

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



<b>NAMA</b>	<b>: NIKEN AYU SAFITRI</b>
<b>N.R.P</b>	<b>: 200441100033</b>
<b>DOSEn</b>	<b>: ACHMAD DAFID ,S.T., M.T</b>
<b>ASISTEN</b>	<b>: ALI GUFRON</b>
<b>TGL PRAKTIKUM</b>	<b>: 31 MARET 2023</b>

**Disetujui : 06 April 2023**  
**Asisten**

**ALI GUFRON**  
**19.04.411.00.140**



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

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Bahasa pemrograman komputer memiliki peranan penting dalam mengembangkan sebuah aplikasi, bahasa pemrograman komputer memungkinkan programmer untuk memberikan instruksi ke dalam bahasa yang dimengerti komputer, dan memungkinkan komputer dengan cepat dan efisien memproses informasi yang besar dan kompleks. Dalam mengembangkan aplikasi Android, terdapat beberapa bahasa pemrograman yang dapat digunakan yaitu C++, Java, dan pada tahun 2017 Google menambahkan bahasa resmi untuk mengembangkan aplikasi Android yaitu Kotlin.

Seiring dengan semakin berkembangnya aplikasi android dan sekarang perkembangan teknologi yang sangat pesat maka semakin banyak juga ragam tampilannya. Pada praktikum sebelumnya sudah membahas tentang pengenalan android. Selanjutnya kita akan mempelajari untuk membuat aplikasi yang lebih menarik dengan menggunakan beberapa object view groups dan view.

Pada android studio yang menjadi bagian dasar terletak pada bagian tampilan. Dalam layout biasanya didalamnya terdiri dari satu atau lebih objek view sehingga dapatlah disebut Layout adalah View Group. Di android studio ada beberapa layout yang digunakan salah satunya Linear Layout yang digunakan untuk menyusun view secara linear vertikal atau linear horizontal.

Android studio juga memiliki view yang lain yaitu Recycle View yang merupakan salah satu komponen pada Android Studio yang memungkinkan aplikasi mampu menampung ratusan ribu data tanpa mengalami eror pada aplikasi. Recycle View cocok digunakan untuk menampung data yang banyak

Pembelajaran praktikum ini mempelajari tentang codelab yang nantinya menuntun kita membangun aplikasi android dengan menggunakan Kotlin. Prasyarat untuk dapat mengikuti praktikum ini dengan baik adalah memiliki pengetahuan dalam Bahasa pemrograman berorientasi objek penuh seperti

java. Untuk pengembangan, disarankan untuk menggunakan sumber referensi selain modul praktikum ini

## **1.2 Tujuan**

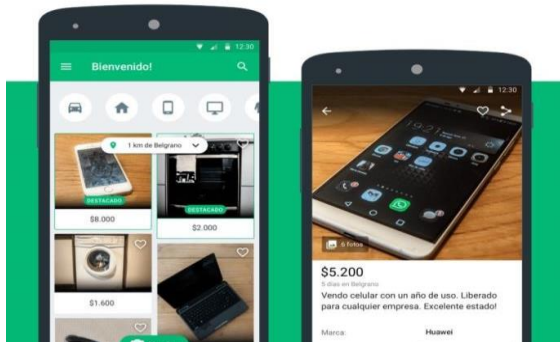
- Membuat Layout dengan Linear Layout dan Constraint Layout
- Mampu menggunakan Widget View (masukan) untuk membuat aplikasi sederhana
- Merepresentasikan data dengan menggunakan komponen recylerview

## 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 seperti pada gambar di bawah ini.



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.

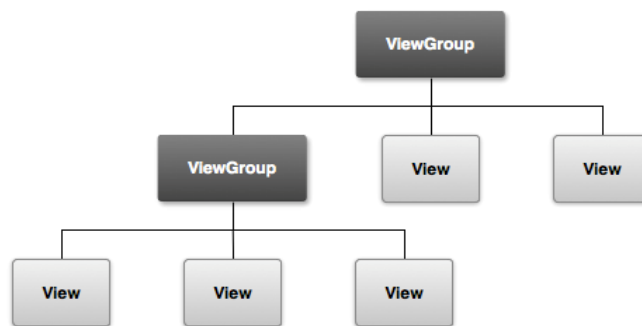
Contoh komponen turunan dari view seperti :

- 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 dari 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:

- LinearLayout
- FrameLayout
- RelativeLayout
- TableLayout

Hierarki komponen view dan viewgroup dapat digambarkan dengan diagram berikut:



Jika diterjemahkan di dalam sebuah viewgroup akan ditampilkan 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 viewgroup seperti ini :

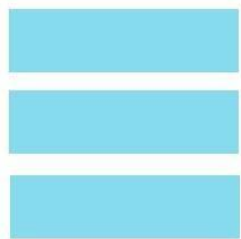
1.	<code>&lt;?xml version="1.0" encoding="utf-8"?&gt;</code>
2.	<code>&lt;LinearLayout</code> <code>xmlns:android="http://schemas.android.com/apk/res/android"</code>
3.	<code>android:layout_width="match_parent"</code>
4.	<code>android:layout_height="match_parent"</code>
5.	<code>android:orientation="vertical" &gt;</code>
6.	<code>&lt;TextView android:id="@+id/text"</code>
7.	<code>android:layout_width="wrap_content"</code>
8.	<code>android:layout_height="wrap_content"</code>
9.	<code>android:text="I am a TextView" /&gt;</code>
10.	<code>&lt;Button android:id="@+id/button"</code>
11.	<code>android:layout_width="wrap_content"</code>
12.	<code>android:layout_height="wrap_content"</code>
13.	<code>android:text="I am a Button" /&gt;</code>
14.	<code>&lt;/LinearLayout&gt;</code>

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.

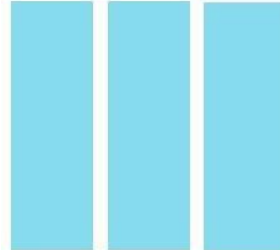
Berikut adalah definisi singkat dan inti dari komponen-komponen di atas terhadap penempatan komponen view (child) di dalamnya. Kita akan membahas Linear Layout dan Constrain Layout.

### 2.1.1 Linear Layout

Linear Layout Merupakan komponen yang tataletaknya ditempatkan secara horizontal atau vertikal. Tata letak linear layout memiliki atribut bobot untuk setiap tampilan anak, yang berfungsi menentukan ukuran desimal tampilan diruang yang tersedia.



`android:orientation="vertical"`

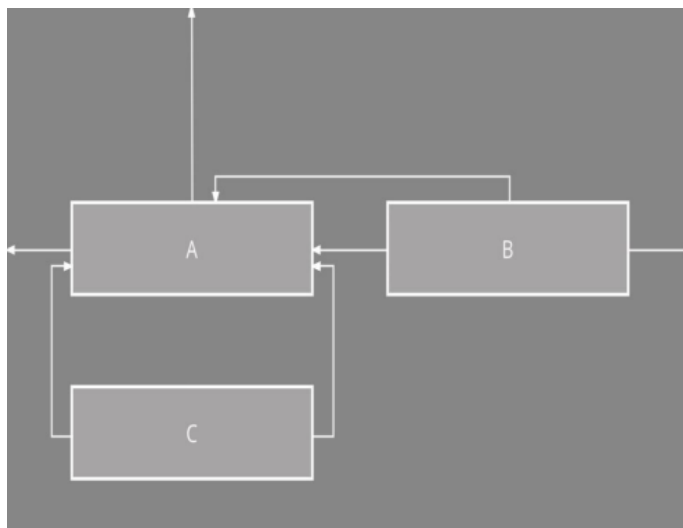


`android:orientation="horizontal"`

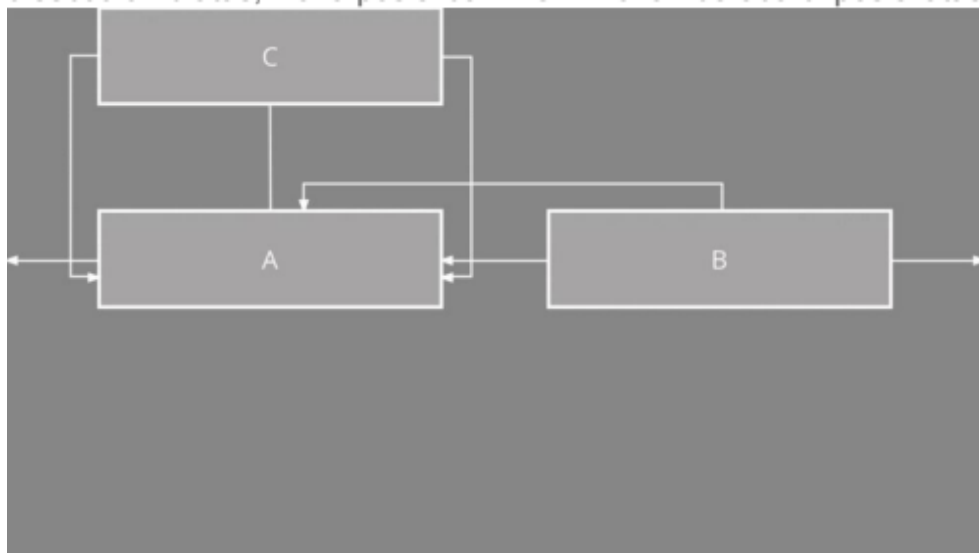
### 2.1.2 Constrain Layout

Constrain Layout merupakan komponen ViewGroup yang bisa digunakan untuk menyusun tampilan aplikasi yang kompleks tanpa layout bertingkat. ConstraintLayout dapat diakses dengan dukungan kesamaan terbaru dari Android 2.3 (Antarmuka pemrograman level 9). Constraint Layout seperti Relative Layout. Ketika digunakan, setiap perspektif yang dikandungnya terletak komparatif dengan pandangan induk dan perspektif yang berbeda. Meskipun demikian, Constraint Layout lebih mudah beradaptasi daripada RelativeLayout dan mudah digunakan dengan bantuan

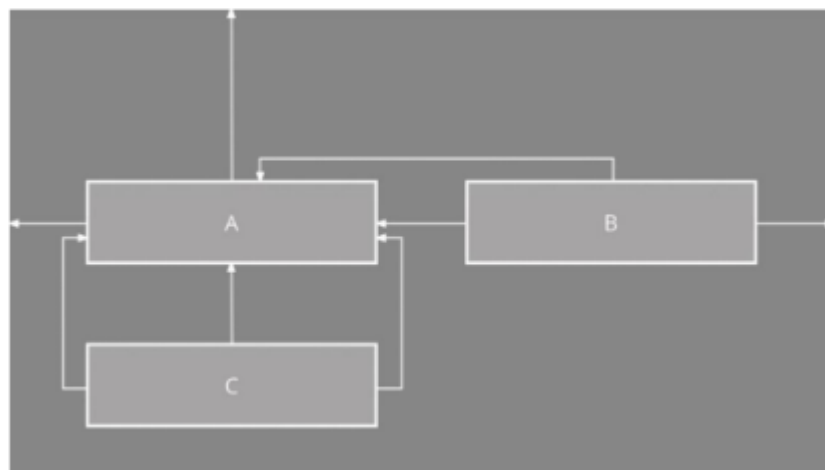
untuk proofreader desain Android Studio. Asumsikan Anda perlu menambahkan tampilan lain ke Constraint Layout. Gunakan yang disederhanakan dalam Format Proofreader pada tab Rencana, atau tambahkan secara fisik dari tab Teks. Anda ingin mengkarakterisasi tempat tampilan, atau hubungannya dengan format induk dan perspektif yang berbeda. Ini karena setelah ditambahkan, tampilan tidak memiliki batasan untuk menghubungkan ke format induk atau perspektif yang berbeda. Makanya, saat eksekusi, posisi etalase berada di kiri atas. Situasinya berbeda dengan Desain Relatif. Untuk menempatkan atau menggabungkan dua perspektif, Anda dapat menggunakan kualitas, misalnya `layout_below` dan `layout_above`. Untuk Constraint Layout, Imperative saat ini digunakan sebagai alasan untuk menunjukkan posisi. Ini memungkinkan Anda untuk menghubungkan tampilan ke tampilan lain



Tata letak tampilan di atas terlihat baik-baik saja. Tidak ada yang salah dengan editor tata letak. Namun, jika Anda perhatikan lebih dekat, hanya ada satu pembatas horizontal di bagian atas Tampilan C yang disetel sejajar dengan Tampilan A. Jadi jika Anda mencoba melakukannya seperti di atas, posisi view C akan berada di posisi teratas sebagai 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) 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.

### **2.2.1 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

### **2.2.2 Event Listener**

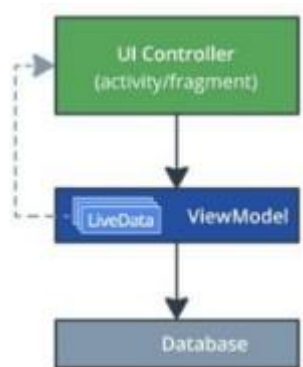
Event listener merupakan antarmuka di kelas View yang berisi method callback tunggal. Method ini akan dipanggil oleh framework Android jika View yang telah didaftarkan dengan listener dipicu oleh interaksi pengguna dengan item dalam UI.

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 `onCreate ContextMenu()` dari `View.OnCreate ContextMenu Listener`. Ini dipanggil saat Menu Konteks sedang dibuat (akibat "klik lama" terus-menerus).

### 2.3 Recycle View

Recycler View adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan UI controller, ViewModel, dan LiveData.



Menampilkan list atau grid data merupakan salah satu tugas UI paling umum di Android. Daftar dapat berkisar dari yang sederhana hingga yang sangat kompleks. Daftar tampilan teks dapat menampilkan data sederhana seperti: B.daftar belanja. Daftar kompleks, seperti daftar tujuan liburan beranotasi, dapat menunjukkan banyak detail kepada pengguna dalam kotak gulir dengan tajuk. Untuk mendukung semua kasus penggunaan ini, Android menyediakan widget Recycler View.

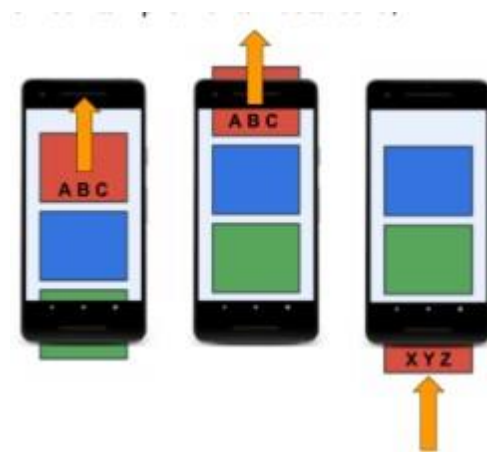


Manfaat terbesar dari RecyclerView adalah sangat efisien untuk daftar besar:

- 1) Secara default, RecyclerView hanya berfungsi untuk memproses atau menggambar item yang sedang ditampilkan di layar. Misalnya, jika Anda memiliki 1.000 item dalam daftar dan hanya 10 yang ditampilkan, RecyclerView hanya akan berfungsi untuk menggambar 10 item di layar Anda. Saat pengguna menggulir, RecyclerView mengenali item baru untuk ditampilkan di layar dan tidak cukup untuk

menampilkan item itu.

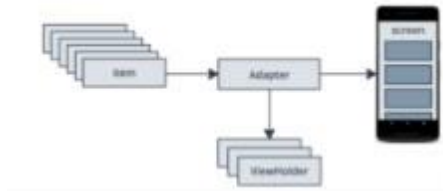
- 2) Saat item digulir keluar dari layar, tampilan item akan digunakan kembali. Ini berarti bahwa elemen tersebut akan diisi dengan konten baru yang menggulir ke bawah layar. Perilaku RecyclerView ini menghemat banyak waktu pemrosesan dan membantu scroll list dengan lancar.
- 3) Ketika suatu item berubah, alihalih 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.



### 2.3.1 Adapter Pattern

Jika bepergian ke negara-negara dengan outlet yang berbeda, mungkin tahu cara menggunakan adaptor untuk menghubungkan perangkat ke outlet listrik. Adaptor memungkinkan untuk mengonversi satu jenis konektor ke yang lain, secara harfiah satu antarmuka ke antarmuka lainnya. Pola adaptor pengembangan perangkat lunak membantu objek berinteroperasi dengan API lain. RecyclerView menggunakan adaptor untuk mengubah data aplikasi Anda menjadi sesuatu yang dapat dilihat RecyclerView, tanpa mengubah cara aplikasi Anda menyimpan dan memproses data. Untuk aplikasi pelacakan tidur, buat adaptor yang menyesuaikan data dengan apa yang dapat

ditampilkan oleh Recycler View tanpa mengubah View Model. Terapkan Recycler View.



Bagian berikut diperlukan untuk menampilkan data di Recycler View.

- Data yang akan ditampilkan.
- Mesin virtual  
Recycler View didefinisikan dalam file layout untuk bertindak sebagai wadah tampilan.
- Tata letak item data tunggal.  
Jika semua item daftar terlihat sama, dapat menggunakan tata letak yang sama untuk semua, tetapi itu tidak wajib. Tata letak elemen harus dibuat secara terpisah dari tata letak fragmen, sehingga dapat membuat tampilan dan mengisi satu elemen dalam satu waktu.
- Manajer tata letak.  
Manajer tata letak menangani organisasi (tata letak) komponen UI dalam tampilan.
- Penampil.  
View holder memperluas kelas View Holder. Berisi informasi tampilan untuk menampilkan satu artikel dari tata letak artikel. Penampil tampilan juga menambahkan informasi yang digunakan Recycler View untuk menggeser tampilan di seluruh layar secara efisien.
- Adaptor.  
Adaptor menghubungkan data ke Recycler View. Ini akan menyesuaikan data sehingga dapat ditampilkan di View Holder. Recycler View menggunakan adaptor untuk memahami bagaimana data akan terlihat di layar.

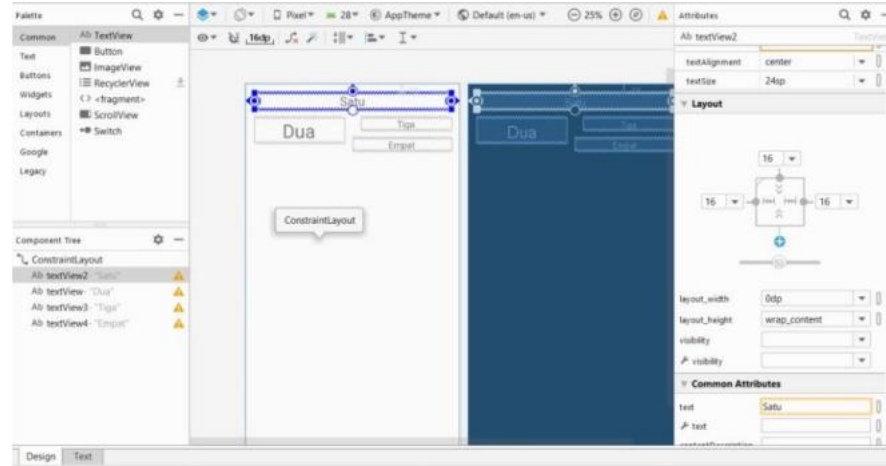
## BAB III

### TUGAS PENDAHULUAN

#### 3.1 Layout

Soal:

1. Buat Project dengan desain sebagai berikut.



Jawab:

a. Analisa

Pada project diatas menggunakan constrain Layout yang didalamnya terdapat komponen yaitu 4 TextView dengan size margin layout yang berbeda-beda untuk mengatur batasnya. Batas atau penempatan layout bisa menggunakan id yang ada pada komponen

b. 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">

    <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>

```

c. Hasil





- TextView untuk penulisan title dengan size layout match\_parent = width dan wrap\_content = height.
- Edit Text untuk pengisian nama, password dengan size sesuai konten.
- Button terdapat 2 button yaitu konfirmasi dan batal dengan warna yang berbeda konfirmasi warna bir dan batal warna merah.

b. SourceCode

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical">

    <TextView
        android:id="@+id/textView6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="15dp"
        android:layout_marginRight="8dp"
        android:text="PERBARUI PROFIL ANDA"
        android:textAlignment="center"
        android:textColor="#070707"
        android:textSize="24sp"
        android:textStyle="bold" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"/>

    <EditText
        android:id="@+id/edittext1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginRight="8dp"
        android:text="Nama"
        android:textColor="#AFAFAF"
        tools:ignore="TouchTargetSizeCheck" />

    <EditText
        android:id="@+id/edittext2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
```

```

        android:layout_marginRight="8dp"
        android:text="NIM"
        android:textColor="#AAA9A9"
        tools:ignore="TouchTargetSizeCheck" />

<EditText
    android:id="@+id/edittext4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:text="Prodi"
    android:textColor="#AEACAC"
    tools:ignore="TouchTargetSizeCheck" />

<EditText
    android:id="@+id/edittext"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:text="Alamat Lengkap"
    android:textColor="#AEACAC"
    tools:ignore="TouchTargetSizeCheck" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

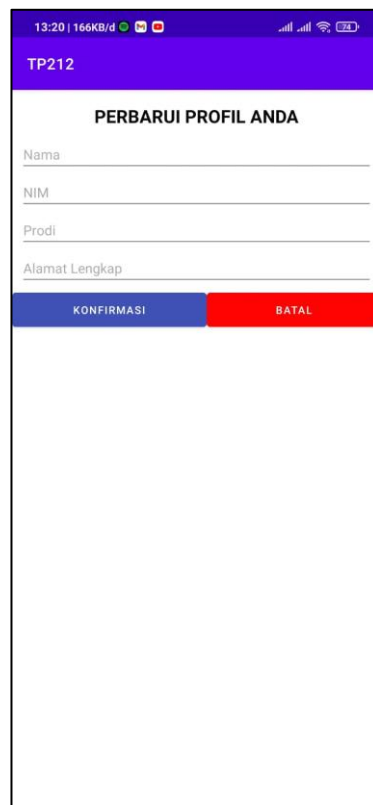
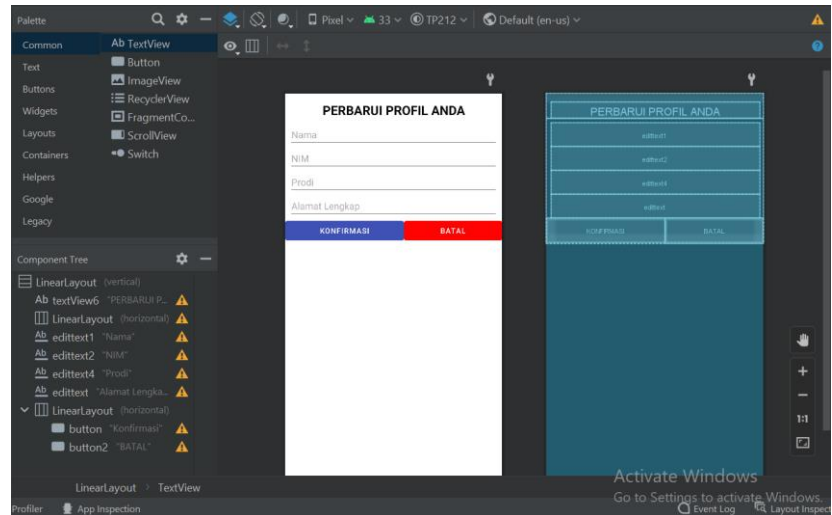
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="50"
        android:backgroundTint="#3F51B5"
        android:text="Konfirmasi" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="50"
        android:backgroundTint="#FF0303"
        android:text="BATAL" />
</LinearLayout>

</LinearLayout>

```

c. Hasil



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

Jawab :

a. Analisa

Tampilan yang menggunakan constrain Layout dengan 3 komponen yaitu Text view, Edit Text, dan button. Yang penataan tampilannya menggunakan id tiap komponen.

b. SourceCode

```
<?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/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DAFTAR"
        android:textColor="#000000"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/et_nama_depan"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:hint="Nama Depan"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView7"
        tools:ignore="TouchTargetSizeCheck" />

    <EditText
        android:id="@+id/et_nama_belakang"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:hint="Nama Belakang"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/et_nama_depan"
        app:layout_constraintTop_toBottomOf="@+id/textView7"
        tools:ignore="TouchTargetSizeCheck" />
```

```

<EditText
    android:id="@+id/et_email"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:hint="Alamat Email"
    android:inputType="textEmailAddress"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/et_nama_depa
n"
    tools:ignore="TouchTargetSizeCheck" />

<EditText
    android:id="@+id/et_password"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

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

<EditText
    android:id="@+id/et_konfirmasi_password"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:hint="Konfirmasi Password"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/et_password"
    tools:ignore="TouchTargetSizeCheck" />

<TextView
    android:id="@+id/textView8"
    android:layout_width="395dp"
    android:layout_height="20dp"
    android:layout_marginTop="20dp"
    android:gravity="center"
    android:text="silahkan cek kembali sebelum
konfirmasi"

app:layout_constraintTop_toBottomOf="@+id/et_konfirmas
i_password"
    tools:layout_editor_absoluteX="8dp"
    tools:ignore="MissingConstraints" />

```

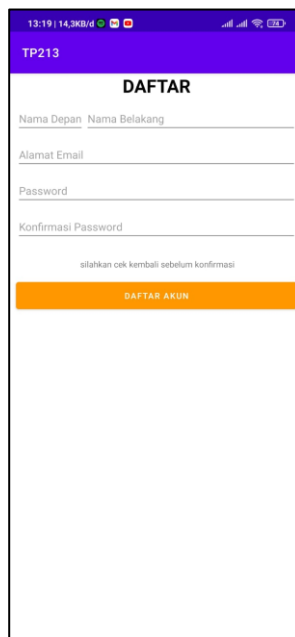
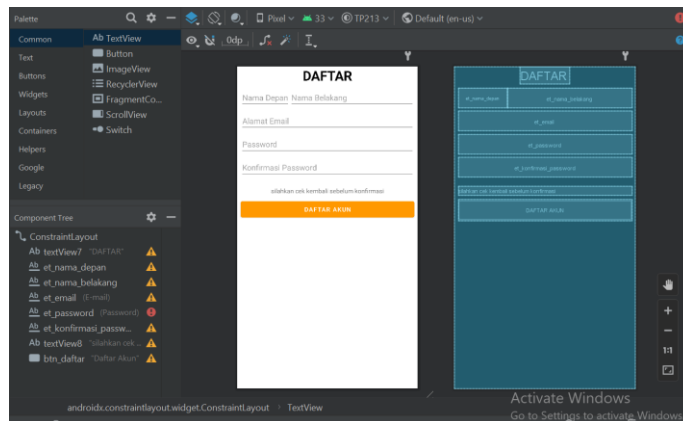
```

<Button
    android:id="@+id/btn_daftar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:backgroundTint="#FF9800"
    android:text="Daftar Akun"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView8"
/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

### c. Hasil



### 3.2 Komponen Widget View

Soal :

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

Jawab :

- a. Analisa

Project yang telah dibuat menggunakan kmponen-komponen yang telah disebutkna diatas, akan tetapi terdapat Radio grup untuk menentukan gender antara laki- laki dan perempuan. Tidak hanya itu terdapat juga checkbox yang digunakan untuk memilih hobi. Untuk memunculkan Toast berada pada class MainActivity.kt yang didalamnya dituliskan fucntion supaya Toast nya bisa berjalan.

- b. 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:layout_height="match_parent"
tools:context=".MainActivity">
    <TextView
        android:id="@+id/txtNama"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="36dp"
        android:text="Nama : "
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtGender"
        android:layout_width="87dp"
        android:layout_height="52dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="14dp"
        android:text="Gender :"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
```

```

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

    <TextView
        android:id="@+id/txtHobi"
        android:layout_width="73dp"
        android:layout_height="28dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:gravity="center"
        android:text="Hobby :"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"

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

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="50dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toEndOf="@+id/txtGender"

app:layout_constraintTop_toBottomOf="@+id/editNama"
    tools:ignore="MissingConstraints">

    <RadioButton
        android:id="@+id/rbLK"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Laki-Laki"
        android:textSize="16sp"
        android:textStyle="bold" />

    <RadioButton
        android:id="@+id/rbPR"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Perempuan"
        android:textSize="16sp"
        android:textStyle="bold" />
</RadioGroup>

<CheckBox
    android:id="@+id/cbCook"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="4dp"
    android:layout_marginEnd="24dp"
    android:text="Cooking"
    android:textSize="18sp"
    android:textStyle="bold"

```



```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toEndOf="@+id/txtHobi"

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

<CheckBox
    android:id="@+id/cbShop"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="12dp"
    android:layout_marginEnd="24dp"
    android:text="Shopping"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/txtHobi"

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

<CheckBox
    android:id="@+id/cbTraveling"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="12dp"
    android:layout_marginEnd="24dp"
    android:text="Traveling"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/txtHobi"
    app:layout_constraintTop_toBottomOf="@+id/cbShop"

/>

<CheckBox
    android:id="@+id/cbSport"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="12dp"
    android:layout_marginEnd="24dp"
    android:text="Sport"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/txtHobi"
    app:layout_constraintTop_toBottomOf="@+id/cbCook"

/>

<EditText
    android:id="@+id/editNama"
    android:layout_width="261dp"
    android:layout_height="40dp"
    android:layout_marginStart="16dp"

```

```

        android:layout_marginTop="28dp"
        android:layout_marginEnd="24dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:minHeight="50dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toEndOf="@+id/txtNama"
        app:layout_constraintTop_toTopOf="parent" />

        <ImageButton
            android:id="@+id/imgButtonSave"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="24dp"
            android:backgroundTint="#3F51B5"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/cbTraveling"
            app:srcCompat="@android:drawable/ic_menu_save"
            tools:ignore="MissingConstraints" />

    </androidx.constraintlayout.widget.ConstraintLayout>

```

- MainActivity.kt

```

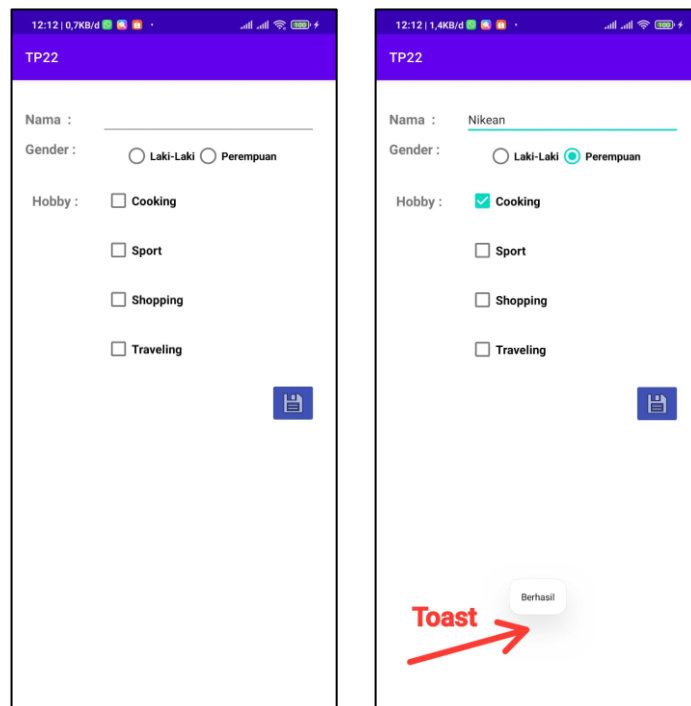
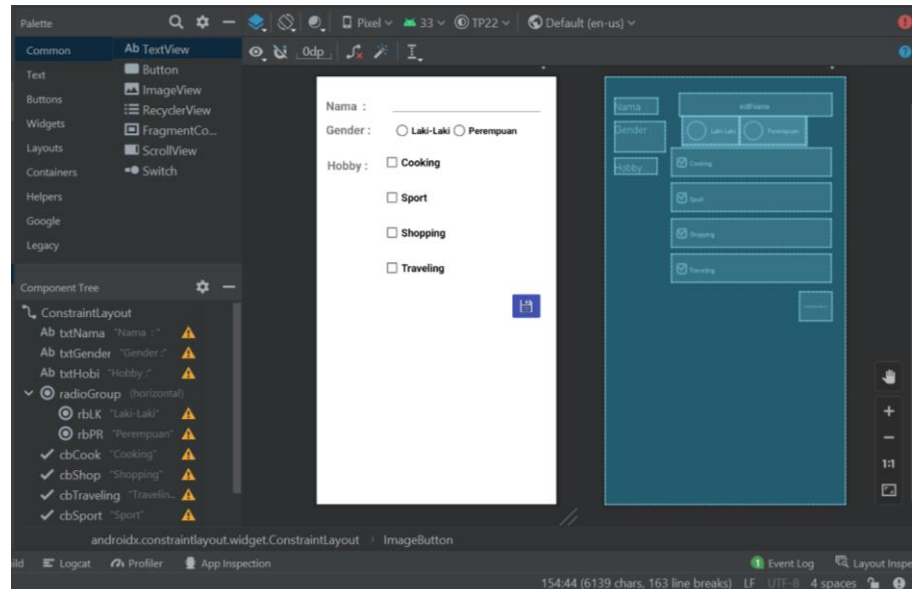
package com.example.tp22

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ImageButton
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val imgBtn =
            findViewById<ImageButton>(R.id.imgButtonSave)
        imgBtn.setOnClickListener {
            Toast.makeText(this@MainActivity,
                "Berhasil", Toast.LENGTH_SHORT).show()
        }
    }
}

```

### c. Hasil



## 3.3 Recycler View

Soal :

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

Jawab:

- a. Analisa

Project yang telah dibuat adalah menggunakan RecyclerView yang dihubungkan dengan beberapa class untuk menampilkan list data. Dan untuk menampilkan Toast terdapat function sendiri dan terdapat pada file MainActivity.

b. 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:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        tools:listitem="@layout/card_main"
        tools:itemCount="5"

        app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Card\_main.xml

```
<?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/human"

app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
        </ImageView>
        <TextView
            android:id="@+id/C"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/black"
            android:text="Halo"
            android:textSize="24sp"
            android:textStyle="bold"
            app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintBottom_toBottomOf="parent"

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

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

```

- MainActivity.kt

```

package com.example.bismillah

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.recyclerview.widget.RecyclerView
import kotlin.collections.List
import java.util.*

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

    }

    private fun cardAdaptor() {
        val card = listOf<ModelAdapter>(
            ModelAdapter(1,"Andi", R.drawable.human) ,
            ModelAdapter(2,"Anton", R.drawable.human),

```

```

        ModelAdapter(3, "Aris", R.drawable.human),
        ModelAdapter(4, "Jono", R.drawable.human),
        ModelAdapter(5, "Joni", R.drawable.human)
    )
    val mainadapter = MainAdapter(card, object :
MainAdapter.OnAdapterListener{
        override fun onClick(model: ModelAdapter)
    {
            val tampil = "Halo, ${model.name}"
            Toast.makeText(applicationContext,
tampil, Toast.LENGTH_SHORT).show()
        }
    })

findViewById<RecyclerView>(R.id.recyclerView).adapter
= mainadapter
    }
}

```

- MainAdapter.kt

```

package com.example.bismillah

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 MainAdapter(
    private val listName: List<ModelAdapter>,
    private val listener: OnAdapterListener
) : RecyclerView.Adapter<MainAdapter.ViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): MainAdapter.ViewHolder {
        return ViewHolder(
LayoutInflater.from(parent.context).inflate(R.layout.c
ard_main, parent, false)
        )
    }

    override fun onBindViewHolder(holder:
MainAdapter.ViewHolder, position: Int) {

        val model = listName[position]
        holder.imageView.setImageResource(model.image)
        holder.textView.text = model.name
        holder.itemView.setOnClickListener {
            listener.onClick(model)
        }
    }

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

```

```

    }

    class ViewHolder(view: View) :
        RecyclerView.ViewHolder(view) {
            val textView =
                view.findViewById<TextView>(R.id.textView)
            val imageView =
                view.findViewById<ImageView>(R.id.imageView)
        }

    interface OnAdapterListener {
        fun onClick(model: ModelAdapter)
    }
}

```

- ModelAdapter.kt

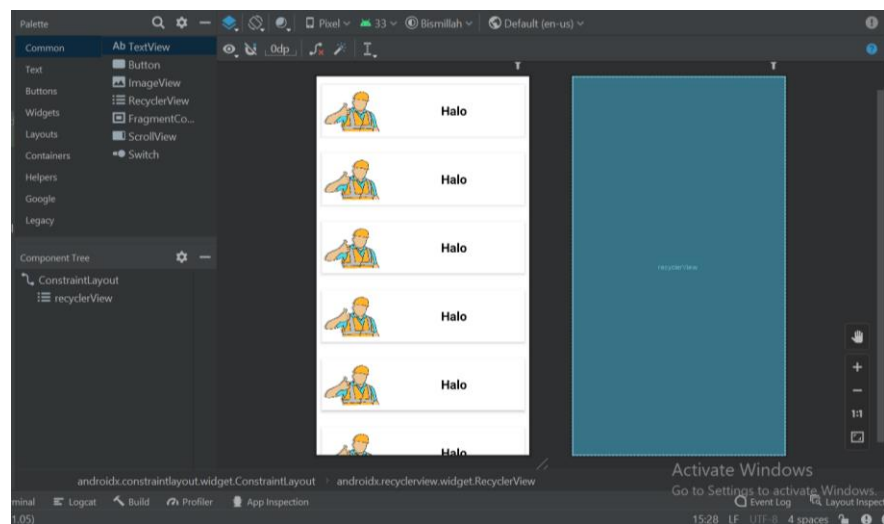
```

package com.example.bismillah

data class ModelAdapter(
    val id : Int,
    val name: String,
    val image: Int
)

```

c. Hasil



## **BAB IV**

### **IMPLEMENTASI**

#### **4.1 Tugas Praktikum**

##### **4.1.1 Layout**

Buat Project pada perangkat komputer anda untuk mengimplementasikan layout dengan berbagai bentuk tampilan.

##### **4.1.2 Komponen Widget View**

Analisislah atribut komponen untuk constrain layout.

##### **4.1.3 Recycler View**

Buat aplikasi baru dengan menerapkan RecyclerView.

#### **Jawaban :**

##### **a. Analisa**

Dalam project ini terdapat 1 project akan tetapi didalamnya terdapat layout, komponen widget dan Recycle view.

- Layout terdapat pada file each\_warna.xml yang dimana berisi ImageView dan TextView.
- Komponen Widget terdapat pada activity\_detail.xml dimana terdapat ImageView dan TextView dalam komponen Linear Layout
- Recycler View terdapat pada activity\_main.xml yang berisi list nama warna yang dihubungkan pada class MainActivity supaya dapat di Scroll keatas kebawah dengan jumlah data yang banyak.

##### **b. 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:layout_height="match_parent"
    tools:context=".DetailActivity">
```



```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
<ImageView
    android:id="@+id/detailActivityIv"
    android:layout_width="0dp"
    android:layout_height="250dp"
    android:scaleType="fitXY"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"

app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/merah" />

<TextView
    android:id="@+id/detailActivityTv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Merah"
    android:textColor="@color/black"
    android:layout_marginStart="16dp"
    android:textSize="24dp"
    android:textStyle="bold"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/detailAct
ivityIv" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

- Each\_warna.xml

```

<?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/merah"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintBottom_toBottomOf="parent">
    </ImageView>
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:text="Merah"
        android:textSize="24sp"
        android:textStyle="bold"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintBottom_toBottomOf="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>

```

- Activity\_detail.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:layout_height="match_parent"
      tools:context=".DetailActivity">

      <LinearLayout
          android:layout_width="match_parent"
          android:layout_height="match_parent"/>
  
```

```

        <ImageView
            android:id="@+id/detailActivityIv"
            android:layout_width="0dp"
            android:layout_height="250dp"
            android:scaleType="fitXY"

            app:layout_constraintEnd_toEndOf="parent"

            app:layout_constraintHorizontal_bias="0.5"

            app:layout_constraintStart_toStartOf="parent"

            app:layout_constraintTop_toTopOf="parent"
            app:srcCompat="@drawable/merah" />

        <TextView
            android:id="@+id/detailActivityTv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="Merah"
            android:textColor="@color/black"
            android:layout_marginStart="16dp"
            android:textSize="24dp"
            android:textStyle="bold"

            app:layout_constraintStart_toStartOf="parent"

            app:layout_constraintTop_toBottomOf="@id/detailA
ctivityIv" />

</androidx.constraintlayout.widget.ConstraintLay
out>

```

- MainActivity.kt

```

• 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.*

class MainActivity : AppCompatActivity() {

    private lateinit var recyclerView:
RecyclerView
    private lateinit var warnaList:
ArrayList<Warna>
    private lateinit var warnaAdapter:
WarnaAdapter

```

```

        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)

            warnaList = ArrayList()

            warnaList.add(Warna(R.drawable.merah,
            "Merah"))

            warnaList.add(Warna(R.drawable.kuning, "Kuning"))
            warnaList.add(Warna(R.drawable.hijau,
            "Hijau"))
            warnaList.add(Warna(R.drawable.biru,
            "Biru"))
            warnaList.add(Warna(R.drawable.oren,
            "Oren"))
            warnaList.add(Warna(R.drawable.ungu,
            "Ungu"))
            warnaList.add(Warna(R.drawable.hitam,
            "Hitam"))
            warnaList.add(Warna(R.drawable.coklat,
            "Coklat"))

            warnaAdapter= WarnaAdapter(warnaList)
            recyclerView.adapter = warnaAdapter

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

```

- Warna.kt

```

• package com.example.praktikum2

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

data class Warna(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<Warna> {
    override fun createFromParcel(parcel:
Parcel): Warna {
        return Warna(parcel)
    }

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

```

- WarnaAdapter.kt

```

• 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 WarnaAdapter(private val
warnaList:ArrayList<Warna> )
:
RecyclerView.Adapter<WarnaAdapter.WarnaViewHolde
r>(){

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

    class WarnaViewHolder(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):
WarnaAdapter.WarnaViewHolder {
        val view =
LayoutInflater.from(parent.context).inflate(R.la
yout.each_warna, parent, false)
        return WarnaViewHolder(view)
    }
}

```

```

    }

    override fun onBindViewHolder(holder:
WarnaViewHolder, position: Int) {
        val warna = warnaList[position]

        holder.imageView.setImageResource(warna.image)
        holder.textView.text = warna.name

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

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

```

- DetailActivity.kt

```

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

            textView.text = warna.name

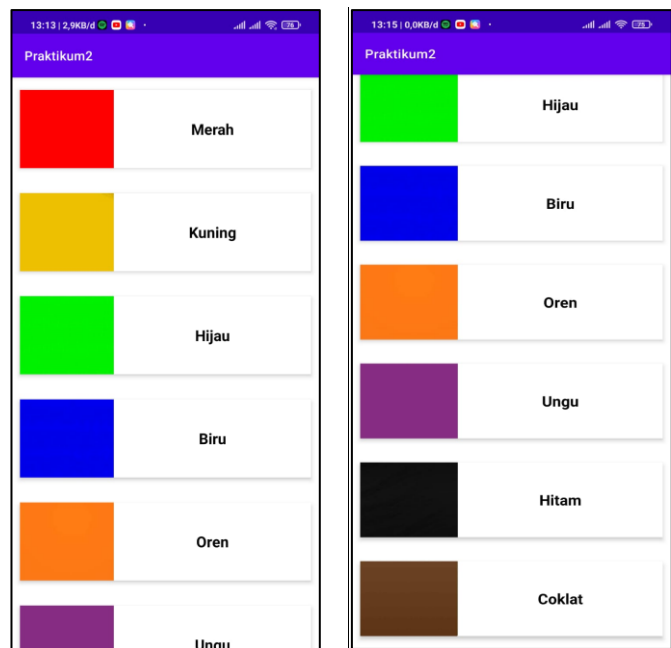
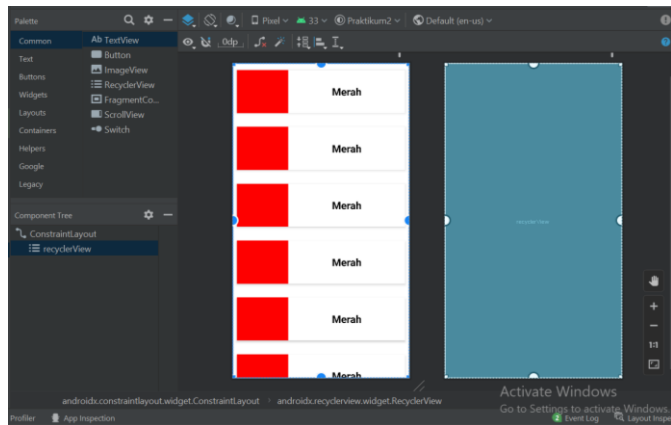
            imageView.setImageResource(warna.image)

        }
    }
}

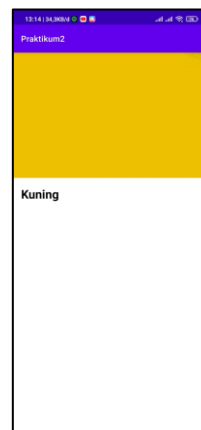
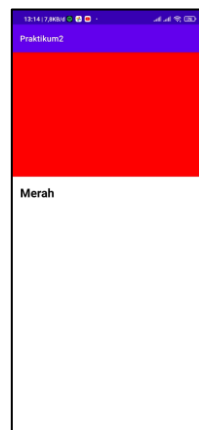
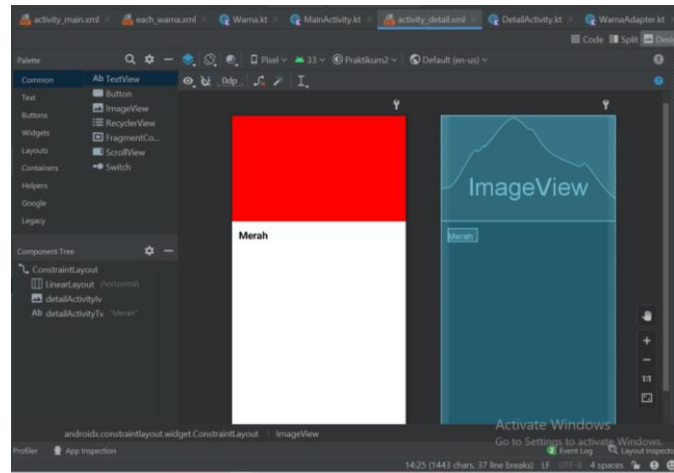
```

c. Hasil

- Halaman Utama



- Halaman Berikutnya





## **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**

- 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.