

TUGAS PERTEMUAN 5  
PRAKTIKUM APP MOBILE  
Dosen Pengampu : Nurfiah, S.ST, M.Kom



Disusun Oleh :  
Yusuf Fatha Mubina Alexander (2211533017)

**Departemen Informatika**  
**Fakultas Teknologi Informasi**  
**Universitas Andalas**  
**2024**

## A. Model

- Java

Kelas ini merupakan model untuk "Pelanggan" dengan kolom untuk id, nama, email, dan hp. Setiap kolom memiliki metode *getter* dan *setter* sendiri.

```
package com.example.connecttojson;

public class ModelPelanggan {
    String id, nama, email, hp;

    public String getId() {return id;}
    public void setId(String id) {this.id = id;}

    public String getNama() {return nama;}
    public void setNama(String nama) {this.nama = nama;}

    public String getEmail() {return email;}
    public void setEmail(String email) {this.email = email;}

    public String getHp() {return hp;}
    public void setHp(String hp) {this.hp = hp;}
}
```

## B. Adapter

Lalu kita juga perlu menambahkan adapter dalam bentuk java class untuk menampung data pelanggan. Adapter pelanggan yang kita buat akan menggunakan recycle view sehingga data pelanggan nantinya akan tertampil membentuk list.

```
package com.example.connecttojson;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class AdapterPelanggan extends
RecyclerView.Adapter<AdapterPelanggan.ViewHolder> {
    private static final String TAG = AdapterPelanggan.class.getSimpleName();
    private Context context;
    private List<ModelPelanggan> list;
    private View.OnClickListener onItemClick;

    public void setOnItemClickListener(View.OnClickListener
itemClickListener) {
        onItemClick = itemClickListener;
    }

    public AdapterPelanggan(Context context, List<ModelPelanggan> list) {
        this.context = context;
    }
}
```

```

        this.list = list;
    }

    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_pelanggan,
parent, false);
        return new RecyclerView.ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        ModelPelanggan item = list.get(position);
        holder.tvNama.setText(item.getNama());
        holder.tvHp.setText(item.getHp());
    }

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

    public void clear() {
        int size = this.list.size();
        this.list.clear();
        notifyItemRangeRemoved(0, size);
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView tvNama, tvHp;

        public ViewHolder(View itemView) {
            super(itemView);
            tvNama = (TextView) itemView.findViewById(R.id.tvItemPelNama);
            tvHp = (TextView) itemView.findViewById(R.id.tvItemPelHp);
            itemView.setTag(this);
            itemView.setOnClickListener(onItemClicked);
        }
    }
}

```

### C. MainActivity

Pada main activity kita perlu mendeklarasikan url yang dituju menggunakan ip address, recycle view, adapter, model, dan lainnya yang nantinya akan ditampilkan pada layout XML nya.

```

package com.example.connecttojson;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;

```

```

import android.provider.ContactsContract;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    //buat variabel volley untuk mengirim request ke jsno
    RequestQueue requestQueue;

    String base_url ="http://BelajarAPI" ;
    RecyclerView rvPelanggan;
    AdapterPelanggan adapterPelanggan;
    ArrayList<ModelPelanggan> list;
    ProgressDialog progressDialog;
    AlphaAnimation btnAnimasi = new AlphaAnimation(1F, 0,5F);
    TextView btnAdd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        View.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v.insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);

```

```

        return insets;
    });
    setContentView();
    requestQueue = Volley.newRequestQueue(this);
    getData();

    btnAdd.setOnClickListener() {
        @Override
        public void onClick (View v){
            Intent intent = new Intent(MainActivity.this,
AddActivity.class);
            startActivity(intent);
        }
    });
}

private void setContentView () {

    progressDialog = new progressDialog (this) ;
    btnAdd = (TextView) findViewById(R.id.btnAdd) ;
    rvPelanggan = findViewById (R.id.rvPelanggan) ;
    list = new ArrayList<>();

    LinearLayoutManager llm = new LinearLayoutManager(this) ;
    llm.setOrientation(LinearLayoutManager.VERTICAL);
    rvPelanggan.setLayoutManager(llm) ;
    rvPelanggan.setHasFixedSize(true);
}

private View.OnClickListener onClickListener = new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        v.startAnimation(btnAnimasi);
        RecyclerView.ViewHolder viewHolder = (RecyclerView.ViewHolder)
v.getTag();
        int position = viewHolder.getAdapterPosition();
        ModelPelanggan np = list.get(position);

        Intent a = new Intent(MainActivity.this, AddActivity.class);
        a.putExtra("param", "edit");
        startActivity(a);
    }
};

private void showMsg() {
    progressDialog.setTitle("Informasi");
    progressDialog.setMessage("Loading Data..");
    progressDialog.setCancelable(false);
    progressDialog.show();
}

private void getData() {
    JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(
        Request.Method.GET, base_url, null, new
Response.Listener<JSONObject>() {

        @Override

```

```

        public void onResponse(JSONObject jsonObject) {
            try {
                String status = jsonObject.getString("status");
                String message = jsonObject.getString("message");

                if (status.equalsIgnoreCase("true")) {
                    // ambil json array dari objek data
                    JSONArray jsonArray =
jsonObject.getJSONArray("data");
                    for (int a = 0; a < jsonArray.length(); a++) {
                        JSONObject jsonObject1 =
jsonArray.getJSONObject(a);
                        ModelPelanggan mp = new ModelPelanggan();
                        mp.setId(jsonObject1.getString("id"));
                        mp.setNama(jsonObject1.getString("nama"));
                        mp.setEmail(jsonObject1.getString("alamat"));
                        mp.setHp(jsonObject1.getString("hp"));
                        list.add(mp);
                    }

                    adapterPelanggan = new
AdapterPelanggan(MainActivity.this, list);
                    adapterPelanggan.notifyDataSetChanged();
                    rvPelanggan.setAdapter(adapterPelanggan);

                    //
                    adapterPelanggan.setOnItemClickListener(onClickListener);
                    Toast.makeText(MainActivity.this,
""+jsonArray.toString(), Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, ""+message,
Toast.LENGTH_SHORT).show();
                }

            } catch (JSONException e) {
                throw new RuntimeException(e);
            }
        }, new Response.ErrorListener(){
            @Override
            public void onErrorResponse(VolleyError volleyError){

            }
        });

        requestQueue.add(jsonObjectRequest);
    }
}

```

## Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvPelanggan"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:padding="16dp"
        tools:ignore="MissingConstraints">

    </androidx.recyclerview.widget.RecyclerView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

## D. Manifest

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

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.ConnectToJson"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <activity
            android:name=".AddActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```
        </intent-filter>
    </activity>
</application>

</manifest>
```

#### E. Data Postman

Pada postman kita perlu mengganti url yg dituju berdasarkan ip addressnya(akan terus berganti sesuai koneksinya).

The screenshot shows the Postman interface. At the top, a POST request is configured with the URL `http://192.168.18.24/belajar_api/layanan.php`. The request body is a JSON object: `{ "id": "3", "nama": "Cuci Sepatu", "harga": "10000" }`. Below the request, the response is displayed in the 'Body' tab, showing a JSON object: `{ "status": true, "message": "Layanan added successfully" }`. The status bar at the top right indicates a `200 OK` response. The interface includes tabs for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. The 'Body' tab is active, and the response is shown in 'Pretty' format. The 'JSON' tab is also visible, and the response is displayed in a readable format.

```
1 {
2   "id" : "3",
3   "nama" : "Cuci Sepatu",
4   "harga" : "10000"
5 }
```

Body Cookies Headers (7) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "message": "Layanan added successfully"
4 }
```



GET http://192.168.18.24/belajar\_api/layanan.php

Params Authorization Headers (7) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

1

ody Cookies Headers (7) Test Results

200 OK 29 ms 427 B

Pretty Raw Preview Visualize JSON

```
10 {
11   "id": "2",
12   "nama": "Satuan",
13   "harga": "2000"
14 },
15 {
16   "id": "4",
17   "nama": "Cuci Sepatu",
18   "harga": "10000"
19 }
20 ]
```