

Pamungkas Dero Ivano

3SI1

222212813

Penugasan Pemrograman Platform Khusus Modul 12

1. Buatlah laporan tangkapan layar pada kegiatan yang anda lakukan pada 12.4.
 2. Tambahkan toast ketika salah satu data mahasiswa di klik.
-

1. Deskripsi Singkat

RecyclerView adalah komponen yang digunakan untuk menampilkan data dalam jumlah besar secara efisien di aplikasi Android. Dengan mekanisme pendauran ulang tampilan (recycling), RecyclerView mampu meningkatkan performa aplikasi, responsivitas, dan mengurangi konsumsi memori. Praktikum ini bertujuan untuk memahami dan menerapkan penggunaan RecyclerView dan CardView sebagai alternatif yang lebih optimal dibandingkan ListView.

2. Tujuan Praktikum

- Mahasiswa memahami konsep RecyclerView dan CardView pada pemrograman Android.
- Mahasiswa mampu mengimplementasikan RecyclerView untuk menampilkan data secara dinamis.

3. Material dan Alat

- **Perangkat Lunak:** Android Studio

4. Langkah-Langkah Praktikum

1. Membuat Proyek Baru:

Membuat project Android baru di Android Studio bernama *RecyclerView Practice*.

2. Mengatur Layout Utama:

Hapus elemen default TextView pada activity_main.xml.

Tambahkan RecyclerView pada layout dengan ID `recycler_view`.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

3. Membuat Class Mahasiswa.java:

Tambahkan class untuk mendefinisikan model data Mahasiswa yang mencakup atribut nama, nim, dan nohp.

```
public class Mahasiswa {
    private String nama, nim, nohp;

    public Mahasiswa(String nama, String nim, String nohp) {
        this.nama = nama;
        this.nim = nim;
        this.nohp = nohp;
    }

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

    public String getNim() { return nim; }
    public void setNim(String nim) { this.nim = nim; }

    public String getNohp() { return nohp; }
    public void setNohp(String nohp) { this.nohp = nohp; }
}
```

Kotlin:

```
package com.example.recyclerviewpractice

class Mahasiswa(var nama: String, var nim: String, var nohp: String)
```

4. Membuat Layout untuk Item RecyclerView:

Buat file `row_mahasiswa.xml` untuk mendesain tampilan tiap item dengan `CardView` dan `LinearLayout`.

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="6dp"
    app:cardElevation="3dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView android:id="@+id/txt_nama_mahasiswa" android:layout_width="match_parent" an
droid:layout_height="wrap_content" />
        <TextView android:id="@+id/txt_nim_mahasiswa" android:layout_width="match_parent" and
roid:layout_height="wrap_content" />
        <TextView android:id="@+id/txt_nohp_mahasiswa" android:layout_width="match_parent" an
droid:layout_height="wrap_content" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

```

5. Membuat Adapter:

Buat class MahasiswaAdapter.java untuk mengelola data pada RecyclerView.

```

public class MahasiswaAdapter extends RecyclerView.Adapter<MahasiswaAdapter.MahasiswaViewHold
er> {
    private ArrayList<Mahasiswa> dataList;

    public MahasiswaAdapter(ArrayList<Mahasiswa> dataList) {
        this.dataList = dataList;
    }

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

    @Override
    public void onBindViewHolder(MahasiswaViewHolder holder, int position) {
        holder.txtNama.setText(dataList.get(position).getNama());
        holder.txtNim.setText(dataList.get(position).getNim());
        holder.txtNoHp.setText(dataList.get(position).getNoHp());
    }

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

    public static class MahasiswaViewHolder extends RecyclerView.ViewHolder {
        TextView txtNama, txtNim, txtNoHp;

        public MahasiswaViewHolder(View itemView) {
            super(itemView);
            txtNama = itemView.findViewById(R.id.txt_nama_mahasiswa);
            txtNim = itemView.findViewById(R.id.txt_nim_mahasiswa);
            txtNoHp = itemView.findViewById(R.id.txt_nohp_mahasiswa);
        }
    }
}

```

```
}  
}
```

Kotlin:

```
package com.example.recyclerviewpractice  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
import com.example.recyclerviewpractice.MahasiswaAdapter.MahasiswaViewHolder  
  
class MahasiswaAdapter(private val dataList: ArrayList<Mahasiswa>?) : RecyclerView.Adapter<MahasiswaViewHolder>() {  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MahasiswaViewHolder {  
        val inflater =  
            LayoutInflater.from(parent.context)  
        val view =  
            inflater.inflate(  
                R.layout.row_mahasiswa, parent,  
                false  
            )  
        return MahasiswaViewHolder(view)  
    }  
  
    override fun onBindViewHolder(holder: MahasiswaViewHolder, position: Int) {  
        holder.txtNama.text = dataList!![position].nama  
        holder.txtNpm.text = dataList[position].nim  
        holder.txtNoHp.text = dataList[position].nohp  
    }  
  
    override fun getItemCount(): Int {  
        return dataList?.size ?: 0  
    }  
  
    inner class MahasiswaViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
        val txtNama: TextView =  
            itemView.findViewById<View>(R.id.txt_nama_mahasiswa) as TextView  
        val txtNpm: TextView =  
            itemView.findViewById<View>(R.id.txt_nim_mahasiswa) as TextView  
        val txtNoHp: TextView =  
            itemView.findViewById<View>(R.id.txt_nohp_mahasiswa) as TextView  
    }  
}
```

6. Memodifikasi MainActivity.java:

Implementasikan RecyclerView di MainActivity.

```
public class MainActivity extends AppCompatActivity {  
    private RecyclerView recyclerView;  
    private MahasiswaAdapter adapter;  
    private ArrayList<Mahasiswa> mahasiswaArrayList;  
  
    @Override
```

```

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

    addData();

    recyclerView = findViewById(R.id.recycler_view);
    adapter = new MahasiswaAdapter(mahasiswaArrayList);

    RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);
    recyclerView.setAdapter(adapter);
}

private void addData() {
    mahasiswaArrayList = new ArrayList<>();
    mahasiswaArrayList.add(new Mahasiswa("Dimas Maulana", "1414370309", "123456789"));
    mahasiswaArrayList.add(new Mahasiswa("Fadly Yonk", "1214234560", "987654321"));
    // data lainnya..
}
}

```

Kotlin:

```

package com.example.recyclerviewpractice

import android.os.Bundle
import android.view.View
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class MainActivity : AppCompatActivity() {
    private var recyclerView: RecyclerView? = null
    private var adapter: MahasiswaAdapter? = null
    private var mahasiswaArrayList: ArrayList<Mahasiswa>? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        addData()
        recyclerView = findViewById<View>(R.id.recycler_view) as RecyclerView
        adapter = MahasiswaAdapter(mahasiswaArrayList)
        val layoutManager: RecyclerView.LayoutManager = LinearLayoutManager(this@MainActivit
y)

        recyclerView!!.layoutManager = layoutManager
        recyclerView!!.adapter = adapter
    }

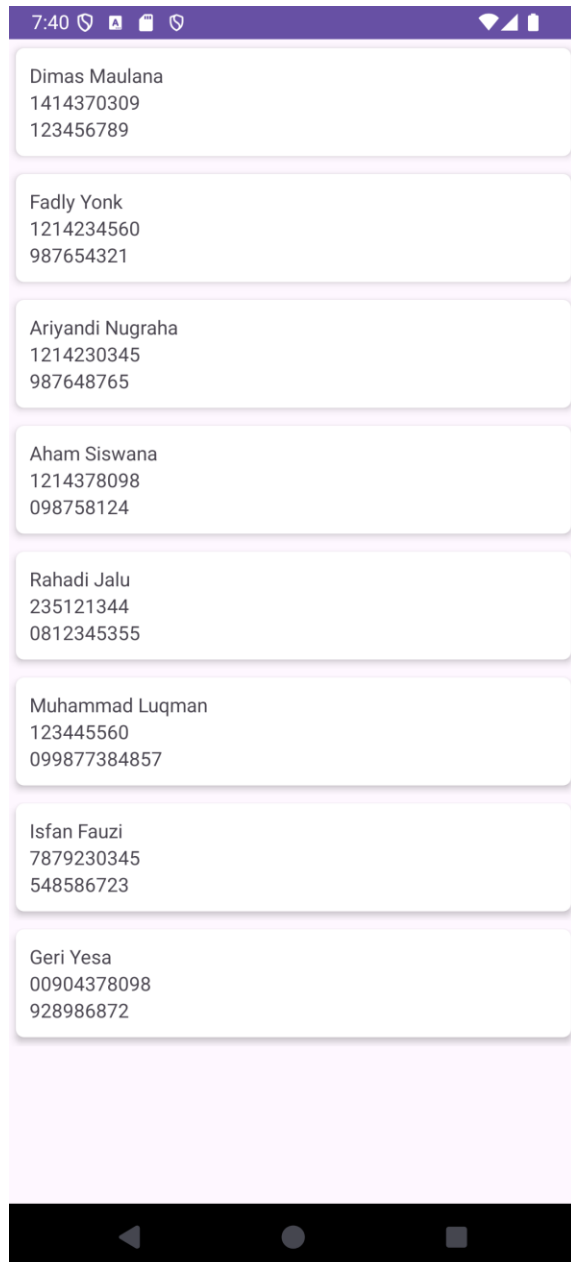
    fun addData() {
        mahasiswaArrayList = ArrayList()
        mahasiswaArrayList!!.add(Mahasiswa("Dimas Maulana", "1414370309", "123456789"))
        mahasiswaArrayList!!.add(Mahasiswa("Fadly Yonk", "1214234560", "987654321"))
        mahasiswaArrayList!!.add(Mahasiswa("Ariyandi Nugraha", "1214230345", "987648765"))
        mahasiswaArrayList!!.add(Mahasiswa("Aham Siswana", "1214378098", "098758124"))
        mahasiswaArrayList!!.add(Mahasiswa("Rahadi Jalu", "235121344", "0812345355"))
        mahasiswaArrayList!!.add(Mahasiswa("Muhammad Luqman", "123445560", "099877384857"))
        mahasiswaArrayList!!.add(Mahasiswa("Isfan Fauzi", "7879230345", "548586723"))
        mahasiswaArrayList!!.add(Mahasiswa("Geri Yesa", "00904378098", "928986872"))
    }
}

```

```
}  
}
```

5. Hasil Praktikum

Ketika aplikasi dijalankan, tampilan daftar mahasiswa akan ditampilkan menggunakan RecyclerView. Setiap item berisi nama, NIM, dan nomor HP mahasiswa.



Penugasan 2.1: Modifikasi MahasiswaAdapter

Tambahkan `setOnClickListener` pada `itemView` di `MahasiswaViewHolder`. Untuk menampilkan `Toast`, gunakan `itemView.context`.

```
class MahasiswaAdapter(private val dataList: ArrayList<Mahasiswa>?) : RecyclerView.Adapter<MahasiswaAdapter.MahasiswaViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MahasiswaViewHolder {
        val inflater = LayoutInflater.from(parent.context)
        val view = inflater.inflate(R.layout.row_mahasiswa, parent, false)
        return MahasiswaViewHolder(view)
    }

    override fun onBindViewHolder(holder: MahasiswaViewHolder, position: Int) {
        val mahasiswa = dataList!![position]
        holder.txtNama.text = mahasiswa.nama
        holder.txtNpm.text = mahasiswa.nim
        holder.txtNoHp.text = mahasiswa.nohp

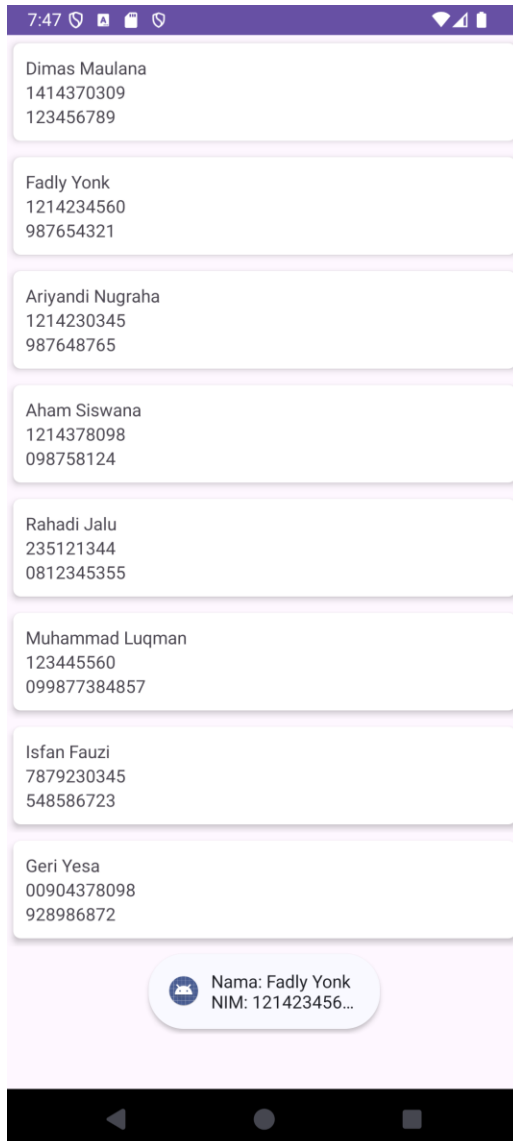
        // Tambahkan OnClickListener untuk item
        holder.itemView.setOnClickListener {
            val context = holder.itemView.context
            Toast.makeText(
                context,
                "Nama: ${mahasiswa.nama}\nNIM: ${mahasiswa.nim}\nNo HP: ${mahasiswa.nohp}",
                Toast.LENGTH_SHORT
            ).show()
        }
    }

    override fun getItemCount(): Int {
        return dataList?.size ?: 0
    }

    inner class MahasiswaViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val txtNama: TextView = itemView.findViewById(R.id.txt_nama_mahasiswa)
        val txtNpm: TextView = itemView.findViewById(R.id.txt_nim_mahasiswa)
        val txtNoHp: TextView = itemView.findViewById(R.id.txt_nohp_mahasiswa)
    }
}
```

Hasil

1. Saat aplikasi dijalankan, `RecyclerView` akan menampilkan daftar mahasiswa.
2. Ketika salah satu item di klik, `Toast` akan muncul menampilkan:
 - o Nama
 - o NIM
 - o No HP



6. Kesimpulan

- RecyclerView menawarkan performa yang lebih baik dibandingkan ListView melalui mekanisme daur ulang elemen.
- Dengan memahami konsep ini, pengelolaan data dalam jumlah besar di aplikasi Android menjadi lebih efisien.