

LAPORAN TUGAS
APLIKASI MOBILE PRAKTIKUM 5

“Connect To Json”



Oleh:

Octavio Yusda Wildani

NIM. 2211533020

Dosen Pengampu Mata Kuliah

Nurfiah, S.ST, M.Kom

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2024

A. Model

Buatlah package baru dengan nama model, kemudian tambahkan kelas baru didalam package model dengan nama ModelPelanggan.java. selanjutnya tambahkan kode program berikut:

```
1  package com.mylauvi.connecttojson;
2
3  public class ModelPelanggan { 7 usages
4      String id, nama, alamat, hp; 2 usages
5
6  >  public String getId() { return id; }
9
10 >  public void setId(String id) { this.id = id; }
13
14 >  public String getName() { return nama; }
17
18 >  public void setName(String nama) { this.nama = nama; }
21
22 >  public String getAddress() { return alamat; }
25
26 >  public void setAddress(String alamat) { this.alamat = alamat; }
29
30 >  public String getHp() { return hp; }
33
34 >  public void setHp(String hp) { this.hp = hp; }
37 }
```

B. Adapter

File AdapterPelanggan berfungsi untuk menampilkan data pelanggan di dalam RecyclerView. Adapter ini menyimpan konteks dan daftar pelanggan yang dapat diakses oleh metode lainnya. Metode onCreateViewHolder memuat layout item, sedangkan onBindViewHolder menghubungkan data ke tampilan. Metode getItemCount menyediakan jumlah item, dan clear berfungsi untuk membersihkan serta memperbarui daftar. ViewHolder menyimpan referensi tampilan dan listener untuk menangani klik pada item.

```

package com.mylauvi.connecttojson;

> import ...

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 ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_pelanggan,
            parent, attachToRoot: false);
        return new 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(onItemClick);
        }
    }
}

```

C. MainActivity

Pada MainActivity.java, kita inisialisasi RecyclerView, AdapterPelanggan, dan List untuk menampilkan data pelanggan yang diterima dari JSON. Kita hubungkan RecyclerView dengan adapter menggunakan setAdapter(), sehingga daftar pelanggan akan muncul di emulator atau aplikasi.

```
package com.mylauvi.connecttojson;

import ...

public class MainActivity extends BaseActivity {
    // buat variable volley untuk mengirim request ke json
    RequestQueue requestQueue; 2 usages
    RecyclerView rvPelanggan; 4 usages
    AdapterPelanggan adapterPelanggan; 4 usages
    ArrayList<ModelPelanggan> list; 5 usages
    ProgressDialog progressDialog; 6 usages
    AlphaAnimation btnAnimasi = new AlphaAnimation(1F,0.5F); 1 usage
    Button btnAdd,btnReload; 2 usages
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);
        ViewCompat.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;
        });
        requestQueue = Volley.newRequestQueue( context: this);
        setContentView();
        getData();
        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent a = new Intent( packageContext: MainActivity.this, AddActivity.class);
                a.putExtra( name: "param", value: "add");
                startActivityForResult(a, requestCode: 200);
            }
        });
    }
}
```

```

        btnReload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { getData(); }
        });
    }

    private View.OnClickListener onClickListener = new View.OnClickListener() { 1 usage
        @Override
        public void onClick(View v) {
            v.startAnimation(btnAnimasi);
            RecyclerView.ViewHolder viewHolder = (RecyclerView.ViewHolder) v.getTag();
            int position = viewHolder.getAdapterPosition();
            ModelPelanggan mp = list.get(position);

            Intent a = new Intent( packageContext: MainActivity.this, AddActivity.class);
            a.putExtra( name: "param", value: "edit");
            a.putExtra( name: "id", mp.getId());
            a.putExtra( name: "nama", mp.getNama());
            a.putExtra( name: "alamat", mp.getAlamat());
            a.putExtra( name: "hp", mp.getHp());
            startActivityForResult(a, requestCode: 200);
        }
    };

    private void showMsg() { 1 usage
        progressDialog.setTitle("Informasi");
        progressDialog.setMessage("Loading Data..");
        progressDialog.setCancelable(false);
        progressDialog.show();
    }
}

```

```

    private void setView() { 1 usage
        progressDialog = new ProgressDialog( context: this);
        rvPelanggan = (RecyclerView) findViewById(R.id.rvPelanggan);
        btnAdd = (Button) findViewById(R.id.btnAdd);
        btnReload = (Button) findViewById(R.id.btnReload);
        list = new ArrayList<>();
        LinearLayoutManager llm = new LinearLayoutManager( context: this);
        llm.setOrientation(LinearLayoutManager.VERTICAL);
        rvPelanggan.setHasFixedSize(true);
        rvPelanggan.setLayoutManager(llm);
    }

    private void getData() { 3 usages
        list.clear();
        showMsg();
        progressDialog.dismiss();
        JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(Request.Method.GET,
            Config.base_url, jsonRequest: null, new Response.Listener<JSONObject>() {
            @Override 3 usages
            public void onResponse(JSONObject jsonObject) {
                try {
                    String status = jsonObject.getString( name: "status");
                    String message = jsonObject.getString( name: "message");
                    if (status.equalsIgnoreCase( anotherString: "true")){
                        // ambil json array dari object data ex : data
                        JSONArray jsonArray = jsonObject.getJSONArray( name: "data");
                        for (int a=0; a<jsonArray.length(); a++){
                            // ambil json object dari data array (ex : id, nama, alamat, hp)
                            JSONObject object = jsonArray.getJSONObject(a);
                            ModelPelanggan mp = new ModelPelanggan();
                            mp.setId(object.getString( name: "id"));
                            mp.setNama(object.getString( name: "nama"));
                            mp.setAlamat(object.getString( name: "alamat"));
                            mp.setHp(object.getString( name: "hp"));
                        }
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

        list.add(mp);
    }
    adapterPelanggan = new AdapterPelanggan(context: MainActivity.this, list);
    adapterPelanggan.notifyDataSetChanged();
    rvPelanggan.setAdapter(adapterPelanggan);
    adapterPelanggan.setOnItemClickListener(onClickListener);

    }else{
        Toast.makeText(context: MainActivity.this, text: ""+message, Toast.LENGTH_SHORT).show();
    }

    } catch (JSONException e) {
        throw new RuntimeException(e);
    }
}

}, new Response.ErrorListener() {
    @Override 2 usages
    public void onErrorResponse(VolleyError volleyError) {
        Log.d(tag: "masnoer", volleyError.toString());
    }
});
requestQueue.add(jsonObjectRequest);
}

```

```

}, new Response.ErrorListener() {
    @Override 2 usages
    public void onErrorResponse(VolleyError volleyError) {
        Log.d(tag: "masnoer", volleyError.toString());
    }
});
requestQueue.add(jsonObjectRequest);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == 200){
        getData();
    }
}
}

```

XML:

Pada layout XML utama (activity_main.xml), kita tambahkan elemen RecyclerView yang berfungsi sebagai tampilan daftar. Kita juga menggunakan file item_pelanggan.xml sebagai template untuk menampilkan detail setiap pelanggan, seperti nama dan alamat.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="2">
        <Button
            android:layout_marginRight="8dp"
            android:layout_weight="1"
            android:layout_marginBottom="16dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Tambah Data"
            android:id="@+id/btnAdd"/>

```

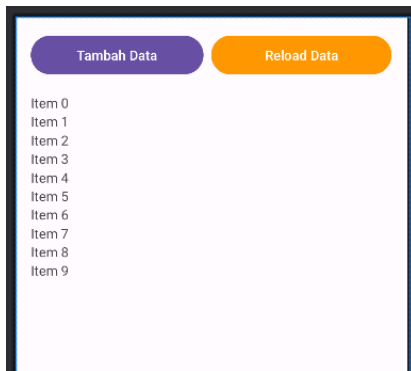
```

        <Button
            android:backgroundTint="#FF9800"
            android:layout_weight="1"
            android:layout_marginBottom="16dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Reload Data"
            android:id="@+id/btnReload"/>
    </LinearLayout>
    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/rvPelanggan"/>

</LinearLayout>

```

Output:



Item_Pelanggan XML:

Layout ini digunakan sebagai template untuk item yang akan diisi dengan data pelanggan dalam RecyclerView

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_margin="12dp"
    android:layout_marginTop="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <androidx.cardview.widget.CardView
        android:id="@+id/cvItemPel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp"
            android:orientation="vertical">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:weightSum="2"
                android:orientation="horizontal">
                <LinearLayout
                    android:layout_weight="1"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">
```



```

        <TextView
            android:id="@+id/tvItemPelNama"
            android:textSize="15sp"
            android:textStyle="bold"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="FULAN"
            android:textColor="@android:color/black"/>
        <TextView
            android:id="@+id/tvItemPelHp"
            android:textSize="12sp"
            android:layout_marginTop="4dp"
            android:textColor="@color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0853xxxx"/>
    </LinearLayout>

</LinearLayout>

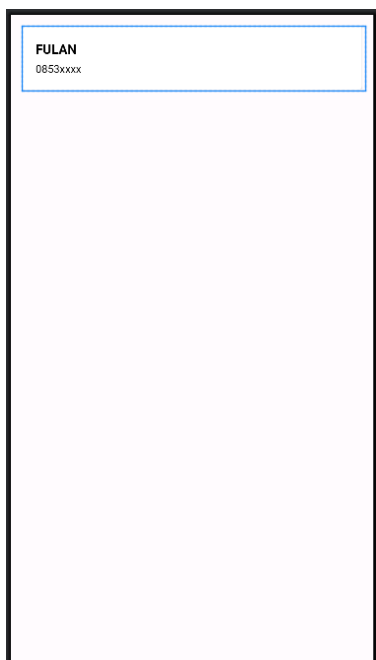
</LinearLayout>

</androidx.cardview.widget.CardView>

</LinearLayout>

```

Output:



D. Manifest

Kita tambahkan kode ke dalam file manifest untuk memastikan izin ini aktif. Pastikan AndroidManifest.xml sudah memiliki izin akses internet agar aplikasi dapat mengakses data dari API.

```
<?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" />

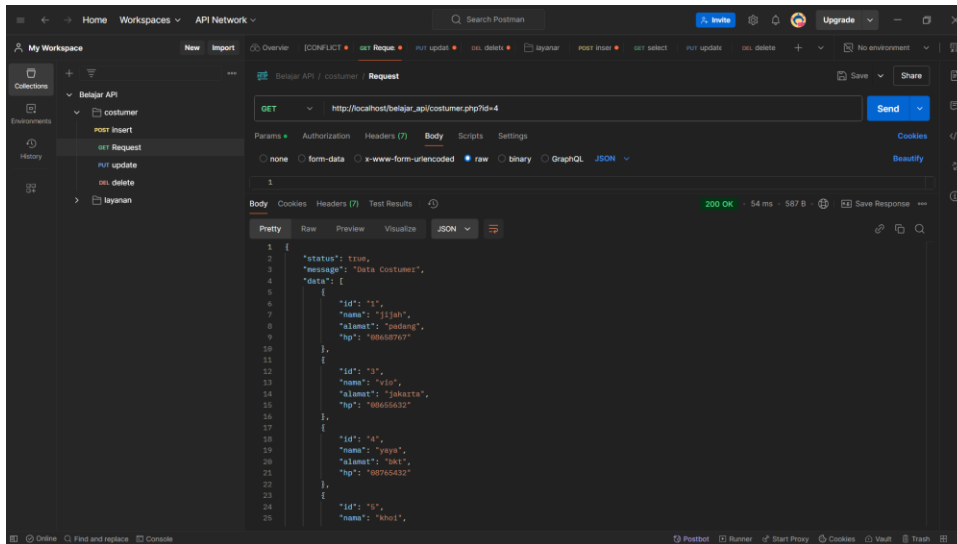
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="ConnectToJson"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="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>
```

E. Data Postman

Buka Postman untuk menguji endpoint API dengan URL http://192.168.43.85/belajar_api/costumer.php. Masukkan URL endpoint API sesuai dengan jaringan perangkat yang digunakan untuk mengakses server, lalu kirim permintaan GET untuk memastikan server mengembalikan data pelanggan dalam format JSON.

Setelah itu, masukkan URL ini ke dalam MainActivity.java. Ini akan memungkinkan aplikasi mengambil data pelanggan dari server dan menampilkannya di RecyclerView



Output:

