

Kemampuan Akhir Yang Direncanakan

- Mahasiswa mampu menginstall dan mengkonfigurasi retrofit pada project Android.
- Mahasiswa mampu membuat aplikasi yang memiliki request REST API berupa POST, GET, UPDATE dan DELETE sederhana ke server.

Modul

Retrofit

Resource yang berbentuk format JSON yang disediakan REST server dapat dimanfaatkan oleh aplikasi android dengan library Retrofit. Retrofit adalah library Rest Client untuk android dan java dari squareup. hal ini membuatnya relatif mudah untuk mengambil dan mengunggah JSON (atau struktur data lainnya) melalui webservice berbasis REST. Di Retrofit Anda mengonfigurasi konverter mana yang digunakan untuk serialisasi data. Biasanya untuk JSON menggunakan GSON, tetapi Anda dapat menambahkan konverter khusus untuk memproses XML atau protokol lain.

Setup retrofit Pastikan untuk meminta izin Internet di file AndroidManifest.xml Anda:

```
<manifest xmlns:android= "http://schemas.android.com/apk/res/android" >  
    <uses-permission android:name= "android.permission.INTERNET" />  
</manifest>
```

Tambahkan berikut ini ke file app/build.gradle :

```
dependencies {  
    implementation 'com.google.code.gson:gson:2.8.5'  
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'  
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'  
}
```

Retrofit dapat dikonfigurasi untuk menggunakan converter. Converter ini menangani de(serialization) data. Beberapa converter sudah tersedia untuk berbagai format serialisasinya.

Untuk mengonversi JSON:

GSON: com.squareup.retrofit:converter-gson

Jackson: com.squareup.retrofit:converter-jackson

Moshi: com.squareup.retrofit:converter-moshi

Untuk mengkonversi Protocol Buffer:

Protobuf: com.squareup.retrofit:converter-protobuf

Untuk mengkonversi Protocol Buffer:

Wire: com.squareup.retrofit:converter-wire

Untuk mengonversi XML:

Simple XML: com.squareup.retrofit:converter-simplexml

Creating Retrofit

Untuk mengirim permintaan jaringan ke API, kita perlu menggunakan kelas builder Retrofit dan menentukan URL dasar untuk layanan tersebut.

```
public static final String BASE_URL = "http://api.myservice.com/";
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(GsonConverterFactory.create())
    .build();
```

Untuk deserialisasi respons menggunakan library Gson.

```
Gson gson = new GsonBuilder()
    .setDateFormat("yyyy-MM-dd'T'HH:mm:ssZ")
    .create();

Retrofit retrofit = new Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(GsonConverterFactory.create(gson))
    .build();
```

Mendefinisikan Endpoint

Cara mendefinisikan setiap endpoint dengan cara berikut:

```
public interface MyApiEndpointInterface {
    // Request method and URL specified in the annotation

    @GET("users/{username}")
    Call<User> getUser(@Path("username") String username);

    @GET("group/{id}/users")
    Call<List<User>> groupList(@Path("id") int groupId, @Query("sort")
    String sort);

    @POST("users/new")
    Call<User> createUser(@Body User user);
}
```

Perlu diperhatikan bahwa beberapa jenis endpoint menentukan anotasi dari HTTP method (GET, POST, dll). Berikut beberapa anotasinya.

@Path-substitusi variabel untuk API endpoint (yaitu nama pengguna akan ditukarkan untuk {username} di endpoint URL).

@Query-menentukan nama kunci query dengan nilai parameter beranotasi.

@Body-payload untuk panggilan POST (diserialisasikan dari objek Java ke string JSON)

@Header-menentukan header dengan nilai parameter beranotasi

Mengubah base URL

URL dasar didefinisikan saat Anda memulai Retrofit instance. Retrofit 2 memungkinkan Anda mengganti URL dasar yang ditentukan dengan mengubahnya dalam anotasi (jika Anda perlu mempertahankan sebuah endpoint menggunakan API endpoint sebelumnya).

```
@POST("https://api.github.com/api/v3")
```

POSTing JSON

Retrofit akan menggunakan pustaka konverter yang dipilih untuk menangani deserialisasi data dari objek Java. Jika Anda membubuhi keterangan parameter dengan parameter @Body, Hal ini akan bekerja secara otomatis. Jika Anda menggunakan library GSon misalnya, bidang apa pun yang termasuk class akan diserialkan untuk. Kita dapat mengubah nama ini menggunakan decorator @SerializedName.

```
public class User {  
  
    @SerializedName("id")  
    int mId;  
  
    @SerializedName("name")  
    String mName;  
  
    public User(int id, String name ) {  
        this.mId = id;  
        this.mName = name;  
    }  
}
```

Endpoint nya akan seperti ini:

```
@POST("/users/new")  
Call<User> createUser(@Body User user);
```

Berikut adalah untuk pemanggilan API nya:

```
User user = new User(123, "John Doe");
Call<User> call = apiService.createUser(user);
call.enqueue(new Callback<User>() {
    @Override
    public void onResponse(Call<User> call, Response<User> response) {

    }

    @Override
    public void onFailure(Call<User> call, Throwable t) {

    }
})
```

Hasilnya akan seperti berikut:

```
{"name":"John Doe","id":123}
```

Mengakses API

Untuk mengakses API kita dapat menggunakan MyApiEndpointInterface dengan mendefinisikan endpoint.

```
MyApiEndpointInterface apiService =
    retrofit.create(MyApiEndpointInterface.class);
```

Untuk menggunakan API secara asynchronously, kita dapat menggunakan service berikut ini:

```
String username = "sarahjean";
Call<User> call = apiService.getUser(username);
call.enqueue(new Callback<User>() {
    @Override
    public void onResponse(Call<User> call, Response<User> response) {
        int statusCode = response.code();
        User user = response.body();
    }

    @Override
    public void onFailure(Call<User> call, Throwable t) {
        // Log error here since request failed
    }
});
```

Praktikum

1. Lakukan registrasi di <http://themoviedb.org> untuk mendapatkan API Key. Setelah registrasi masuk ke **Profile -> Setting -> API -> Request API Key**.
2. Silahkan isi biodata, kemudian akan muncul Example API Request dimana Key nya yang akan kita gunakan nanti.
3. Buatlah sebuah Project baru
4. Buka **build.gradle** tambahkan dependensi Retrofit, Gson, dan RecyclerView.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-
layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-
core:3.0.2'

    //Retrofit, Gson, RecyclerView
    implementation 'com.google.code.gson:gson:2.6.2'
    implementation 'com.squareup.retrofit2:retrofit:2.0.2'
    implementation 'com.squareup.retrofit2:converter-gson:2.0.2'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
}
```

5. Karena retrofit ini bekerja dengan menggunakan koneksi internet, maka perlu menambahkan izin internet di **AndroidManifest.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.odhitya.retrofitandroid">

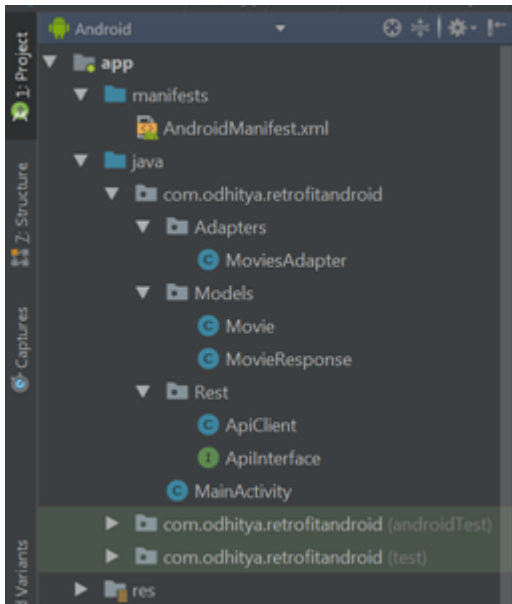
<uses-permission android:name="android.permission.INTERNET"/>

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

```
</manifest>
```

6. Buatlah 3 sub package yaitu **Adapters**, **Models**, **Rest** dan beberapa java class sesuai sub package nya seperti dibawah ini.



7. Untuk mengetahui isi API yang kita dapat dari <http://themoviedb.org> kalian bisa menggunakan JSON viewer seperti yang terdapat pada link ini <http://jsonviewer.stack.hu/>
8. Buatlah class dengan nama **Movie.java** di dalam **Models** package.

```
package com.odhitya.retrofitandroid.Models;

import com.google.gson.annotations.SerializedName;

public class Movie {

    @SerializedName("overview")
    private String overview;
    @SerializedName("release_date")
    private String releaseDate;
    @SerializedName("id")
    private Integer id;
    @SerializedName("title")
    private String title;
    @SerializedName("vote_average")
    private Double voteAverage;

    public Movie(String overview, String releaseDate, Integer id,
                  String title, Double voteAverage) {
        this.overview = overview;
        this.releaseDate = releaseDate;
        this.id = id;
        this.title = title;
        this.voteAverage = voteAverage;
    }
}
```

```
}

public String getOverview() {
    return overview;
}

public void setOverview(String overview) {
    this.overview = overview;
}

public String getReleaseDate() {
    return releaseDate;
}

public void setReleaseDate(String releaseDate) {
    this.releaseDate = releaseDate;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public Double getVoteAverage() {
    return voteAverage;
}

public void setVoteAverage(Double voteAverage) {
    this.voteAverage = voteAverage;
}
}
```

9. Buatlah class **MovieResponse.java** di dalam package **Models**, class ini berisi semua film dan informasi tambahan yang diambil.

```
package com.odhitya.retrofitandroid.Models;

import com.google.gson.annotations.SerializedName;

import java.util.List;
```

```
public class MovieResponse {
    @SerializedName("page")
    private int page;
    @SerializedName("results")
    private List<Movie> results;
    @SerializedName("total_results")
    private int totalResults;
    @SerializedName("total_pages")
    private int totalPages;

    public int getPage() {
        return page;
    }

    public void setPage(int page) {
        this.page = page;
    }

    public List<Movie> getResults() {
        return results;
    }

    public void setResults(List<Movie> results) {
        this.results = results;
    }

    public int getTotalResults() {
        return totalResults;
    }

    public void setTotalResults(int totalResults) {
        this.totalResults = totalResults;
    }

    public int getTotalPages() {
        return totalPages;
    }

    public void setTotalPages(int totalPages) {
        this.totalPages = totalPages;
    }
}
```

10. Untuk mengirim permintaan ke API, kita perlu menggunakan class builder retrofit dan menentukan URL dasar untuk layanan tersebut. Buatlah class bernama **ApiClient.java** di dalam package **Rest**.

```
package com.odhitya.retrofitandroid.Rest;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
```



```

public class ApiClient {

    public static final String BASE_URL = "http://api.themoviedb.org/3/";
    private static Retrofit retrofit = null;

    public static Retrofit getClient() {
        if (retrofit==null) {
            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}

```

11. Endpoint didefinisikan dalam interface menggunakan anotasi retrofit khusus untuk encode detail tentang parameter dan metode permintaan. Buatlah **ApiInterface.java** di dalam package **Rest**.

```

package com.odhitya.retrofitandroid.Rest;

import com.odhitya.retrofitandroid.Models.MovieResponse;
import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Path;
import retrofit2.http.Query;

public interface ApiInterface {
    @GET("movie/top_rated")
    Call<MovieResponse> getTopRatedMovies(@Query("api_key") String apiKey);

    @GET("movie/{id}")
    Call<MovieResponse> getMovieDetails(@Path("id") int id,
    @Query("api_key") String apiKey);
}

```

12. Buka **colors.xml** pada **res -> values** kemudian tambahkan warna seperti dibawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="orange">#FF3909</color>
    <color name="colorAccentDark">#00B482</color>
    <color name="colorBlack">#555555</color>

```

```

<color name="colorWhite">#FFFFFF</color>
<color name="colorGrey">#707070</color>
<color name="colorGreyLight">#8A8A8A</color>
</resources>

```

13. Buka **activity_main.xml** kemudian buatlah sebuah widget recycler view di dalam nya.

```

<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

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

</RelativeLayout>

```

14. Buatlah layout dengan nama **list_item_movie.xml** kemudian isi dengan widget seperti dibawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:id="@+id/movies_layout"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:minHeight="72dp"
    android:orientation="horizontal"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="top"
            android:paddingRight="16dp"
            android:textStyle="bold"

```

```

        android:textColor="@color/colorBlack"
        android:textSize="16sp" />

<TextView
    android:id="@+id/subtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingRight="16dp"
    android:textColor="@color/colorGreyLight" />

<TextView
    android:id="@+id/description"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:maxLines="3"
    android:paddingRight="16dp"
    android:textColor="@color/colorGreyLight" />

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="35dp"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/star"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/star" />

    <TextView
        android:id="@+id/rating"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:text="5.0" />

</LinearLayout>

</LinearLayout>

```

15. Buatlah adapter untuk recycler view di dalam package **Adapters** dengan nama **MoviesAdapter.java**

```

package com.odhitya.retrofitandroid.Adapters;

import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;

```

```
import android.widget.TextView;

import com.odhitya.retrofitandroid.Models.Movie;
import com.odhitya.retrofitandroid.R;

import java.util.List;

public class MoviesAdapter extends
RecyclerView.Adapter<MoviesAdapter.MovieViewHolder> {

    private List<Movie> movies;
    private int rowLayout;
    private Context context;

    public static class MovieViewHolder extends RecyclerView.ViewHolder {
        LinearLayout moviesLayout;
        TextView movieTitle;
        TextView data;
        TextView movieDescription;
        TextView rating;

        public MovieViewHolder(View v) {
            super(v);
            moviesLayout = (LinearLayout)
v.findViewById(R.id.movies_layout);
            movieTitle = (TextView) v.findViewById(R.id.title);
            data = (TextView) v.findViewById(R.id.subtitle);
            movieDescription = (TextView)
v.findViewById(R.id.description);
            rating = (TextView) v.findViewById(R.id.rating);
        }
    }

    public MoviesAdapter(List<Movie> movies, int rowLayout, Context
context) {
        this.movies = movies;
        this.rowLayout = rowLayout;
        this.context = context;
    }

    @Override
    public MoviesAdapter.MovieViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(rowLayout, parent,
false);
        return new MovieViewHolder(view);
    }

    @Override
    public void onBindViewHolder(MovieViewHolder holder, final int
position) {
        holder.movieTitle.setText(movies.get(position).getTitle());
        holder.data.setText(movies.get(position).getReleaseDate());
    }
}
```

```

holder.movieDescription.setText(movies.get(position).getOverview());

holder.rating.setText(movies.get(position).getVoteAverage().toString());
}

@Override
public int getItemCount() {
    return movies.size();
}
}

```

16. Bukalah **MainActivity.java** kemudian buatlah sebuah request dan pastikan anda memasukkan **API_KEY** yang didapat dengan benar.

```

package com.odhitya.retrofitandroid;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.widget.Toast;

import com.odhitya.retrofitandroid.Adapters.MoviesAdapter;
import com.odhitya.retrofitandroid.Models.Movie;
import com.odhitya.retrofitandroid.Models.MovieResponse;
import com.odhitya.retrofitandroid.Rest.ApiClient;
import com.odhitya.retrofitandroid.Rest.ApiInterface;

import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MainActivity extends AppCompatActivity {
    private static final String TAG = MainActivity.class.getSimpleName();

    private final static String API_KEY =
"ea91e0ad33feadabc80e140983b537df";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (API_KEY.isEmpty()) {
            Toast.makeText(getApplicationContext(), "Please obtain your
API KEY first from themoviedb.org", Toast.LENGTH_LONG).show();
            return;
        }
    }
}

```

```
    }

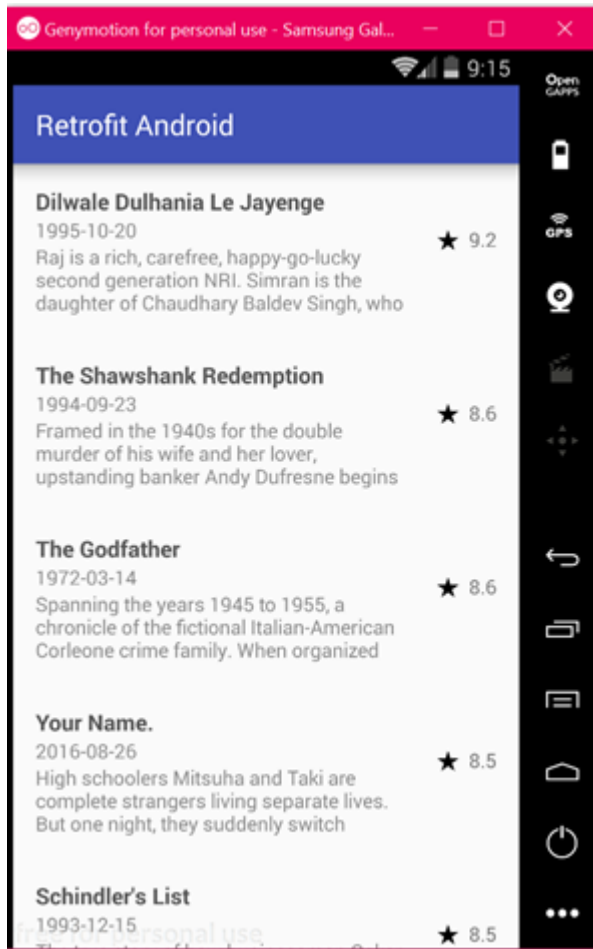
    final RecyclerView recyclerView = (RecyclerView)
    findViewById(R.id.movies_recycler_view);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));

    ApiInterface apiService =
    ApiClient.getClient().create(ApiInterface.class);

    Call<MovieResponse> call = apiService.getTopRatedMovies(API_KEY);
    call.enqueue(new Callback<MovieResponse>() {
        @Override
        public void onResponse(Call<MovieResponse>call,
        Response<MovieResponse> response) {
            int statusCode = response.code();
            List<Movie> movies = response.body().getResults();
            recyclerView.setAdapter(new MoviesAdapter(movies,
            R.layout.list_item_movie, getApplicationContext()));
        }

        @Override
        public void onFailure(Call<MovieResponse>call, Throwable t) {
            // Log error here since request failed
            Log.e(TAG, t.toString());
        }
    });
}
```

17. Coba jalankan aplikasi maka hasilnya akan seperti dibawah ini.



Tugas

- Dari latihan praktikum diatas, silahkan melanjutkannya dengan menambahkan fitur intent pada tampilan recycler view nya, sehingga bisa masuk kedalam activity yang berisikan informasi detail dari setiap list yang di klik pada tampilan recycler view.