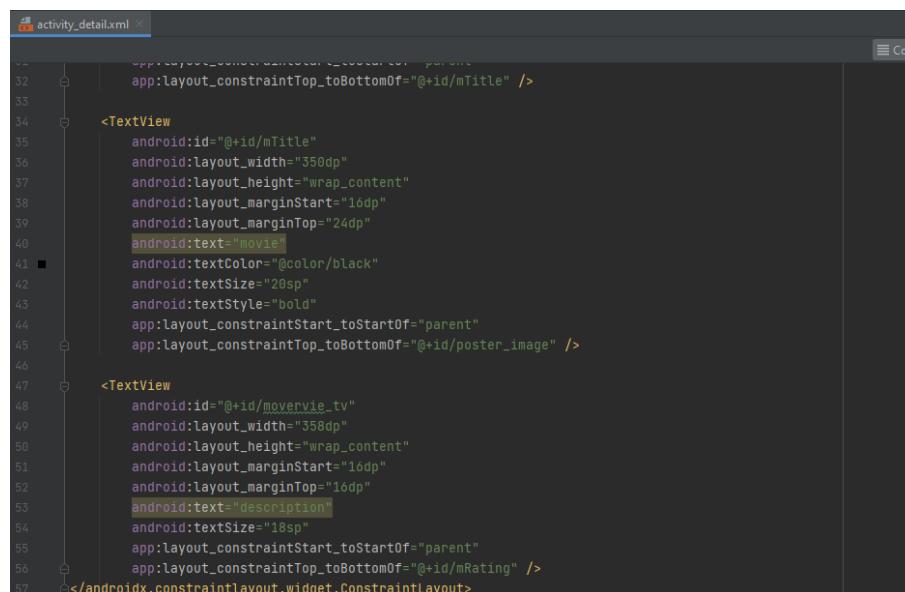
Gambar 1. Activity\_main.xml

Pada activity\_main menggunakan layout berupa constrainlayout karena constraintlayout adalah layout yang tergolong cukup mudah untuk mengaturnya. Kemudian membuat recyclerview untuk menampilkan pilihan – pilihan untuk user supaya dapat memilih pilihan movie yang ada.

##### B. Activity\_detail.xml



Gambar 2. activity\_detail.xml

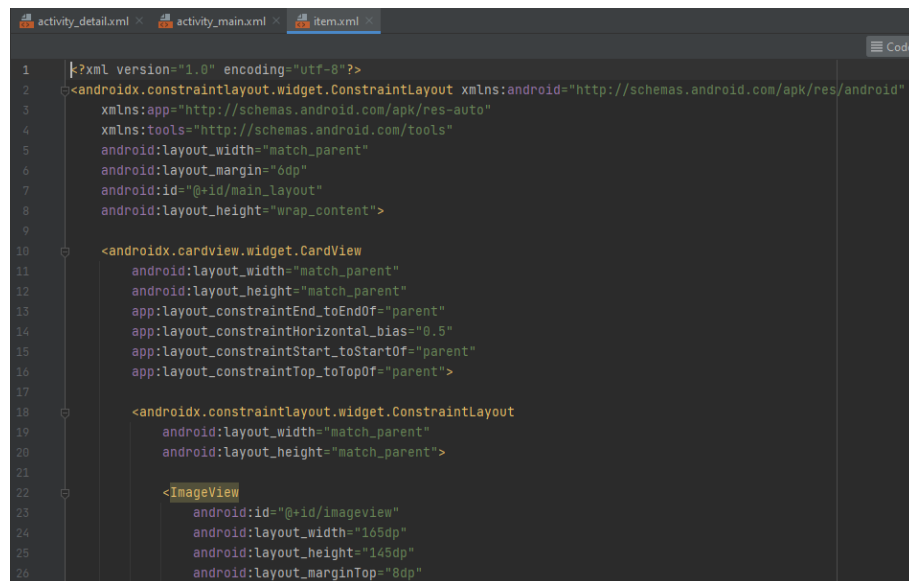


Gambar 3. activity\_detail.xml

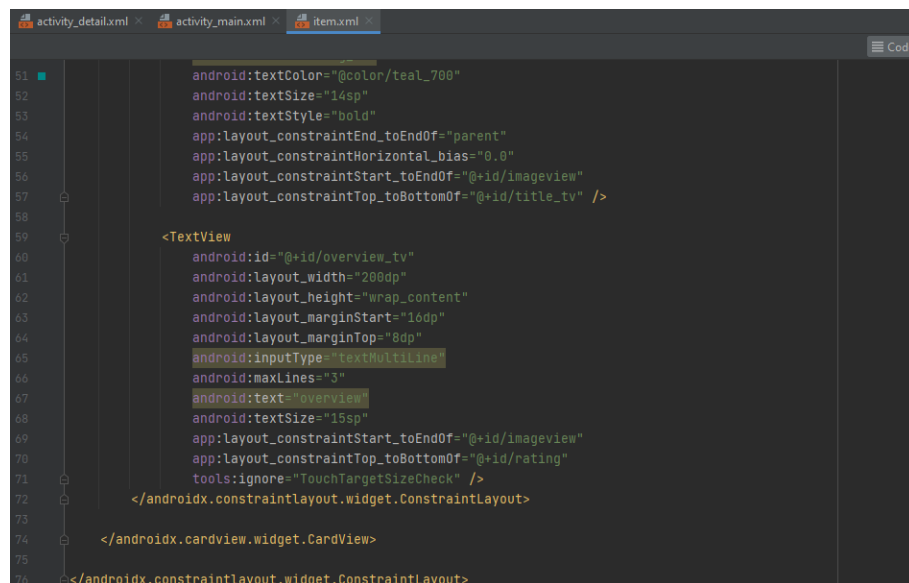
Pada activity\_detail.xml digunakan untuk memberikan detail – detail pada layout, disini misalnya saya memberikan imageview untuk menaruh gambar, kemudian terdapat textview yang saya gunakan untuk bagian title / judul, penjelasan / deskripsi. Menggunakan android:id”@+id/nama” untuk memberikan id pada text atau juga bisa digunakan pada imageview, untuk memanggil pada java menggunakan id tersebut.



## C. Item.xml



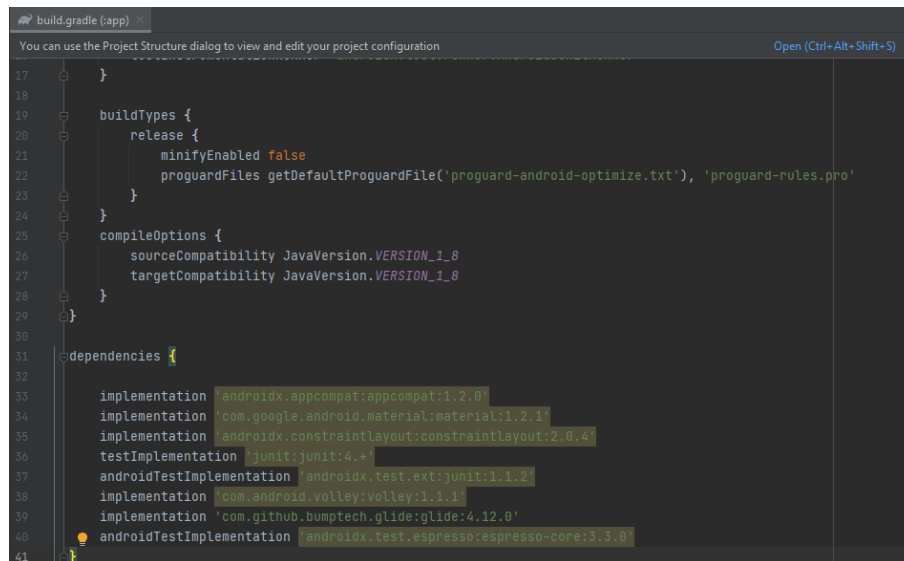
Gambar 4. Item.xml



Gambar 5. Item.xml

Item.xml digunakan untuk memberikan layout pada bagian – bagian partikel item seperti cardview dan didalamnya terdapat imageview dan juga textview, pada textview di item.xml adalah untuk overview.

## D. Gradle



```

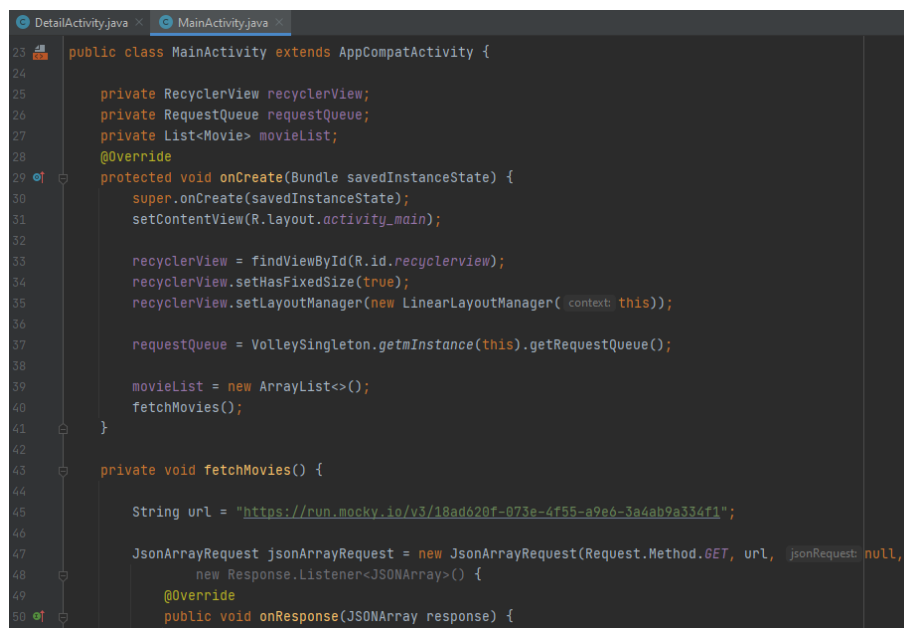
17 }
18
19 buildTypes {
20     release {
21         minifyEnabled false
22         proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
23     }
24 }
25 compileOptions {
26     sourceCompatibility JavaVersion.VERSION_1_8
27     targetCompatibility JavaVersion.VERSION_1_8
28 }
29 }
30
31 dependencies {
32
33     implementation 'androidx.appcompat:appcompat:1.2.0'
34     implementation 'com.google.android.material:material:1.2.1'
35     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
36     testImplementation 'junit:junit:4.+'
37     androidTestImplementation 'androidx.test.ext:junit:1.1.2'
38     implementation 'com.android.volley:volley:1.1.1'
39     implementation 'com.github.bumptech.glide:glide:4.12.0'
40     androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
41 }

```

Gambar 6. Gradle

Pada gradle yang harus ditambahkan dalam dependencies adalah implementation androidx.appcompat, com.google.android.material dimana untuk mengambil material android yang nantinya digunakan dalam aplikasi, dan yang paling inti adalah com.android.volley:volley yang nantinya akan digunakan untuk mengambil atau menerapkan sistem volley.

## E. MainActivity.java



```

23 public class MainActivity extends AppCompatActivity {
24
25     private RecyclerView recyclerView;
26     private RequestQueue requestQueue;
27     private List<Movie> movieList;
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_main);
32
33         recyclerView = findViewById(R.id.recyclerview);
34         recyclerView.setHasFixedSize(true);
35         recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
36
37         requestQueue = VolleySingleton.getInstance(this).getRequestQueue();
38
39         movieList = new ArrayList<>();
40         fetchMovies();
41     }
42
43     private void fetchMovies() {
44
45         String url = "https://run.mocky.io/v3/18ad620f-073e-4f55-a9e6-3a4ab9a334f1";
46
47         JSONArrayRequest jsonArrayRequest = new JSONArrayRequest(Request.Method.GET, url, jsonArrayRequest: null,
48             new Response.Listener<JSONArray>() {
49                 @Override
50                 public void onResponse(JSONArray response) {

```

Gambar 7. Mainactivity.java

```

52 for (int i = 0 ; i < response.length() ; i ++){
53     try {
54         JSONObject jsonObject = response.getJSONObject(i);
55         String title = jsonObject.getString( name: "title");
56         String overview = jsonObject.getString( name: "overview");
57         String poster = jsonObject.getString( name: "poster");
58         Double rating = jsonObject.getDouble( name: "rating");
59
60         Movie movie = new Movie(title , poster , overview , rating);
61         movieList.add(movie);
62     } catch (JSONException e) {
63         e.printStackTrace();
64     }
65
66     MovieAdapter adapter = new MovieAdapter( context: MainActivity.this , movieList);
67
68     recyclerView.setAdapter(adapter);
69 }
70 }, new Response.ErrorListener() {
71     @Override
72     public void onErrorResponse(VolleyError error) {
73         Toast.makeText( context: MainActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
74     }
75 }
76 });
77
78 requestQueue.add(jsonArrayRequest);
79 }

```

Gambar 8. MainActivity.java

Mainactivity.java disini untuk memberikan effect serta aksi pada layout activity\_main.xml. Terdapat set content view untuk memanggil activity\_main. Untuk recyclerview disini menggunakan posisi tetap supaya yaitu dengan memberikan perintah `recyclerview.sethasfixedsize(true);`

Data yang didapat untuk ditampilkan pada aplikasi yaitu pada bagian `privateMovies()` disini diberikan url dengan tipe data String yaitu dengan perintah `String url = "link"` jadi program akan menjalankan perintah untuk mengambil data yang ada di link tersebut untuk ditampilkan dalam aplikasi.

Kemudian untuk array pemberian atau memasukkan content dari link tersebut dengan `for`, selanjutnya dalam `try` dimasukkan apa yang dibutuhkan dan item sudah di deklarasi dan di inisiasi dalam `try` tersebut yaitu seperti `jsonObject`, `title`, `overview`, `poster`, dan juga `rating`.

## F. DetailActivity.java

```

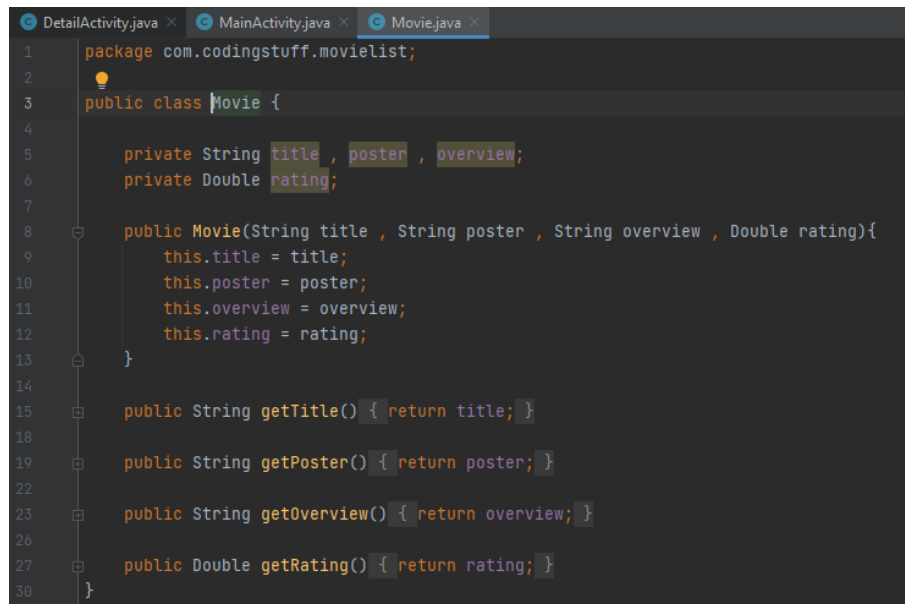
10
11 import ...
12
13 public class DetailActivity extends AppCompatActivity {
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_detail);
19
20         ImageView imageView = findViewById(R.id.poster_image);
21         TextView rating_tv = findViewById(R.id.mRating);
22         TextView title_tv = findViewById(R.id.mTitle);
23         TextView overview_tv = findViewById(R.id.mOverview_tv);
24
25         Bundle bundle = getIntent().getExtras();
26
27         String mTitle = bundle.getString( key: "title");
28         String mPoster = bundle.getString( key: "poster");
29         String mOverview = bundle.getString( key: "overview");
30         double mRating = bundle.getDouble( key: "rating");
31
32         Glide.with( activity: this).load(mPoster).into(imageView);
33         rating_tv.setText(Double.toString(mRating));
34         title_tv.setText(mTitle);
35         overview_tv.setText(mOverview);
36     }
37 }

```

Gambar 9. DetailActivity.java

DetailActivity.java disini memberikan aksi pada layout activity\_detail.xml seperti pada bagian perintah setContentView(R.layout.activity\_detail). Selanjutnya pengambilan item dalam layout activity\_detail yaitu dengan pemanggilan jenis item layout disini terdapat imageView = findViewById dengan id yang bernama poster\_image dengan perintah R.id.poster\_image Begitu pula dengan yang lainnya dengan memanggil jenis item layoutnya misal mRating disini mRating menggunakan TextView.Selanjutnya untuk tipe data dibawahnya String ada mtitle, mposter, moverview, dan mrating menggunakan tipe data double.

## G. Movie.java



```

1 package com.codingstuff.movielist;
2
3 public class Movie {
4
5     private String title , poster , overview;
6     private Double rating;
7
8     public Movie(String title , String poster , String overview , Double rating){
9         this.title = title;
10        this.poster = poster;
11        this.overview = overview;
12        this.rating = rating;
13    }
14
15    public String getTitle() { return title; }
18
19    public String getPoster() { return poster; }
22
23    public String getOverview() { return overview; }
26
27    public Double getRating() { return rating; }
30 }

```

Gambar 10. movie.java

Pada movie.java digunakan untuk memberikan tipe data dan juga pemanggilan atau inisiasi, seperti pemberian tipe data private String untuk tipe data string disini terdapat title, poster dan overview. Kemudian untuk tipe data Double disini hanya untuk bagian rating. Selanjutnya dilakukan pemanggilan dan penginisiasi pada deklarasi tipe data diatas.

## H. MovieAdapter.java

```

6  import android.view.LayoutInflater;
7  import android.view.View;
8  import android.view.ViewGroup;
9  import android.widget.ImageView;
10 import android.widget.TextView;
11
12 import androidx.annotation.NonNull;
13 import androidx.constraintlayout.widget.ConstraintLayout;
14 import androidx.recyclerview.widget.RecyclerView;
15
16 import com.bumptech.glide.Glide;
17
18 import java.util.List;
19
20 public class MovieAdapter extends RecyclerView.Adapter<MovieAdapter.MovieHolder> {
21
22     private Context context;
23     private List<Movie> movieList;
24
25     public MovieAdapter(Context context, List<Movie> movies){
26         this.context = context;
27         movieList = movies;
28     }
29
30     @NonNull
31     @Override
32     public MovieHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
33         View view = LayoutInflater.from(context).inflate(R.layout.item, parent, attachToRoot: false);
34         return new MovieHolder(view);
35     }

```

Gambar 11. MovieAdapter.java

```

35
36     @Override
37     public void onBindViewHolder(@NonNull MovieHolder holder, int position) {
38
39         Movie movie = movieList.get(position);
40         holder.rating.setText(movie.getRating().toString());
41         holder.title.setText(movie.getTitle());
42         holder.overview.setText(movie.getOverview());
43         Glide.with(context).load(movie.getPoster()).into(holder.imageView);
44
45         holder.constraintLayout.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View v) {
48                 Intent intent = new Intent(context, DetailActivity.class);
49
50                 Bundle bundle = new Bundle();
51                 bundle.putString("title", movie.getTitle());
52                 bundle.putString("overview", movie.getOverview());
53                 bundle.putString("poster", movie.getPoster());
54                 bundle.putDouble("rating", movie.getRating());
55
56                 intent.putExtras(bundle);
57
58                 context.startActivity(intent);
59             }
60         });
61     }
62 }

```

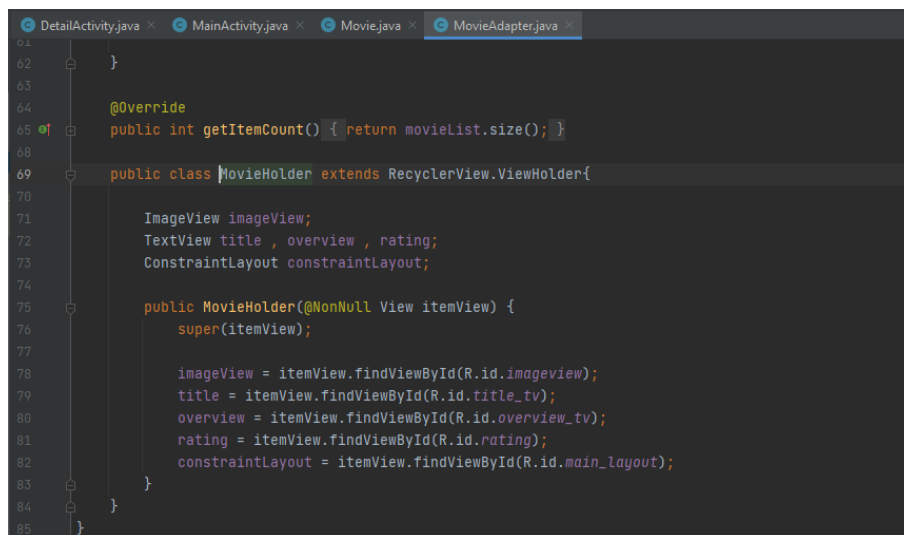
Gambar 12. MovieAdapter.java

Untuk bagian penghubung disini menggunakan nama MovieAdapter dimana movieadapter mengambil data atau menghubungkan dengan extends RecyclerView.Adapter<MovieAdapter.MovieHolder>. Kemudian untuk MovieHolder disini mengambil LayoutInflater dari context dan meng

infalte dari layout item.xml kemudian untuk menjalankan perulangan  
return new MovieHolder(view).

Langkah selanjutnya pengambilan data onBindViewHolder dimana  
movieList.get atau mengambil (position) dan berkaitan dengan xml atau  
layout yang sudah dirancang. Kemudian untuk memberikan aksi click  
pada OnClickListener disini menggunakan pemanggilan dengan  
tipedatanya terlebih dahulu misal bundle.putString lalu nama idnya Title  
jadi movie.getTitle(), dan untuk memulai atau menjalankan perintah  
context.startActivity(intent);

## I. MovieHolder

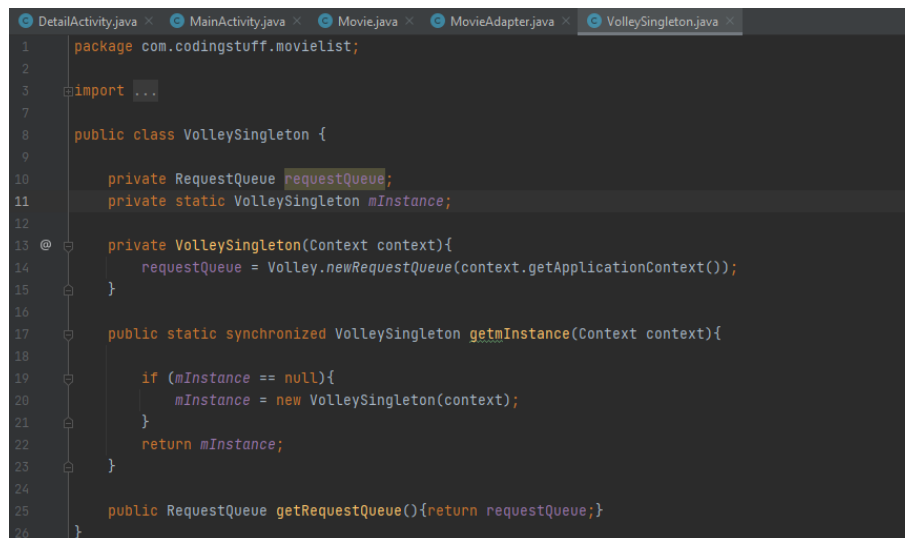


```
61  
62 }  
63  
64 @Override  
65 public int getItemCount() { return movieList.size(); }  
66  
67  
68  
69 public class MovieHolder extends RecyclerView.ViewHolder {  
70  
71     ImageView imageView;  
72     TextView title , overview , rating;  
73     ConstraintLayout constraintLayout;  
74  
75     public MovieHolder(@NonNull View itemView) {  
76         super(itemView);  
77  
78         imageView = itemView.findViewById(R.id.imageview);  
79         title = itemView.findViewById(R.id.title_tv);  
80         overview = itemView.findViewById(R.id.overview_tv);  
81         rating = itemView.findViewById(R.id.rating);  
82         constraintLayout = itemView.findViewById(R.id.main_layout);  
83     }  
84 }  
85 }
```

Gambar 13. movieholder

Untuk movieholder ada didalam MovieAdapter hanya dengan  
membuat public class dan menghubungkan dengan perintah extends  
RecyclerView.ViewHolder. Disini digunakan untuk pemberian deklarasi  
yaitu dengan ImageView berisi id ImageView, selanjutnya pada TextView  
berisi title,overview, dan juga rating. Jadi digunakan untuk  
pengelompokan masing masing tipe layout seperti imageview dan juga  
textview. Selanjutnya pengambilan susunan layout dari xml dengan  
pemanggilan id nya misal R.id.title\_tv, dan seterusnya.

## J. VolleySingleton.java



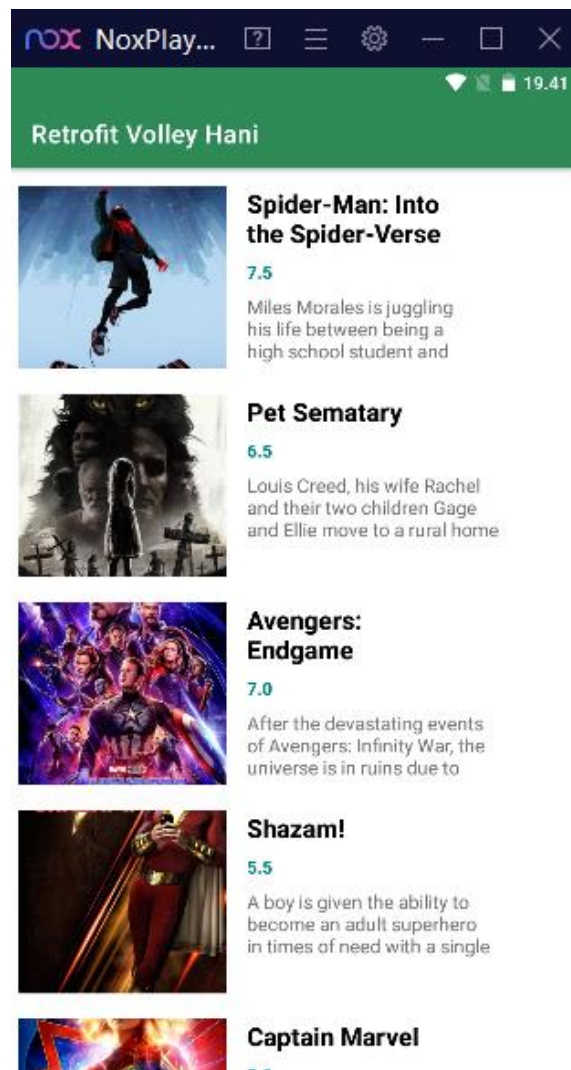
```
1 package com.codingstuff.movielist;
2
3 import ...
4
5
6
7
8 public class VolleySingleton {
9
10     private RequestQueue requestQueue;
11     private static VolleySingleton mInstance;
12
13     @
14     private VolleySingleton(Context context){
15         requestQueue = Volley.newRequestQueue(context.getApplicationContext());
16     }
17
18     public static synchronized VolleySingleton getInstance(Context context){
19
20         if (mInstance == null){
21             mInstance = new VolleySingleton(context);
22         }
23         return mInstance;
24     }
25
26     public RequestQueue getRequestQueue(){return requestQueue;}
27 }
```

Gambar 14. VolleySingleton.java

Pada bagian VolleySingleton.java digunakan untuk menjalankan permintaan atau request seperti aplikasi newRequestQueue dalam context untuk mendapatkan yaitu dengan getApplicationContext. Untuk sinkronasi disini dengan perintah public static synchronized maka VolleySingleton sinkronasi dengan getInstance(Context context).

## K. Output





Gambar 15. output

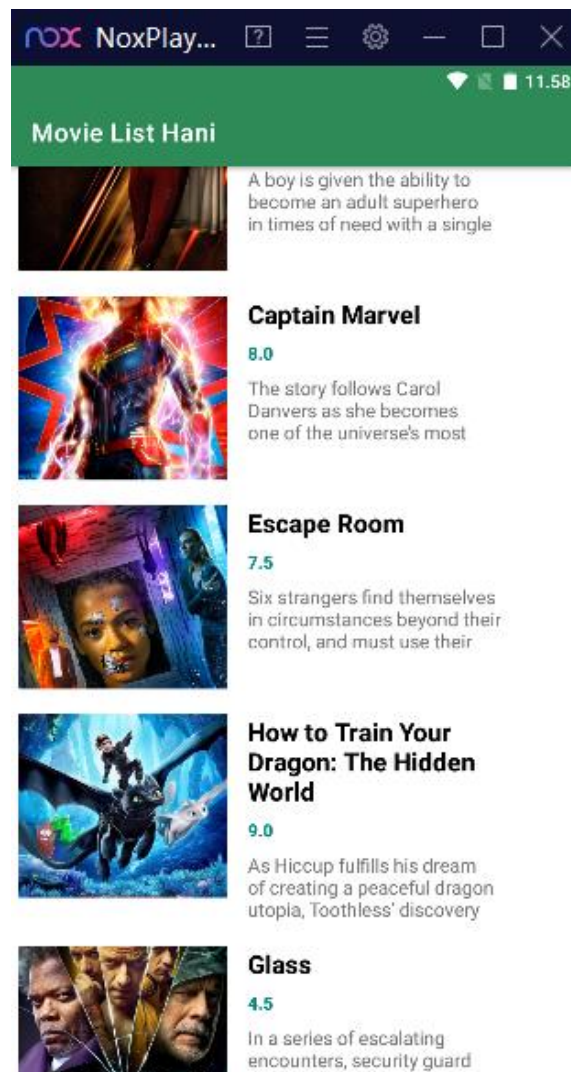


### Avengers: Endgame

7.0

After the devastating events of Avengers: Infinity War, the universe is in ruins due to the efforts of the Mad Titan, Thanos. With the help of remaining allies, the Avengers must assemble once more in order to undo Thanos' actions and restore order to the universe once and for all, no matter what consequences may be in store.

Gambar 16. output



Gambar 17. output

## **BAB III**

### **PENUTUP**

#### **3.1. Kesimpulan**

Mahasiswa memahami serta menerapkan konsep retrofit, dengan implementasi source code pada kasus tertentu. Mahasiswa memahami serta menerapkan konsep volley, dengan implementasi source code pada kasus tertentu. Volley unggul di operasi tipe RPC yang digunakan untuk mengisi UI, seperti mengambil halaman hasil penelusuran sebagai data terstruktur. Volley terintegrasi secara mudah dengan protokol apa pun dan memberi performa terbaik dengan dukungan untuk string raw, gambar, dan JSON.

## **DAFTAR PUSTAKA**

Khoiriyah, H. (2021). *Laporan Workshop Mobile Applications*. Jember:  
Politeknik Negeri Jember.