

Aplikasi Mobile

“Laporan Praktikum 5”



Oleh :

MUHAMMAD FADHIL HAMDI
(2211533018)

Dosen Pengampu :

NURFIAH, S.ST, M.Kom

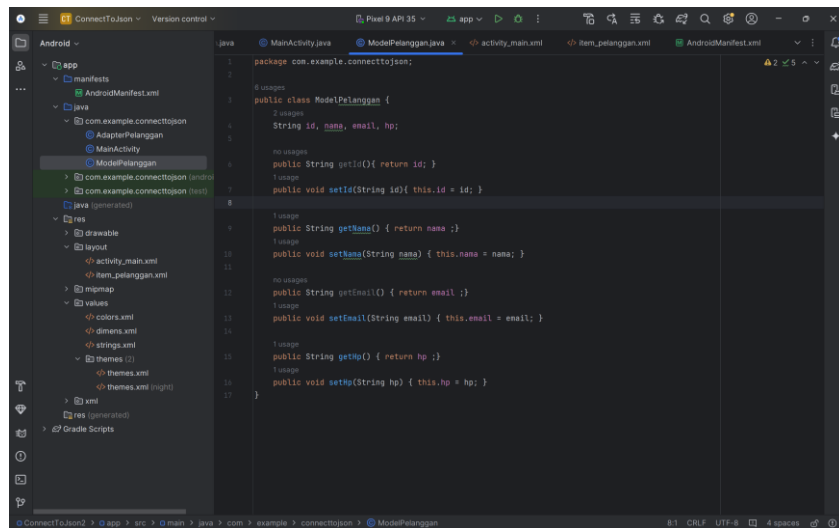
FAKULTAS TEKNOLOGI INFORMASI
DEPERTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
PADANG
2024

Laporan Praktikum 5 (Connect to JSON)

A. Model Pelanggan

- Java

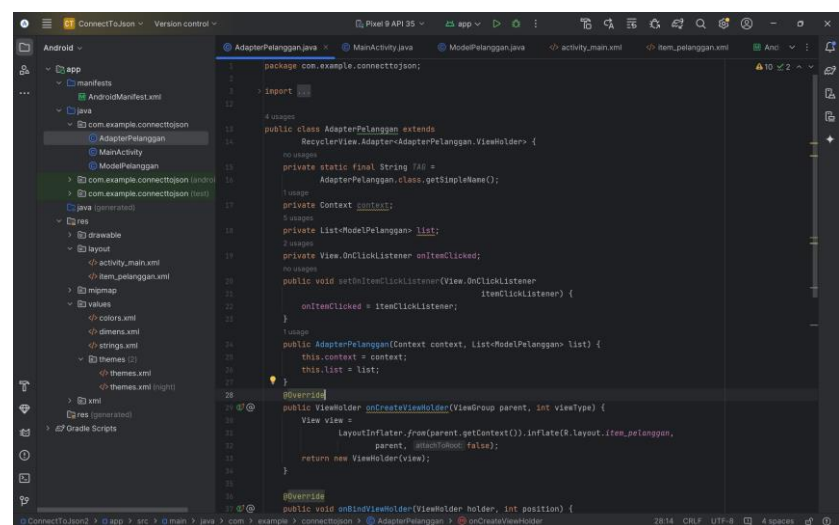
Langkah pertama adalah membuat model data pelanggan. Kita bisa menyalin dari proyek sebelumnya, yaitu laundryapps bagian ModelPelanggan. Di dalam kelas ModelPelanggan, kita definisikan atribut yang sesuai dengan data pelanggan, seperti id, nama, alamat, dan telepon. Setiap atribut ini harus memiliki konstruktor, getter, dan setter untuk memudahkan manipulasi data.



B. Adapter Pelanggan

- Java

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.




```

11 public class MainActivity extends AppCompatActivity {
12
13     private void getData() {
14         JsonObjectRequest jsonObjectRequest = new JsonObjectRequest
15             (Request.Method.GET, base_url, null, new Response.Listener<JSONObject>() {
16                 @Override
17                 public void onResponse(JSONObject jsonObject) {
18                     try {
19                         String status = jsonObject.getString("status");
20                         String message = jsonObject.getString("message");
21                         if (status.equalsIgnoreCase("success")) {
22                             // until json array dont go to data
23                             JSONArray jsonArray = jsonObject.getJSONArray("data");
24                             for (int g=0; g<jsonArray.length(); g++){
25                                 JSONObject jsonObject1 = jsonArray.getJSONObject(g);
26                                 ModelPelanggan mp = new ModelPelanggan();
27                                 mp.setId(jsonObject1.getString("id"));
28                                 mp.setName(jsonObject1.getString("name"));
29                                 mp.setEmail(jsonObject1.getString("email"));
30                                 mp.setHp(jsonObject1.getString("hp"));
31                                 list.add(mp);
32                             }
33                             adapterPelanggan = new AdapterPelanggan(MainActivity.this, list);
34                             adapterPelanggan.notifyDataSetChanged();
35                             rvPelanggan.setAdapter(adapterPelanggan);
36                             // adapterPelanggan.setOnItemClickListener(onClickListener);
37                             Toast.makeText(MainActivity.this, ""+jsonArray.toString()
38                                 , Toast.LENGTH_SHORT).show();
39                         }
40                     } catch (JSONException e) {
41                         throw new RuntimeException(e);
42                     }
43                 }
44             });
45         requestQueue.add(jsonObjectRequest);
46     }
47 }

```

```

11 public class MainActivity extends AppCompatActivity {
12     private void getData() {
13         JsonObjectRequest jsonObjectRequest = new JsonObjectRequest
14             (Request.Method.GET, base_url, null, new Response.Listener<JSONObject>() {
15                 @Override
16                 public void onResponse(JSONObject jsonObject) {
17                     adapterPelanggan = new AdapterPelanggan(MainActivity.this, list);
18                     adapterPelanggan.notifyDataSetChanged();
19                     rvPelanggan.setAdapter(adapterPelanggan);
20                     // adapterPelanggan.setOnItemClickListener(onClickListener);
21                     Toast.makeText(MainActivity.this, ""+jsonArray.toString()
22                         , Toast.LENGTH_SHORT).show();
23                 }
24             }
25         );
26         requestQueue.add(jsonObjectRequest);
27     }
28 }

```

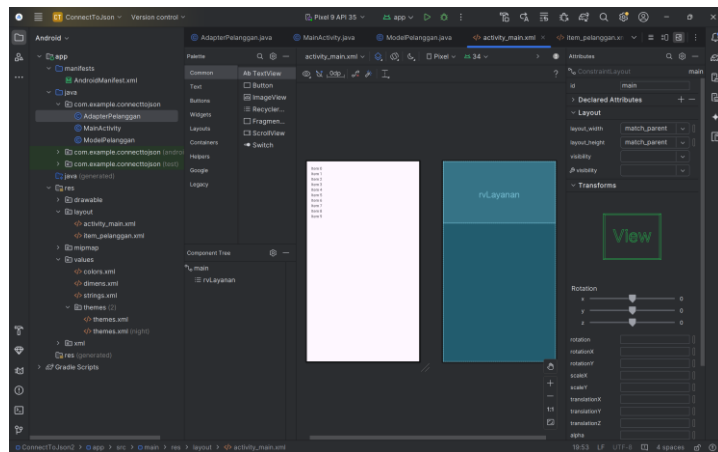
- XML
Pada layout XML utama (activity_main.xml), kita tambahkan elemen RecyclerView yang berfungsi sebagai tampilan daftar. Kita juga menggunakan file item_pelanggan.xml sebagai template untuk menampilkan detail setiap pelanggan, seperti nama dan alamat.

```

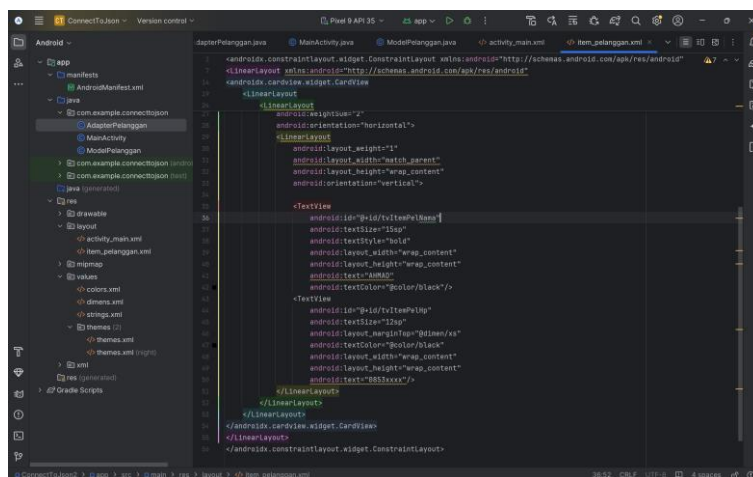
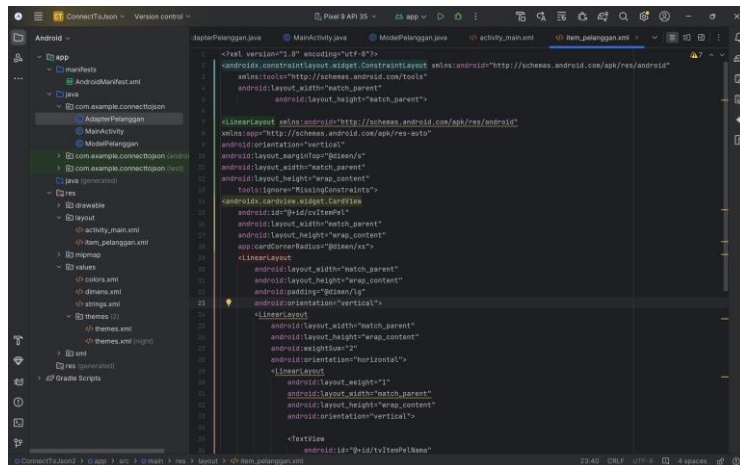
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10     <android.support.design.widget.RecyclerView
11         android:id="@+id/rvPelanggan"
12         android:layout_width="match_parent"
13         android:layout_height="0dp"
14         android:layout_weight="1"
15         android:padding="10dp"
16         tools:ignore="MissingConstraints">
17         </android.support.design.widget.RecyclerView>
18
19     </android.support.design.widget.ConstraintLayout>

```

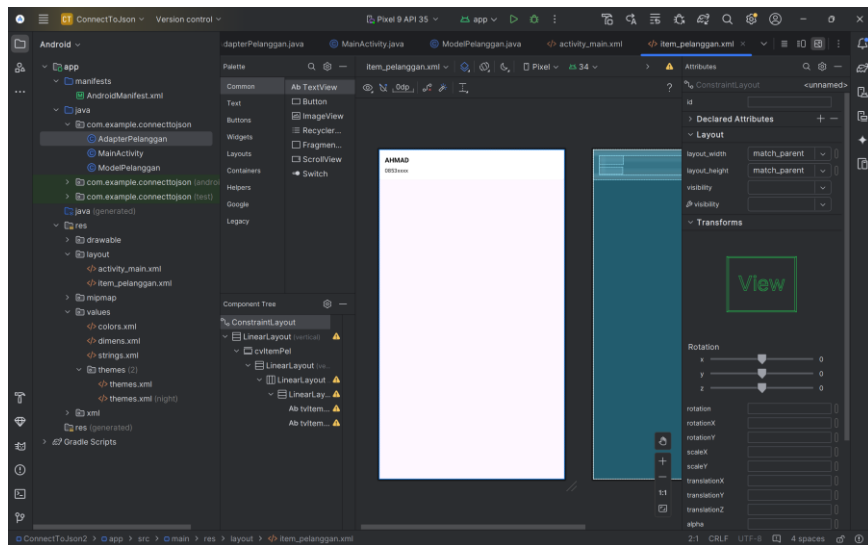
- Output



- Item_pelanggan.xml
Layout ini digunakan sebagai template untuk item yang akan diisi dengan data pelanggan dalam RecyclerView.

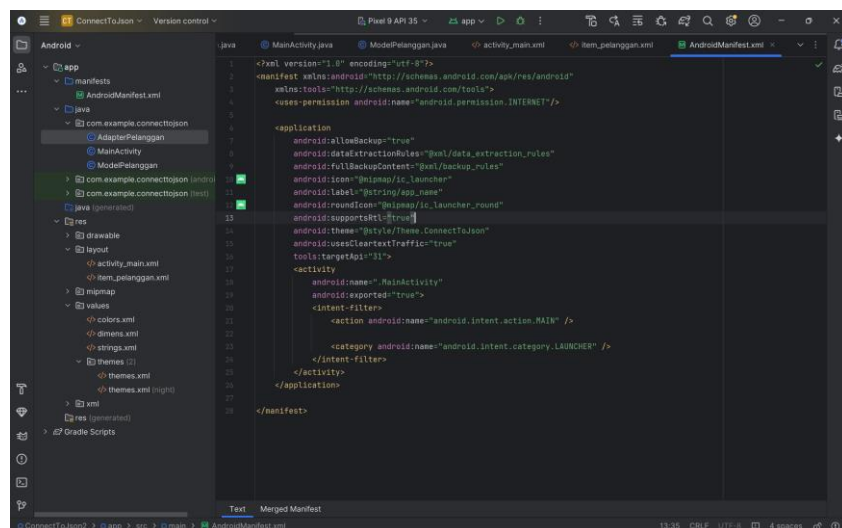


- Output



D. Manifest

Kita tambahkan kode `<uses-permission android:name="android.permission.INTERNET"/>` ke dalam file manifest untuk memastikan izin ini aktif. Pastikan AndroidManifest.xml sudah memiliki izin akses internet agar aplikasi dapat mengakses data dari API.



E. Postman

Buka Postman untuk menguji endpoint API dengan URL `http://192.168.1.17/belajarAPI/api.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.

PrettyRawPreviewVisualizeJSON

```
1 {
2   "status": true,
3   "message": "Data Costumer",
4   "data": [
5     {
6       "id": "1",
7       "nama": "fadhil",
8       "alamat": "medan",
9       "hp": "09872345"
10    },
11    {
12      "id": "5",
13      "nama": "angga",
14      "alamat": "kanopan",
15      "hp": "0877095568"
16    },
17    {
18      "id": "7",
19      "nama": "ibnu",
20      "alamat": "jakarta",
21      "hp": "0898"
22    }
23  ]
24 }
```

Random hardware addresses

Help protect your privacy by making it harder for people to track your device location when takes effect the next time you connect to this network.

IP assignment:	Automatic (DHCP)
DNS server assignment:	Automatic (DHCP)
SSID:	Kost Fadhillah
Protocol:	Wi-Fi 4 (802.11n)
Security type:	WPA2-Personal
Manufacturer:	Intel Corporation
Description:	Intel(R) Wi-Fi 6 AX201 160MHz
Driver version:	22.120.0.3
Network band:	2.4 GHz
Network channel:	11
Link speed (Receive/Transmit):	130/130 (Mbps)
Link-local IPv6 address:	fe80:d9c4:a85d:4c29:c8%12
IPv4 address:	192.168.1.17
IPv4 DNS servers:	118.98.115.69 (Unencrypted) 118.98.115.78 (Unencrypted)
Physical address (MAC):	84-14-4D-B5-62-88

GET

http://192.168.1.17/belajarAPI/api.php

- **Output**
Setelah selesai semuanya, kita jalankan aplikasi di emulator Android (Memu) untuk memeriksa tampilan dan fungsinya. Jika aplikasi berhasil mengambil data dari API, RecyclerView akan menampilkan daftar pelanggan sesuai dengan data JSON yang diterima.

