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

NAMA	**: MUHAMMAD ARIQ ARDIANSYAH**
N.R.P	**: 210441100161**
DOSEN	**: Ir.Ach. Dafid, S.T., M.T.**
ASISTENSI	**: David Nasrulloh**
TGL PRAKTIKUM	**: 31 Maret 2023**

Disetujui : Maret 2023
Asisten

DAVID NASRULLOH
190441100060

LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Android Studio yang merupakan peranti pengembang aplikasi Android yang disediakan oleh Android Developers. Peranti ini menyediakan Android SDK (Software Development Kit) yang memudahkan pembuatan atau pengembangan aplikasi Android.dan kali ini akan membahas tentang Layout, Widget View Dan Recycler View.

Layout adalah struktur tampilan antarmuka pengguna atau User Interface (UI) pada aplikasi Android. Dalam Layout, kita dapat menentukan posisi, ukuran, dan tampilan elemen UI seperti tombol, teks, gambar, dan lain-lain. Ada beberapa jenis Layout seperti Linear Layout, Relative Layout, Constraint Layout, dan lain-lain, yang dapat kita gunakan untuk membuat antarmuka pengguna yang berbeda-beda.

Widget View adalah komponen yang digunakan untuk menampilkan informasi atau mengumpulkan input dari pengguna dalam antarmuka pengguna pada aplikasi Android. Widget View bisa berupa tombol, teks, gambar, dan lain-lain, dan dapat ditempatkan dalam Layout.

Recycler View adalah salah satu jenis Widget View yang digunakan untuk menampilkan daftar item atau data dalam antarmuka pengguna pada aplikasi Android. Recycler View memungkinkan kita untuk mengambil data dalam jumlah besar dan menampilkannya secara efisien dalam sebuah daftar.

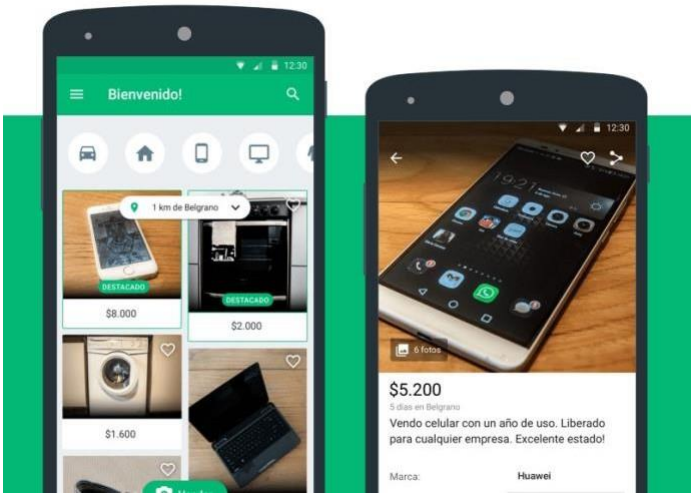
1.2 Tujuan

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

BAB II
DASAR TEORI

2.1 Layout

Pada modul ini, kita akan mempelajari komponen View dan ViewGroup. Kedua komponen ini dapat berkolaborasi sehingga membentuk antar muka dengan contoh sepertipada gambar di bawah ini:



Pada dasarnya semua elemen antar pengguna di aplikasi Android dibangun menggunakandua 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 :

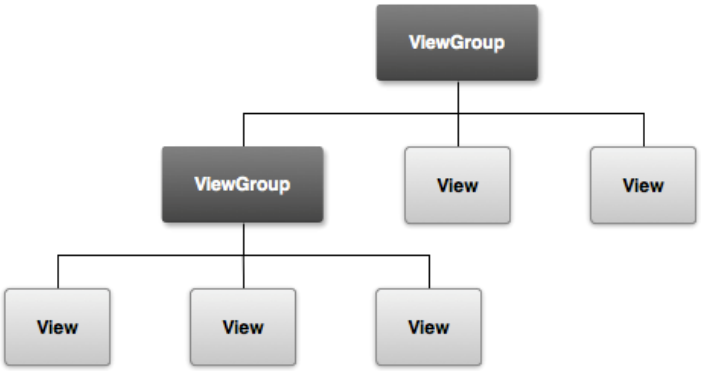
- **TextView**, komponen yang berguna untuk menampilkan teks ke layar.
- **Button**, komponen yang membuat pengguna dapat berinteraksi dengan cara ditekanuntuk melakukan sesuatu.
- **ImageView**, Komponen untuk menampilkan gambar.
- **ListView**, komponen untuk menampilkan informasi dalam bentuk list.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

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

Contoh komponen viewgroup adalah:

- **LinearLayout**
- **FrameLayout**
- **RelativeLayout**
- **TableLayout**

Hierarki komponen view dan viewgroup dapat digambarkan dengan diagram berikut:



Jika diterjemahkan di dalam sebuah viewgroup akan ditampung dua buah komponen view dan satu komponen viewgroup yang terdiri dari 3 buah komponen view. Salah satu contoh dari tampilan dalam file layout xml untuk merepresentasikan kolaborasi view dan viewgroupseperti ini：

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
3.      android:layout_width="match_parent"
4.      android:layout_height="match_parent"
5.      android:orientation="vertical" >
6.    <TextView android:id="@+id/text"
7.      android:layout_width="wrap_content"
8.    android:layout_height="wrap_content"
9.    android:text="I am a TextView" />
```

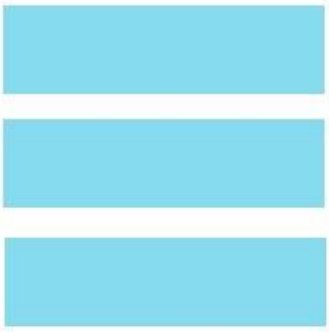
```
10.    <Button android:id="@+id/button"
11.            android:layout_width="wrap_content"
12.            android:layout_height="wrap_content"
13.            android:text="I am a Button" />
14. </LinearLayout>
```

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.

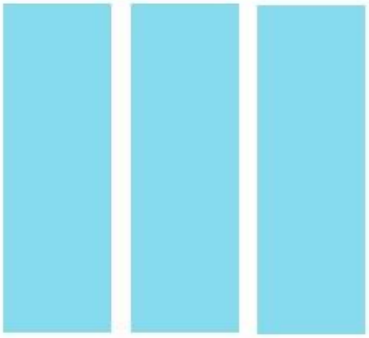
Berikut adalah definisi singkat dan inti dari komponen-komponen di atas terhadap penempatan komponen view (*child*) di dalamnya. Kita akan membahas Linear Layout dan Constrain Layout.

LinearLayout

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



android:orientation=”vertical”



android:orientation=”horizontal”

ConstrainLayout. Apaitu ConstraintLayout?

(https://blog.dicoding.com/kenal- lebihdekat-dengan-constraintlayout/)

ConstraintLayout 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.

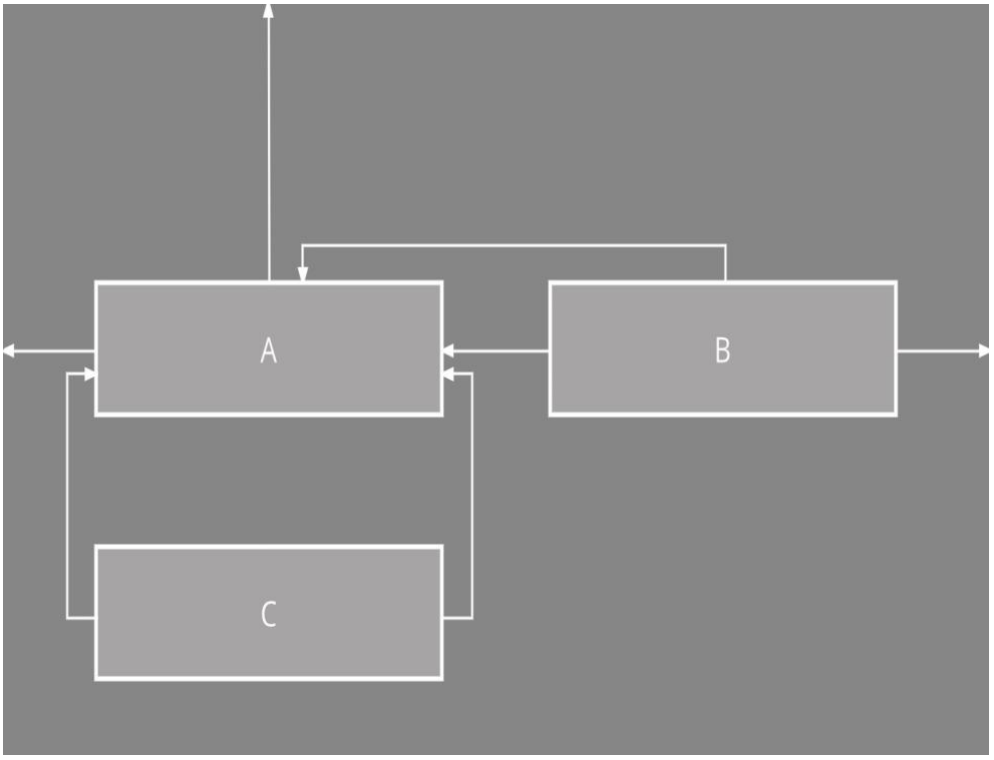
ConstraintLayout memiliki kesamaan dengan RelativeLayout. Dalam penggunaan semua view yang berada di dalamnya disusun berhubungan antara parent dan view lainnya. Tapi ConstraintLayout lebih fleksibel dari RelativeLayout dan mudah digunakan dengan dukungan Layout Editor pada Android Studio.

Let’s say kita menambah view baru ke dalam ConstraintLayout. Kita gunakan drag and drop di Layout Editor yang berada pada tab Design atau dengan menambahnya secara manual melalui tab Text. Kita perlu menentukan posisi dari view atau bagaimana agar view tersebut terhubung dengan parent layout atau view lainnya.

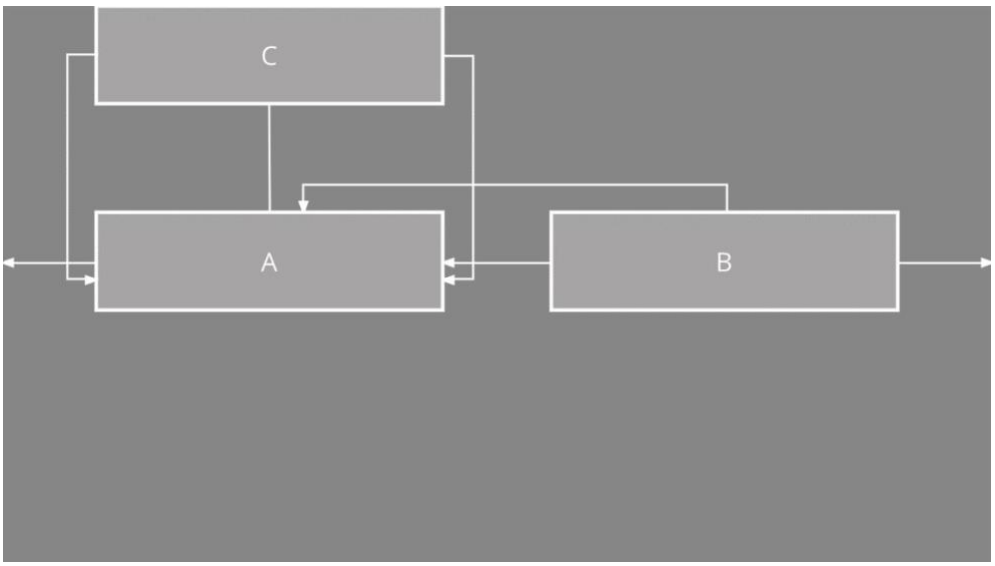
Kenapa gerangan? Karena setelah ditambahkan, view tersebut tidak memiliki constraint yang menghubungkannya dengan parent layout atau view lainnya. Sehingga ketika dijalankan, posisi dari view tersebut akan berada di bagian atas sebelah kiri.

Berbeda ceritanya dengan RelativeLayout. Saat kita ingin menentukan posisi atau menghubungkan dua buah view, kita bisa menggunakan attribute seperti layout_below atau layout_above. Nah untuk ConstraintLayout kita akan menggunakan constraint sebagai dasar dalam menentukan posisi agar sebuah view dapat terhubung dengan view lainnya sesuai harapan kita.

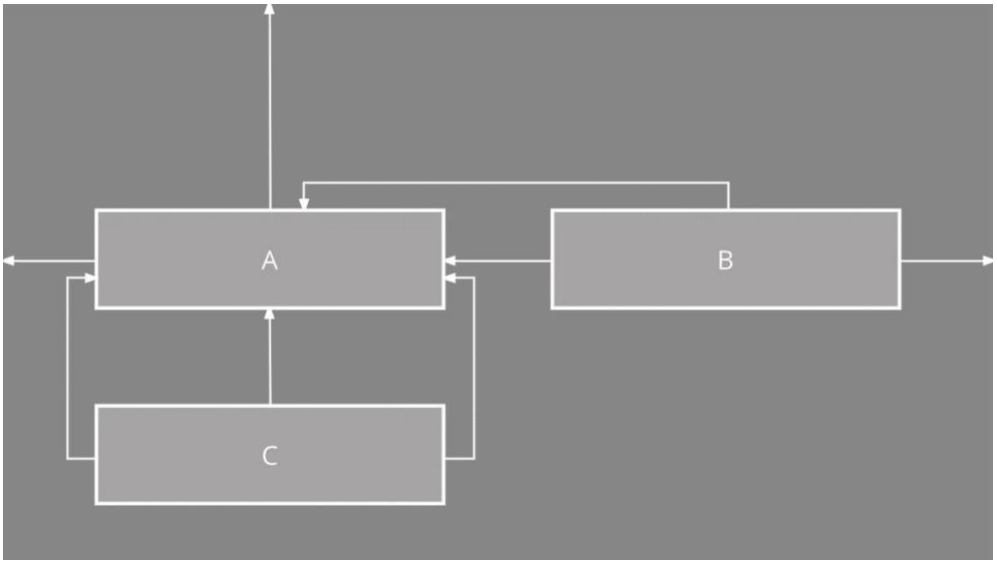
Setiap view setidaknya memiliki satu vertikal dan horizontal constraint. Misal kita memiliki sebuah layout dengan tampilan pada Layout Editor seperti berikut:



Susunan tampilan di atas akan terlihat normal. Tidak ada yang salah di Layout Editor. Tapijika kita perhatikan seksama, **view C** diatas hanya memiliki *horizontal constraint* yang diatursejajar dengan **view A**. Sehingga ketika jika kita coba menjalankannya, sama seperti yangdisebutkan diatas, maka posisi dari **view C** akan berada di posisi atas seperti berikut:



Berbeda jika kita *menambahkan* vertikal constraint pada **view C** yang diatur terikat dengan**view A** seperti berikut:



Ketika dijalankan, apa yang terjadi? Yang tampil akan sesuai dengan apa yang terlihat di Layout Editor.

2.2 Komponen Widget View

Paket widget pada dasarnya merupakan visualisasi dari elemen user interface (UI) yangdigunakan pada layar aplikasi Android di mana kita dapat merancang sendiri sesuaikebutuhan.

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-fileyang 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 tekske 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 RadioGroupuser 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 callbacktunggal. Method ini akan dipanggil oleh framework Android jika View yang telah didaftarkandengan listener dipicu oleh interaksi pengguna dengan item dalam UI.Yang juga disertakan dalam antarmuka event listener adalah method callback berikut ini:

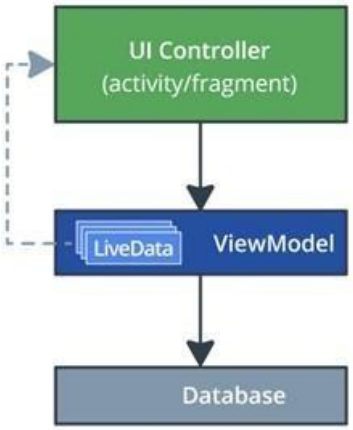
1. Method onClick() dari View.OnClickListener. Ini dipanggil baik saat pengguna menyentuh item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan

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. MethodonCreateContextMenu() dariView.OnCreateContextMenuListene. Ini dipanggilsaat 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.

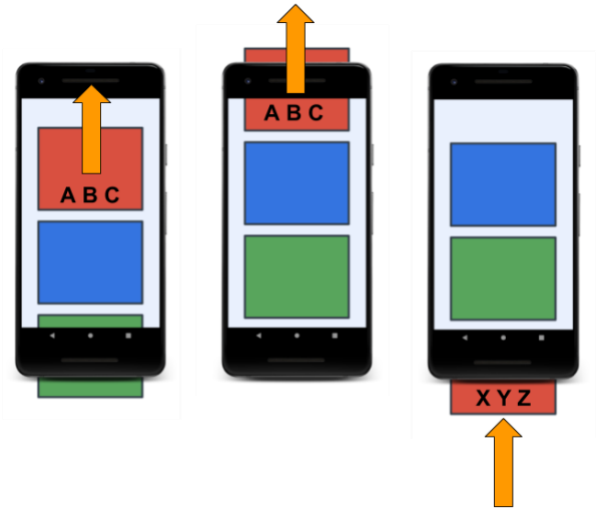


Menampilkan list atau grid data adalah salah satu tugas UI paling umum di Android. Daftarbervariasi dari yang sederhana hingga yang sangat kompleks. Daftar tampilan teks mungkin menampilkan data sederhana, seperti daftar belanja. Daftar yang kompleks, seperti daftar tujuan liburan yang beranotasi, dapat 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 menggambaritem yang saat ini terlihat di layar. Misalnya, jika list memiliki seribu elemen tetapi hanya 10elemen 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 banyakwaktu pemrosesan dan membantu scroll list dengan lancar.
- Ketika suatu item berubah, alih-alih menggambar ulang seluruh daftar,

RecyclerViewdapat memperbarui satu item itu. Ini adalah keuntungan efisiensi yang sangat besar ketikamenampilkan daftar item kompleks!

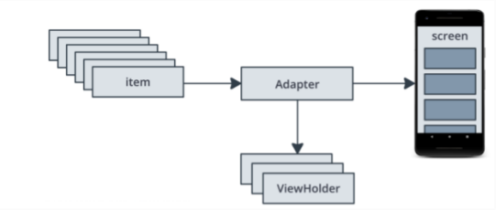
Dalam urutan yang ditunjukkan di bawah ini, kita dapat melihat bahwa satu tampilan telah diisi dengan data, ABC. Setelah itu tampilan bergulir dari layar, RecyclerView menggunakankembali tampilan untuk data baru, XYZ.



Adapter pattern

Jika kita pernah bepergian antar negara yang menggunakan soket listrik yang berbeda, kitamungkin tahu bagaimana kita bisa mencolokkan perangkat kita ke outlet dengan menggunakan adaptor. Adaptor memungkinkan kita mengonversi satu jenis steker ke yang lain, yang benar-benar mengubah satu antarmuka menjadi yang lain. 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.

Mengimplementasikan sebuah RecyclerView



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

- Data untuk ditampilkan.
- Mesin virtual RecyclerView didefinisikan dalam file layout, untuk bertindak sebagai wadahuntuk tampilan.
- 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.
- Layout Manager.

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

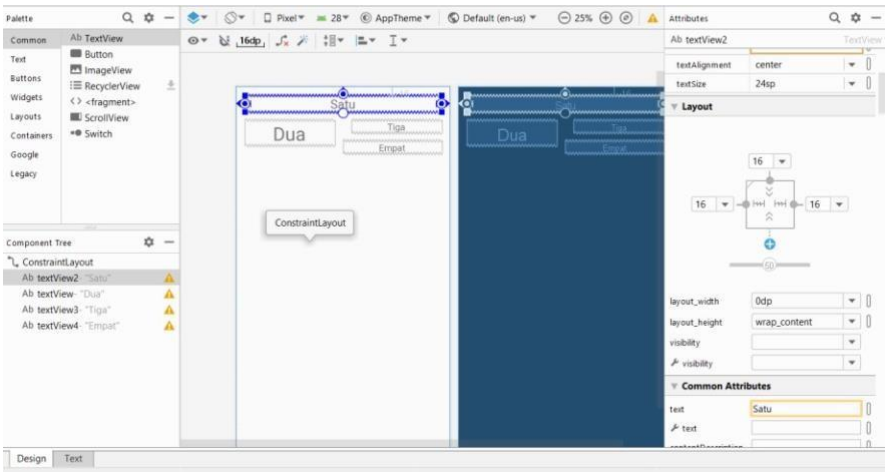
  Adaptor menghubungkan data kita ke RecyclerView. Ini menyesuaikan data sehingga dapat ditampilkan di ViewHolder. RecyclerView menggunakan adaptor untuk mengetahuicara menampilkan data di layar.

BAB III

TUGAS PENDAHULUAN

3.1 Layout

1. Buat project baru dengan desain sebagai berikut.



2. Buat project baru dengan menggunakan Linear Layout dengan minimal 3 komponen(TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.
3. Buat project baru dengan menggunakan Constrain Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.

3.2 Komponen Widget View

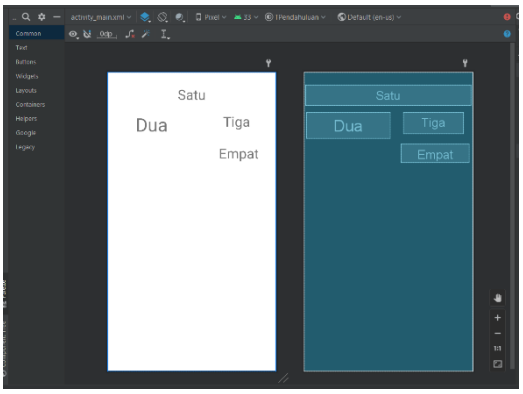
1. Buat project baru, buat antar muka berbeda yang melibatkan komponen-komponen diatas.

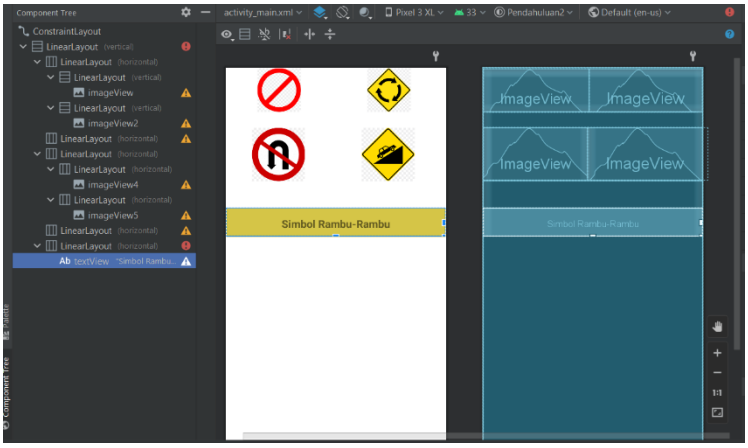
3.3 Recycler View

1. Modifikasilah aplikasi dengan menambahkan Toast jika salah satu list dipilih.

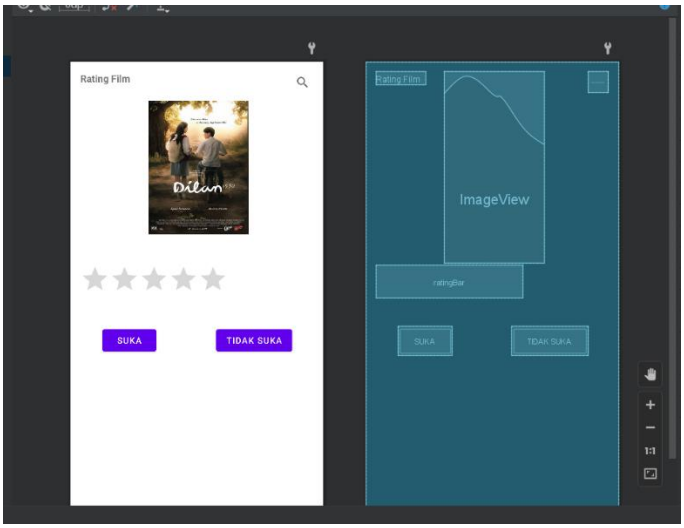
3.4 Jawaban

- **Layouting**

1. 

2. 

- **Komponen Widget View**

  1. 

BAB III

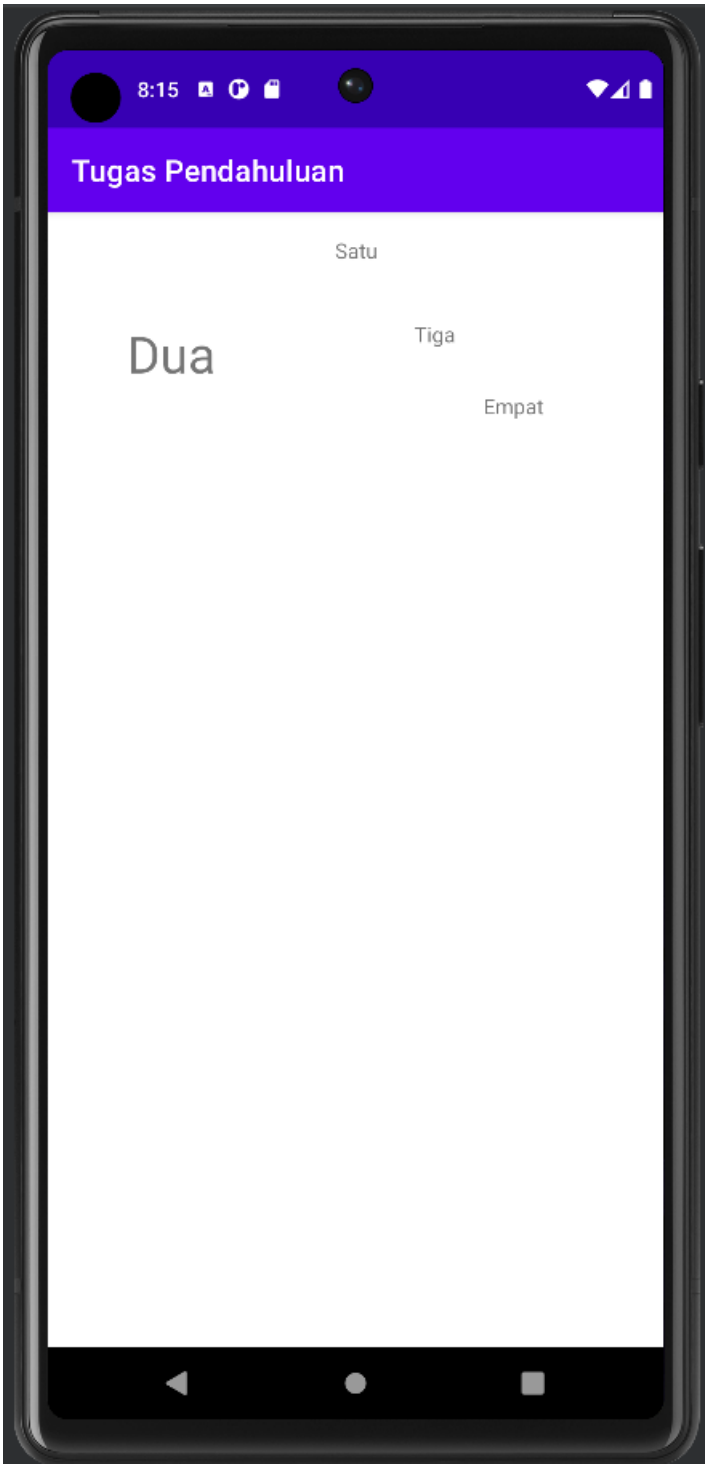
TUGAS PENDAHULUAN

1. 1 Source Code

```
2. <?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">

       <TextView
           android:id="@+id/textView2"
           android:layout_width="0dp"
           android:layout_height="24dp"
           android:layout_marginStart="21dp"
           android:layout_marginTop="16dp"
           android:layout_marginEnd="20dp"
           android:text="Satu"
           android:textAlignment="center"
           app:layout_constraintEnd_toEndOf="parent"
           app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toTopOf="parent" />

       <TextView
           android:id="@+id/textView"
           android:layout_width="124dp"
           android:layout_height="49dp"
           android:layout_marginStart="20dp"
           android:layout_marginTop="72dp"
           android:layout_marginEnd="267dp"
           android:text="Dua"
           android:textAlignment="center"
           android:textSize="34sp"
           app:layout_constraintEnd_toEndOf="parent"
           app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toTopOf="parent" />

       <TextView
           android:id="@+id/textView3"
           android:layout_width="131dp"
           android:layout_height="22dp"
           android:layout_marginStart="281dp"
           android:layout_marginTop="72dp"
           android:layout_marginEnd="72dp"
           android:text="Tiga"
           app:layout_constraintEnd_toEndOf="parent"
           app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toTopOf="parent"
           tools:textAlignment="center" />

       <TextView
           android:id="@+id/textView4"
```

```
            android:layout_width="149dp"
            android:layout_height="20dp"
            android:layout_marginStart="282dp"
            android:layout_marginTop="120dp"
            android:layout_marginEnd="71dp"
            android:text="Empat"
            android:textAlignment="center"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

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

1.2 Hasil output



2.1. Buat project baru dengan menggunakan Linear Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.

2.2. Source Code

```
3. <?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">

       <LinearLayout
           android:layout_width="409dp"
           android:layout_height="729dp"
           android:layout_marginStart="1dp"
           android:layout_marginTop="1dp"
           android:layout_marginEnd="1dp"
           app:layout_constraintEnd_toEndOf="parent"
           app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toTopOf="parent">

           <TextView
               android:id="@+id/textView"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:layout_weight="1"
               android:text="HELLO"
               tools:textAlignment="center" />

           <TextView
               android:id="@+id/textView2"
               android:layout_width="wrap_content"
               android:layout_height="35dp"
               android:layout_weight="1"
               android:text="WORLD!!!"
               android:textSize="24sp"
               android:translationY="3dp"
               tools:textAlignment="center" />

           <Button
               android:id="@+id/button"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:layout_weight="1"
               android:text="Button" />
       </LinearLayout>
   </androidx.constraintlayout.widget.ConstraintLayout>
```

2.3 Hasil Output

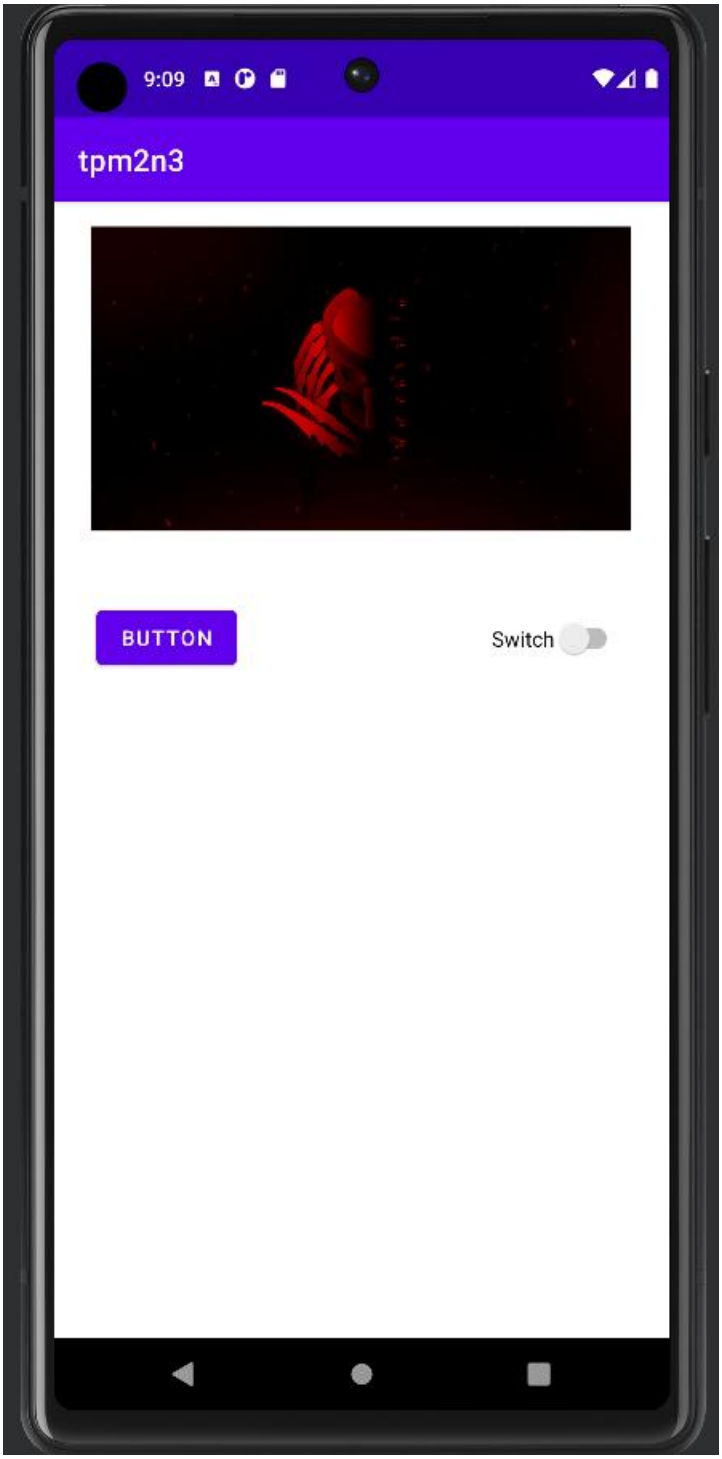
3.1 Buat project baru dengan menggunakan Constrain Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.

3.2 Source Code

```
<?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">


    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="362dp"
        android:layout_height="203dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.489"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.03"
        app:srcCompat="@drawable/_209321" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="28dp"
        android:layout_marginTop="34dp"
        android:layout_marginEnd="289dp"
        android:layout_marginBottom="430dp"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/imageView2" />

    <Switch
        android:id="@+id/switch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="170dp"
        android:layout_marginTop="45dp"
        android:layout_marginEnd="30dp"
        android:layout_marginBottom="440dp"
        android:text="Switch"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/button"
```

```
app:layout_constraintTop_toBottomOf="@+id/imageView2" />

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

3.3 Hasil Output



BAB IV

IMPLEMENTASI

4.1 Soal

1. Buat Project pada perangkat komputer anda untuk mengimplementasikan layout dengan berbagai bentuk tampilan.
2. Komponen Widget View,Analisislah atribut komponen untuk constrain layout.
3. Recycler View,Buat aplikasi baru dengan menerapkan RecycleView.

4.2 Hasil

- **Source Code**

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:layout_width="match_parent"
    android:orientation="vertical"
    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"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

RecycleViewTemp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="8dp"
    android:orientation="horizontal"
    android:background="#808080">
```

```
    <ImageView
        android:id="@+id/img_view"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@mipmap/ic_launcher"/>

    <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="#000000"
            android:textSize="18sp"/>

        <TextView
            android:id="@+id/txt_sub_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Sub Title"/>


    </LinearLayout>
</LinearLayout>
```

gamePs2

```
package com.test.ariqrecycleview

data class gamePs2(val imgView: Int, val txtTitle: String, val
txtSubTitle: String)
```

RvAdapter

```
package com.test.ariqrecycleview

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

class RvAdapter(private val data: ArrayList<gamePs2>):
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) {
        holder.bind(data[position])
    }
}
```

ViewHolder

```
package com.test.ariqrecycleview

import android.view.LayoutInflater
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.recycler_view_te
mp, parent, false)) {

        private var imgView: ImageView? = null
        private var txtTitle: TextView? = null
        private var txtSubTitle: TextView? = null

        init {
            imgView = itemView.findViewById(R.id.img_view)
            txtTitle = itemView.findViewById(R.id.txt_title)
            txtSubTitle =
itemView.findViewById(R.id.txt_sub_title)
        }

        fun bind(data: gamePs2){
            imgView?.setImageResource(data.imgView)
            txtTitle?.text = data.txtTitle
            txtSubTitle?.text = data.txtSubTitle
        }


}
```

MainActivity.kt

```
package com.test.ariqrecycleview

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

class MainActivity : AppCompatActivity() {

    lateinit var recycleView: RecyclerView
    lateinit var adapter: RvAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        init()

            recycleView.layoutManager = LinearLayoutManager(this)
                    recycleView . adapter = adapter

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

        var data = ArrayList<gamePs2>()
        data.add(gamePs2(R.drawable.ace5, txtTitle = "Ace Combat
5", txtSubTitle = "permainan video penerbangan tempur semi-
realistis untuk Playstation 2. Seperti gim lain di serial Ace
Combat milik Namco, Ace Combat 5 menampilkan permainan yang
merupakan campuran antara penerbangan arcade dan simulasi
penerbangan nyata. Gim ini dikembangkan oleh Project Aces, sebuah
kelompok internal Namco yang bertanggungjawab atas pengembangan
serial Ace Combat,[3] dan dipublikasikan oleh Namco pada bulan
Oktober 2004. Sejumlah gim terbatas dibundel dengan aksesoris Hori
Flightstick 2."))
        data.add(gamePs2(R.drawable.god_hand, txtTitle = "God
Hand", txtSubTitle = " permainan video action beat 'em up yang
dikembangkan oleh Clover Studio dan diterbitkan oleh Capcom untuk
konsol permainan PlayStation 2. Permainan ini disutradarai oleh
desainer Resident Evil Shinji Mikami, dan dirilis di Jepang dan
Amerika Utara pada 2006 dan pada 2007 untuk wilayah PAL; pada 4
Oktober 2011, permainan ini dirilis ulang untuk PlayStation 3
sebagai title dapat diunduh di PlayStation Network. Mikami ini
adalah untuk menciptakan sebuah permainan Aksi yang ditujukan
untuk \" hardcore gamers \" intermixed dengan sejumlah besar comic
relief. Permainan ini menerima respon positif dari kritikus dan
dirilis di Jepang terjual wajar. Ini adalah permainan video
terakhir Clover Studio."))
        data.add(gamePs2(R.drawable.mksm, txtTitle = "Mortal
Kombat Shaolin Monks", txtSubTitle = " aksi petualangan Permainan
bertarung Berjuang permainan video berdasarkan Mortal Kombat
serangkaian game pertarungan. Shaolin Monks dikembangkan oleh
Midway Studios LA (fsebelumnya Pengembangan Paradoks), Midway -
San Diego, dan Mortal Kombat Team - Chicago, dan diterbitkan oleh
Midway Games for the PlayStation 2 and Xbox.[1] Itu dirilis 16
September 2005 di Amerika Serikat dan 30 September 2005 di Eropa
untuk kedua platform."))
        data.add(gamePs2(R.drawable.re4, txtTitle = "Resident Evil
4", txtSubTitle = "Leon adalah seorang agen spesialis dibawah
komando langsung oleh presiden Amerika Serikat, yang ditugaskan
untuk menyelamatkan putri sang presiden. Zombie spesies baru pun
muncul, diberi nama Ganados, yang merupakan manusia yang dikontrol
oleh satu orang dengan menggunakan sebuah parasit yang bernam
Plaga. Sepanjang perjalanan, Leon bertemunya dengan kawan-kawan
lamanya yaitu Ada dan Krauser."))
        data.add(gamePs2(R.drawable.mw, txtTitle = "Most Wanted",
txtSubTitle = "Pemain tiba di Rockport City, keluar dari dunia
balap ilegal bawah tanah, dengan mobil BMW M3 GTR GT (E46) yang
dimodifikasi. Mengikuti Mia Townsend (Josie Maran), pemain
membuktikan kemampuan mengemudinya saat dia dikejar oleh seorang
petugas polisi veteran bernama Cross (Dean McKenzie), dengan
```

```
asistennya yang namanya tidak disebut (Simone Bailey), yang
bersumpah untuk menangkap pemain tersebut."))


        adapter = RvAdapter(data)
    }

}
```

- **Hasil Run**

Penjelasan Program

Pertama Setelah membuat projek baru kita ke bagian gradle scripts an buka gradle build app kemudian memasukkan implementation untuk recyclerview

Setelah itu beralih ke file activity_main.xml dan mengganti constrainlayout menjadi linear layout dengan orientation vertical diikuti dengan memasukkan recyclerview tadi dan dikasih id. Kemudian kita membuat template agar bisa memasukkan data kita le recyclerview dengan menambah file xml baru di folder layout dengan nama file recycler_view_temp kemudian mengisinya dengan komponen yang diperlukan

Kemudian membuat data class dengan nama file gamePs2, ke MainActuvity.kt membuat fungsi init, diatas oncreate membuat 1 varibael yaitu lateinit var recyclerview dengan tipe data Recyclerview, di fungsi init kita menaggil recyclerview = findviewbyid, dengan diikuti sebuah data yang akan ditampilkan di recyclerview

Membuat class adapter dengan nama RvAdapter kemudian passing data yang di recyclerview tadi dan extend nya dengan Recyclerview.adapter<

Lanjut membuat class ViewHolder dengan diberi 3 parameter yaitu layout, recycler_view_temp, parent, false. Kemudian membuat suatu variabekl yang akan dipakai seperti imageview, textview, subtexttitle terus menginisialisasiketuga variable tersebut kemudian bind data tersebut dengan membuat function bind yang menerima parameter data dengan tipe data gameps2, dengan begitu View Holder sudah selesai

Selanjutnya yaitu meng-implement member Rvadapter karena classnya msih abstrak, yang terakhir membuat lateinit var adapter: RvAdapter di mainActivity.kt kemudian membuat init recyclerview, sebelum membuat init tersebut kita membuat set adapter = RvAdapter(data), kemudian Run.

BAB V
PENUTUP

5.1 Analisa

Dalam keseluruhan, Bab 2 dari Android Studio sangat penting bagi pengembang Android karena Android SDK (Software Development Kit) yang memudahkan pembuatan atau pengembangan aplikasi Android yang sangat diperlukan dalam pembuatan aplikasi Android yaitu tentang Layout, Widget View Dan Recycler View. Dengan memahami dan menguasai konsep-konsep ini, pengembang dapat membuat aplikasiyang lebih efektif dan responsif bagi pengguna.

5.2 Kesimpulan

Layout adalah cara untuk menentukan tata letak atau posisi dari elemen-elemen UI pada sebuah tampilan. Android Studio menyediakan beberapa jenis layout yang berbeda seperti LinearLayout, RelativeLayout, dan ConstraintLayout untuk membantu pengembang dalam membuat tampilan yang responsif dan mudah diatur.

Widget View adalah elemen UI seperti tombol, label, teks box, dan lainnya yang dapat ditemukan dalam tampilan aplikasi. Android Studio menyediakan banyak widget view yang siap digunakan. Beberapa widget view yang populer adalah TextView untuk menampilkan teks, Button untuk tombol, ImageView untuk menampilkan gambar, dan CheckBox untuk opsi pilihan.

RecyclerView adalah tampilan daftar yang dapat menampilkan elemen-elemen UI dalam jumlah besar dan dinamis. Android Studio menyediakan RecyclerView sebagai alternatif yang lebih efisien dan fleksibel dari ListView, yang digunakan untuk menampilkan daftar statis. RecyclerView memiliki kemampuan untuk menampilkan tampilan item yang berbeda, seperti grid, list, atau

tampilan kustom, dan dapat menangani perubahan data yang dinamis dengan lebih efisien.