

**LAPORAN RESMI**  
**MODUL II**  
**LAYOUT, WIDGET VIEW, RECYCLER VIEW**  
**PEMROGRAMAN BERGERAK**



<b>NAMA</b>	<b>: SEPTIYA YUTANTRI</b>
<b>N.R.P</b>	<b>: 200441100023</b>
<b>DOSEn</b>	<b>: ACHMAD DAFID, S.T., M.T</b>
<b>ASISTEN</b>	<b>: MUHAMMAD YAFIE ANWARY RAHMAN</b>
<b>TGL PRAKTIKUM</b>	<b>: 31 MARET 2023</b>

**Disetujui : 07 April 2023**  
**Asisten**

**M. YAFIE ANWARY RAHMAN**  
**190441100052**



**LABORATORIUM BISNIS INTELIJEN SISTEM**  
**PRODI SISTEM INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di era sekarang perkembangan teknologi sudah meningkat, manusia sudah tidak bisa dipisahkan dengan yang namanya teknologi. Untuk saat ini teknologi menjadi kebutuhan banyak orang. Dunia IT berkembang begitu pesat karena ditunjang dengan adanya perkembangan teknologi yang semakin canggih dan modern, pemanfaatan mobile learning yang dirasa perlu untuk menunjang proses belajar mengajar mahasiswa di pembelajaran Pemrograman Objek dan Perangkat Bergerak. Mobile Learning berbasis android dipilih mengingat hampir seluruh siswa di kelas memiliki handphone. Diharapkan dengan adanya mobile learning mampu meningkatkan semangat belajar karena didukung oleh teknologi yang bagus.

Pembelajaran praktikum ini mempelajari tentang layout, widget view, dan recycler view dimana ketiga komponen tersebut merupakan dasar antar muka dari sebuah aplikasi, dengan begitu kita dapat membangun sebuah aplikasi android dengan menggunakan Kotlin. Prasyarat untuk dapat mengikuti praktikum ini dengan baik adalah memiliki pengetahuan dalam Bahasa pemrograman berorientasi objek penuh seperti java, C++, dan juga Kotlin. Untuk pengembangan, disarankan menggunakan sumber referensi selain modul praktikum ini, sehingga kita bisa belajar lebih mendalam tentang materi yang terdapat di modul dari sumber lain.

### **1.2 Tujuan**

- Membuat Layout dengan Linear Layout dan Constraint Layout
- Mampu menggunakan Widget View (masukan) untuk membuat aplikasi sederhana
- Merepresentasikan data dengan menggunakan komponen recyclerview

## **BAB II**

### **DASAR TEORI**

#### **2.1 Layout**

Pada dasarnya semua elemen antar pengguna di aplikasi Android dibangun menggunakan dua buah komponen inti, yaitu view dan viewgroup. Sebuah view adalah obyek yang menggambar komponen tampilan ke layar yang mana pengguna dapat melihat dan berinteraksi langsung.

Contoh komponen turunan dari view seperti :

- TextView, komponen yang berguna untuk menampilkan teks ke layar.
- Button, komponen yang membuat pengguna dapat berinteraksi dengan cara ditekan untuk melakukan sesuatu.
- ImageView, Komponen untuk menampilkan gambar.
- ListView, komponen untuk menampilkan informasi dalam bentuk list.
- GridView, komponen untuk menampilkan informasi dalam bentuk grid.
- RadioButton, komponen yang memungkinkan pengguna dapat memilih satu pilihan dari berbagai pilihan yang disediakan.
- Checkbox, komponen yang memungkinkan pengguna dapat memilih lebih dari satu pilihan yang ada.

Sedangkan viewgroup adalah sebuah obyek yang mewadahi obyek-obyek view dan viewgroup itu sendiri sehingga membentuk satu kesatuan tampilan aplikasi yang utuh.

Contoh komponen viewgroup adalah :

- Linear Layout
- Frame Layout
- Relative Layout
- Table Layout

Jika diterjemahkan di dalam sebuah viewgroup akan ditampung dua buah komponen view dan satu komponen viewgroup yang terdiri dari 3 buah komponen view.

Obyek turunan viewgroup LinearLayout menjadi kontainer untuk obyek turunan view, button, dan textview. Beberapa komponen viewgroup seperti linearlayout, relativelayout, framelayout, dan tablelayout merupakan komponen

yang paling banyak digunakan untuk menjadi *parent/root* dari komponen-komponen view.

Berikut adalah definisi singkat dan inti dari komponen-komponen di atas terhadap penempatan komponen view (*child*) di dalamnya. Kita akan membahas Linear Layout dan Constrain Layout.

➤ **LinearLayout**

Layout ini akan menempatkan komponen-komponen di dalamnya secara horizontal atau vertikal. LinearLayout memiliki atribut *weight* untuk masing-masing *child* view yang berguna untuk menentukan porsi ukuran view dalam sebuah ruang (*space*) yang tersedia.

➤ **Constraint Layout**

Constraint Layout merupakan salah satu komponen ViewGroup yang dapat kita gunakan untuk menyusun tampilan aplikasi yang kompleks tanpa adanya nested layout. ConstraintLayout tersedia dengan dukungan kompatibilitas mulai dari Android 2.3 (API Level 9) sampai dengan yang terbaru.

ConstraintLayout memiliki kesamaan dengan RelativeLayout. Dalam penggunaan semua view yang berada di dalamnya disusun berhubungan antara parent dan view lainnya. Tapi ConstraintLayout lebih fleksibel dari RelativeLayout dan mudah digunakan dengan dukungan Layout Editor pada Android Studio.

## 2.2 Widget View

Widget pada dasarnya merupakan visualisasi dari elemen user interface (UI) yang digunakan pada layar aplikasi Android di mana kita dapat merancang sendiri sesuai kebutuhan. Widget di dalam Android ditampilkan dengan konsep View. Di mana aplikasi Android pada umumnya menggunakan widget sebagai Layout XML. Untuk mengimplementasikan widget, selain file kotlin kita juga membutuhkan tambahan dua file. Berikut ini adalah file-file yang umumnya kita butuhkan apabila kita membuat widget:

- 1) File Kotlin. Berupa file yang mengimplementasikan aksi dari widget. Jika kita mendefinisikan suatu widget beserta posisinya di layar yang

didefinisikan dari file XML, kita harus melakukan coding di file kotlin yang dapat mengambil semua nilai atribut dari file layout XML yang didefinisikan.

- 2) File XML. Sebuah file yang mendefinisikan komponen elemen-elemen XML yang digunakan untuk inisialisasi widget serta atribut yang mendukungnya.
- 3) Layout XML. File XML menggambarkan atau penambahan keterangan pada layout widget kita.

Komponen widget TextView dan Button sudah kita bahas pada modul sebelumnya. Beberapa komponen widget akan kita bahas saat ini. Widget EditText untuk menuliskan teks ke aplikasi dan akan ditangkap oleh aplikasi untuk diolah. Widget Image Button untuk membuat button yang diberi gambar. Widget Image View untuk membuat tampilan gambar. Sedangkan widget RadioButton/ RadioGroup biasanya digunakan bersama-sama.

Di dalam satu RadioGroup terdapat beberapa RadioButton. Dan di dalam satu RadioGroup user hanya dapat melakukan satu check/pemilihan RadioButton. Dan yang terakhir widget akan kita bahas CheckBox, pilihan yang dapat dipilih lebih dari satu item.

#### ❖ **Event Handling.**

Android dapat menangani event dari interaksi dengan pengguna. Saat mempertimbangkan event dalam user interface, pendekatannya adalah menangkap event dari objek View tertentu yang digunakan pengguna untuk berinteraksi. Kelas View menyediakan sarana untuk melakukannya.

Dalam berbagai kelas View yang akan digunakan untuk menyusun layout, mungkin dapat dilihat beberapa method callback publik yang tampak berguna untuk kejadian UI. Method ini dipanggil oleh framework Android ketika masing-masing tindakan terjadi pada objek itu. Misalnya, jika View (seperti Button) disentuh, method onTouchEvent() akan dipanggil pada objek itu. Kelas View salah satunya berisi sekumpulan interface bertumpuk dengan callback yang mudah didefinisikan. Antarmuka ini, yang disebut event listener, digunakan untuk melakukan interaksi pengguna dengan UI.

#### ❖ **Event listener**

Event listener merupakan antarmuka di kelas [View](#) yang berisi method callback tunggal. Method ini akan dipanggil oleh framework Android jika View yang telah didaftarkan dengan listener dipicu oleh interaksi pengguna dengan item dalam UI. Yang juga disertakan dalam antarmuka event listener adalah method callback berikut ini :

- 1) Method `onClick()` dari [View.OnClickListener](#). Ini dipanggil baik saat pengguna menyentuh item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan menekan tombol "enter" yang sesuai atau menekan trackball.
- 2) Method `onLongClick()` dari [View.OnLongClickListener](#). Ini dipanggil baik saat pengguna menyentuh dan menahan item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan menekan serta menahan tombol "enter" yang sesuai atau menekan dan menahan trackball (selama satu detik).
- 3) Method `onFocusChange()` dari [View.OnFocusChangeListener](#). Ini dipanggil saat pengguna menyusuri ke atau dari item, dengan menggunakan tombol navigasi atau trackball.
- 4) Method `onKey()` dari [View.OnKeyListener](#). Ini dipanggil saat pengguna memfokuskan pada item dan menekan atau melepas tombol perangkat keras pada perangkat.
- 5) Method `onTouch()` dari [View.OnTouchListener](#). Ini dipanggil saat pengguna melakukan tindakan yang digolongkan sebagai peristiwa sentuh, termasuk penekanan, pelepasan, atau isyarat perpindahan pada layar (dalam batasan item itu).
- 6) Method `onCreateContextMenu()` dari [View.OnCreateContextMenuListener](#). Ini dipanggil saat Menu Konteks sedang dibuat (akibat "klik lama" terus-menerus).

## 2.3 Recycler View

RecyclerView adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan UI controller, ViewModel, dan LiveData. Menampilkan list atau grid data adalah salah satu tugas UI paling umum di Android. Daftar

bervariasi dari yang sederhana hingga yang sangat kompleks. Daftar tampilan teks mungkin menampilkan data sederhana, seperti daftar belanja. Daftar yang kompleks, seperti daftar tujuan liburan yang berannotasi, dapat menunjukkan kepada pengguna banyak detail di dalam scrolling grid dengan header. Untuk mendukung semua kasus penggunaan ini, Android menyediakan widget RecyclerView.

Manfaat terbesar dari RecyclerView adalah sangat efisien untuk daftar besar:

- Secara default, RecyclerView hanya berfungsi untuk memproses atau menggambar item yang saat ini terlihat di layar. Misalnya, jika list memiliki seribu elemen tetapi hanya 10 elemen yang terlihat, RecyclerView hanya berfungsi untuk menggambar 10 item di layar. Ketika pengguna melakukan scroll, RecyclerView mengetahui item baru apa yang seharusnya ada di layar dan tidak cukup berfungsi untuk menampilkan item itu.
- Ketika suatu item scroll dari layar, tampilan item tersebut didaur ulang. Itu berarti item diisi dengan konten baru yang scroll ke layar. Perilaku RecyclerView ini menghemat banyak waktu pemrosesan dan membantu scroll list dengan lancar.
- Ketika suatu item berubah, alih-alih menggambar ulang seluruh daftar, RecyclerView dapat memperbarui satu item itu. Ini adalah keuntungan efisiensi yang sangat besar ketika menampilkan daftar item kompleks!

Dalam urutan yang ditunjukkan di bawah ini, kita dapat melihat bahwa satu tampilan telah diisi dengan data, ABC. Setelah itu tampilan bergulir dari layar, RecyclerView menggunakan kembali tampilan untuk data baru, XYZ.

Pola adaptor dalam rekayasa perangkat lunak membantu objek bekerja dengan API lain. RecyclerView menggunakan adaptor untuk mengubah data aplikasi menjadi sesuatu yang dapat ditampilkan RecyclerView, tanpa mengubah cara aplikasi menyimpan dan memproses data. Untuk aplikasi pelacak tidur, kita membuat adaptor yang mengadaptasi data menjadi sesuatu yang RecyclerView tahu cara menampilkannya, tanpa mengubah ViewModel.

Untuk menampilkan data dalam RecyclerView, memerlukan bagian-bagian berikut:

- 1) Data untuk ditampilkan.
- 2) Mesin virtual RecyclerView didefinisikan dalam file layout, untuk bertindak sebagai wadah untuk tampilan.
- 3) Layout untuk satu item data.

Jika semua item list terlihat sama, kita dapat menggunakan layout yang sama untuk semuanya, tetapi itu tidak wajib. Layout item harus dibuat secara terpisah dari layout fragmen, sehingga tampilan satu item pada satu waktu dapat dibuat dan diisi dengan data.

- 4) Layout Manager.

Layout Manager menangani organisasi (layout) komponen UI dalam tampilan.

- 5) View holder.

View holder extends kelas ViewHolder. Ini berisi informasi tampilan untuk menampilkan satu item dari layout item. Penampil tampilan juga menambahkan informasi yang digunakan RecyclerView untuk memindahkan tampilan di layar secara efisien.

- 6) Adaptor.

Adaptor menghubungkan data kita ke RecyclerView. Ini menyesuaikan data sehingga dapat ditampilkan di ViewHolder. RecyclerView menggunakan adaptor untuk mengetahui cara menampilkan data di layar.



## BAB III

### TUGAS PENDAHULUAN

#### 3.1 TuPen1

##### a. Source Code

###### - activity\_main

```
<?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">
<TextView
    android:id="@+id/textView"
    android:layout_width="390dp"
    android:layout_height="43dp"
    android:layout_marginStart="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp"
    android:text="SATU"
    android:textAlignment="center"
    android:textColor="#000000"
    android:textSize="24sp"
    android:textStyle="bold|italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="186dp"
    android:layout_height="68dp"
    android:layout_marginStart="4dp"
    android:layout_marginTop="24dp"
    android:text="DUA"
    android:textAlignment="center"
    android:textColor="#000000"
    android:textSize="34sp"
    android:textStyle="bold"

    app:layout_constraintStart_toStartOf="@+id/textView"

    app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="155dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="TIGA"
    android:textAlignment="center"
    android:textColor="#000000"
    android:textSize="24sp"
```

```

        android:textStyle="bold|italic"

app:layout_constraintEnd_toEndOf="@+id/textView"

app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="175dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="EMPAT"
    android:textAlignment="center"
    android:textColor="#000000"
    android:textSize="24sp"
    android:textStyle="bold|italic"

app:layout_constraintEnd_toEndOf="@+id/textView4"

app:layout_constraintTop_toBottomOf="@+id/textView4" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

## - MainActivity

```

package com.example.tupen1

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}

```

## b. Hasil Run



## 3.2 TuPen2

### a. Source Code

- activity\_main

```
<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">

    <TextView
        android:id="@+id/textnama"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="24dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginStart="24dp"/>

    <TextView
        android:id="@+id/textgender"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gender :"
        android:textSize="20sp"

        app:layout_constraintTop_toBottomOf="@+id/textnama"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginTop="32dp"
        android:textStyle="bold"/>

    <TextView
        android:text="Hobi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txtHobi"

        app:layout_constraintTop_toBottomOf="@+id/textgender"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:textSize="20sp"
        android:textStyle="bold"/>

    <EditText
        android:id="@+id/editnama"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:layout_marginTop="24dp"
```

```

        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="24dp"
        app:layout_constraintEnd_toEndOf="parent">

</EditText>
<RadioGroup
    android:layout_width="0dp"
    android:layout_height="wrap_content"

app:layout_constraintTop_toBottomOf="@+id/editnama"

app:layout_constraintStart_toEndOf="@+id/textgender"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"
    android:layout_marginTop="16dp"
    android:layout_marginStart="8dp"
    android:orientation="horizontal"
    android:id="@+id/radioGroup">
    <RadioButton
        android:text="PRIA"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/rbPria"
        android:layout_weight="1"
        android:textSize="16sp"
        android:textStyle="bold"/>
    <RadioButton
        android:text="WANITA"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/rbWanita"
        android:layout_weight="1"
        android:textSize="16sp"
        android:textStyle="bold"/>
</RadioGroup>

<CheckBox
    android:text="Coding"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:id="@+id/cbCoding"

app:layout_constraintStart_toEndOf="@+id/txtHobi"
    android:layout_marginStart="24dp"
    android:layout_marginTop="16dp"

app:layout_constraintTop_toBottomOf="@+id/radioGroup"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"
    android:textSize="18sp"
    android:textStyle="bold"/>
<CheckBox
    android:text="Reading"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:id="@+id/Reading"
    android:layout_marginTop="8dp"

```

```

app:layout_constraintTop_toBottomOf="@+id/cbCoding"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"

app:layout_constraintStart_toEndOf="@+id/txtHobi"
    android:layout_marginStart="24dp"
    android:textSize="18dp"
    android:textStyle="bold"/>
<CheckBox
    android:text="Traveling"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:id="@+id/cbTraveling"
    android:layout_marginEnd="24dp"

app:layout_constraintTop_toBottomOf="@+id/Reading"
    app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toEndOf="@+id/txtHobi"
    android:layout_marginStart="24dp"
    android:layout_marginTop="8dp"
    android:textSize="16sp"
    android:textStyle="bold"/>

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:srcCompat="@android:drawable/ic_delete"
    android:id="@+id/imgButtonCancel"
    android:layout_marginTop="32dp"

app:layout_constraintTop_toBottomOf="@+id/cbTraveling"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:srcCompat="@android:drawable/ic_input_add"
    android:id="@+id/imgButtonAdd"

app:layout_constraintEnd_toStartOf="@+id/imgButtonCancel"
    android:layout_marginEnd="24dp"
    android:layout_marginTop="32dp"

app:layout_constraintTop_toBottomOf="@+id/cbTraveling"/
>

</androidx.constraintlayout.widget.ConstraintLayout>

```

## - MainActivity

```

package com.example.tupen2

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

```

```

import android.widget.ImageButton
import android.widget.Toast

class MainActivity : AppCompatActivity() {

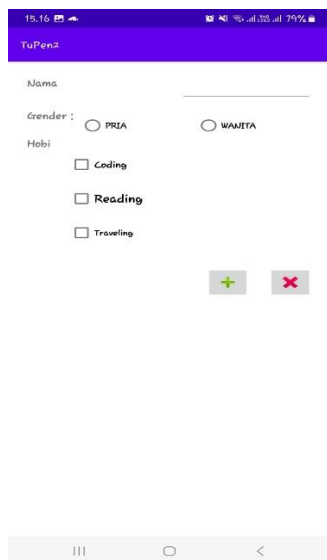
    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        var
imgButtonAdd=findViewById<ImageButton>(R.id.imgButtonAdd)

        imgButtonAdd.setOnClickListener {
            Toast.makeText(this, "Data Diinput",
Toast.LENGTH_LONG).show()
        }
    }
}

```

## b. Hasil Run



## 3.3 TuPen3

### a. Source Code

#### - MainActivity

```

package com.example.tupen3

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.tprv.MyAdapter

class MainActivity : AppCompatActivity() {

    lateinit var recyclerView: RecyclerView

```

```

        lateinit var adapter: MyAdapter
        override fun onCreate(savedInstanceState: Bundle?)
        {
            super.onCreate(savedInstanceState)
            setContentView(R.layout.activity_main)
            init()
            recyclerView.layoutManager =
                LinearLayoutManager(this)
            recyclerView.adapter=adapter
        }

        private fun init() {
            recyclerView = findViewById(R.id.recycler_view)

            var data = ArrayList<tprv>()
            data.add(tprv(R.drawable.baseline_person_24,
"Holaw", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Nisa", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Sabyan", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Lola", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Noa", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Huski", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"ruhiw", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Ruhie", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Holaw", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Nisa", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Sabyan", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Lola", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Noa", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Huski", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"ruhiw", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Ruhie", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Holaw", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Nisa", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Sabyan", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Lola", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Noa", "haloo"))
            data.add(tprv(R.drawable.baseline_person_24,
"Huski", "haloo"))

```

```

        data.add(tprv(R.drawable.baseline_person_24,
"ruhiw", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Ruhie", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Holaw", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Nisa", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Sabyan", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Lola", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Noa", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Huski", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"ruhiw", "haloo"))
        data.add(tprv(R.drawable.baseline_person_24,
"Ruhie", "haloo"))

        adapter=MyAdapter(data)
    }
}

```

#### - MyAdapter

```

package com.example.tprv

import android.view.LayoutInflater
import android.view.ViewGroup
import android.widget.Toast
import androidx.recyclerview.widget.RecyclerView

class MyAdapter (private val data: ArrayList<tprv>):
RecyclerView.Adapter<ViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): ViewHolder {
        val inflater: LayoutInflater =
LayoutInflater.from(parent.context)
        return ViewHolder(inflater, parent)
    }

    override fun getItemCount(): Int {
        return data.size
    }

    override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
        val name = data[position]
        holder.bind(data[position])
        holder.itemView.setOnClickListener {
            Toast.makeText(holder.itemView.context,
"${data[position].txtTitle}",
Toast.LENGTH_SHORT).show()
            Toast.makeText(holder.itemView.context,
"${data[position].txtsubtitle}",
Toast.LENGTH_SHORT).show()

```



```

    }
}

```

### - Tupen3

```

package com.example.tupen3

data class tprv (val imgView: Int, val txtTitle:
String, val txtsubtitle: String)

```

### - ViewHolder

```

package com.example.tprv

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class ViewHolder(inflater: LayoutInflater, parent:
ViewGroup):
RecyclerView.ViewHolder(inflater.inflate(R.layout.rv_it
em, parent, false)) {
private var imgView: ImageView? = null
    private var txt_view: TextView? = null
    private var subtitle: TextView? = null

    init {
        imgView = itemView.findViewById(R.id.img_view)
        txt_view=itemView.findViewById(R.id.txt_title)

        subtitle=itemView.findViewById(R.id.txt_sub_title)

    }
fun bind(data: tprv){
    imgView?.setImageResource(data.imgView)
    txt_view?.text=data.txtTitle
    subtitle?.text=data.txtsubtitle
}
}

```

### - activity\_main

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">
<androidx.recyclerview.widget.RecyclerView

```

```

        android:id="@+id/recycler_view"
        android:layout_width="match_parent"

        android:layout_height="match_parent"></androidx.recyclerview.widget.RecyclerView>

```

```

</LinearLayout>

```

#### - rv\_item

```

- <?xml version="1.0" encoding="utf-8"?>
  <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:orientation="horizontal">

    <ImageView
        android:id="@+id/img_view"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:layout_marginRight="8dp"

        android:src="@drawable/baseline_person_24"></ImageView>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/txt_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Title"
            android:textColor="#F8F0F0"
            android:textSize="20sp"
            android:textStyle="bold"></TextView>
        <TextView
            android:id="@+id/txt_sub_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="subtitle"
            android:textColor="#E8E5E5"></TextView>
    </LinearLayout>

  </LinearLayout>

```

#### b. Hasil Run

## BAB IV

### IMPLEMENTASI

#### 4.1 RecyclerView

##### 4.1.1 Source Code

###### a. DetailActivity

```
package com.example.praktikum2

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ImageView
import android.widget.TextView

class DetailActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)

        val pahlawan =
            intent?.getParcelableExtra<Pahlawan>("Pahlawan")
        if (pahlawan != null) {
            val textView : TextView =
                findViewById(R.id.detailActivityTv)
            val imageView : ImageView =
                findViewById(R.id.detailActivityIv)

            textView.text = pahlawan.name
            imageView.setImageResource(pahlawan.image)
        }
    }
}
```

###### b. MainActivity

```
package com.example.praktikum2

import android.annotation.SuppressLint
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import java.util.*
import kotlin.collections.ArrayList

class MainActivity : AppCompatActivity() {

    private lateinit var recyclerView: RecyclerView
    private lateinit var PahlawanList: ArrayList<Pahlawan>
    private lateinit var PahlawanAdapter: PahlawanAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
```

```

        recyclerView = findViewById(R.id.recyclerView)
        recyclerView.setHasFixedSize(true)
        recyclerView.layoutManager=
LinearLayoutManager(this)

        PahlawanList = ArrayList()

PahlawanList.add(Pahlawan(R.drawable.aminoto,"H.Oemar Said
Tjokroaminoto"))

PahlawanList.add(Pahlawan(R.drawable.bonjol,"Tuanku Imam
Bonjol"))
        PahlawanList.add(Pahlawan(R.drawable.dewi,"Raden
Dewi Sartika"))
        PahlawanList.add(Pahlawan(R.drawable.dinata,"Otto
Iskandar Dinata"))

PahlawanList.add(Pahlawan(R.drawable.diponegoro,"Pangeran
Diponegoro"))

PahlawanList.add(Pahlawan(R.drawable.jenderal,"Jenderal
Soedirman"))

PahlawanList.add(Pahlawan(R.drawable.kapitan,"Kapitan
Pattimura"))
        PahlawanList.add(Pahlawan(R.drawable.malaka,"Tan
Malaka"))
        PahlawanList.add(Pahlawan(R.drawable.nyak,"Tjut
Nyak Dhien"))

PahlawanList.add(Pahlawan(R.drawable.pangeran,"Pangeran
Antasari"))
        PahlawanList.add(Pahlawan(R.drawable.soetomo,"Dr.
Soetomo"))

PahlawanList.add(Pahlawan(R.drawable.sultan,"Sultan
Hasanuddin"))
        PahlawanList.add(Pahlawan(R.drawable.wr,"W.R
Soepratman"))
        PahlawanList.add(Pahlawan(R.drawable.wahid,"K.H.
Abdul Wahid Hasjim"))
        PahlawanList.add(Pahlawan(R.drawable.yani,"Achmad
Yani"))

        PahlawanAdapter= PahlawanAdapter(PahlawanList)
        recyclerView.adapter = PahlawanAdapter

        PahlawanAdapter.onItemClick= {
            val intent=Intent(this,
DetailActivity::class.java)
            intent.putExtra("Pahlawan", it)
            startActivity(intent)
        }
    }
}

```

### c. Pahlawan

```
package com.example.praktikum2

import android.os.Parcel
import android.os.Parcelable

data class Pahlawan(val image:Int, val name:String) :
Parcelable{
    constructor(parcel: Parcel) : this(
        parcel.readInt(),
        parcel.readString()!!
    ) {
    }

    override fun writeToParcel(parcel: Parcel, flags: Int)
    {
        parcel.writeInt(image)
        parcel.writeString(name)
    }

    override fun describeContents(): Int {
        return 0
    }

    companion object CREATOR :
Parcelable.Creator<Pahlawan> {
        override fun createFromParcel(parcel: Parcel):
Pahlawan {
            return Pahlawan(parcel)
        }

        override fun newArray(size: Int): Array<Pahlawan?>
        {
            return arrayOfNulls(size)
        }
    }
}
```

### d. PahlawanAdapter

```
package com.example.praktikum2

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import java.util.*

class PahlawanAdapter(private val
pahlawanList:ArrayList<Pahlawan> )
:
RecyclerView.Adapter<PahlawanAdapter.PahlawanViewHolder>()
{
```

```

        var onItemClick : ((Pahlawan) -> Unit)? = null

        class PahlawanViewHolder(itemView: View) :
            RecyclerView.ViewHolder(itemView) {
                val imageView: ImageView =
                    itemView.findViewById(R.id.imageView)
                val textView:
                    TextView = itemView.findViewById(R.id.textView)
            }

        override fun onCreateViewHolder(parent: ViewGroup,
            viewType: Int): PahlawanAdapter.PahlawanViewHolder {
            val view =
                LayoutInflater.from(parent.context).inflate(R.layout.each_
                    item, parent, false)
            return PahlawanViewHolder(view)
        }

        override fun onBindViewHolder(holder:
            PahlawanViewHolder, position: Int) {
            val Pahlawan = pahlawanList[position]
            holder.imageView.setImageResource(Pahlawan.image)
            holder.textView.text = Pahlawan.name

            holder.itemView.setOnClickListener {
                onItemClick?.invoke(Pahlawan)
            }
        }

        override fun getItemCount(): Int {
            return pahlawanList.size
        }
    }
}

```

#### e. activity\_detail

```

<?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=".DetailActivity">

    <ImageView
        android:id="@+id/detailActivityIv"
        android:layout_width="0dp"
        android:layout_height="250dp"
        android:scaleType="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/aminoto"
    />

```

```

        <TextView
            android:id="@+id/detailActivityTv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="H.Oemar Said Tjokroaminoto"
            android:textColor="@color/black"
            android:layout_marginStart="16dp"
            android:textSize="24dp"
            android:textStyle="bold"
            app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/detailActivityIv"
/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

#### f. activity\_main

```

<?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/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"

app:layoutManager="androidx.recyclerview.widget.LinearLayo
utManager"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:itemCount="15"
        tools:listitem="@layout/each_item">
    </androidx.recyclerview.widget.RecyclerView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

#### g. each\_item

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    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="wrap_content"
    app:cardCornerRadius="0dp"
    android:layout_marginHorizontal="10dp"
    android:layout_marginVertical="16dp"

```

```

        app:cardElevation="4dp"
        tools:context=".MainActivity">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <ImageView
                android:layout_width="120dp"
                android:layout_height="100dp"
                android:id="@+id/imageView"
                android:scaleType="fitXY"
                android:src="@drawable/aminoto"

            app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" >
            </ImageView>

            <TextView
                android:id="@+id/textView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/black"
                android:text="TJOKROAMINOTO"
                android:textSize="24sp"
                android:textStyle="bold"

            app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.5"

            app:layout_constraintStart_toEndOf="@id/imageView"
                app:layout_constraintTop_toTopOf="parent">
            </TextView>

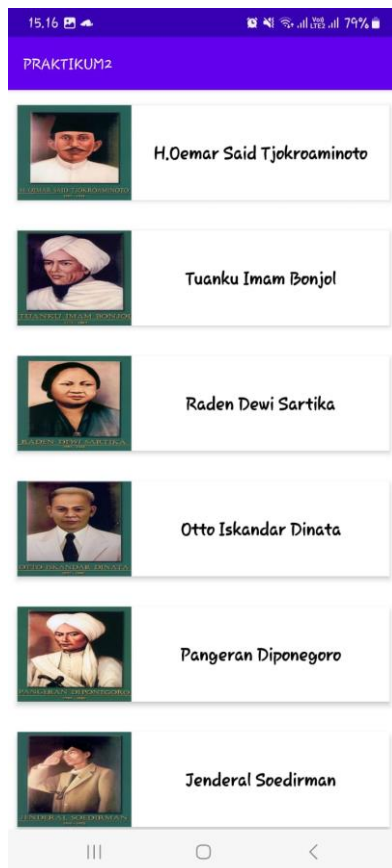
        </androidx.constraintlayout.widget.ConstraintLayout>

    </androidx.cardview.widget.CardView>

```



## 4.1.2 Hasil run



## **BAB V**

### **PENUTUP**

#### **5.1 Analisa**

Dari hasil laporan ini, dapat dianalisa bahwa dalam pembuatan antarmuka aplikasi ini membutuhkan beberapa komponen dasar yang ada. pada dasarnya semua elemen antar pengguna di Aplikasi Android dibangun menggunakan dua buah komponen inti, yaitu View dan ViewGroup, namun dari setiap View dan ViewGroup sendiri memiliki beberapa turunan atau komponen lagi untuk menyusunnya sehingga terdapat komponen – komponen kecil untuk dapat membangun sebuah antarmuka aplikasi

Pada modul ini kita belajar membuat antar muka sebuah aplikasi dasar, dengan Bahasa pemrograman Java dan Kotlin. Namun berbeda dengan materi Modul 1 kemarin untuk Modul 2 ini lebih banyak mempelajari komponen – komponen dasar yang ada pada Android Studio.

#### **5.2 Kesimpulan**

Dalam pembuatan aplikasi pada pembelajaran kali ini tidak jauh beda dengan modul kemarin, namun materi yang dipelajari pada modul ini ada beberapa komponen dasar untuk membangun antarmuka sebuah aplikasi di Android studio, antara lain :

- Layout merupakan halaman depan atau bagian paling depan yang terdiri dari komponen View dan ViewGroup dan dari masing – masing komponen tersebut terdiri dari komponen – komponen kecil lainnya.
- Widget View merupakan visualisasi dari elemen user interface (UI) yang digunakan pada layar aplikasi Android di mana kita dapat merancang sendiri sesuai kebutuhan. Widget View sendiri terdiri dari 2 komponen yaitu Event Handling dan Event Listener
- RecyclerView adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan UI controller, ViewModel, dan LiveData. RecyclerView sendiri hampir sama dengan scroll view namun komponen dari recyclerview lebih baik karena penyimpanannya bisa memuat ribuan data tanpa mengalami hang.