

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



NAMA	: CHERYL ALMIRAH AZMI
N.R.P	: 210441100130
DOSEN	: ACHMAD ZAIN NUR, S.Kom,M.T.
ASISTEN	: MUHAMMAD YAFIE ANWARY RAHMAN
TGL PRAKTIKUM	: 31 MARET 2023

Disetujui : 06 April 2023
Asisten

M. YAFIE ANWARY RAHMAN
19.04.411.00052



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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan aplikasi Android, salah satu hal penting yang harus dipahami adalah konsep tata letak (layout) dan tampilan widget (widget view) yang digunakan dalam membangun antarmuka pengguna (user interface) aplikasi. Layout adalah cara untuk menentukan posisi, ukuran, dan tata letak elemen UI (user interface) pada halaman aplikasi Android. Android Studio menyediakan beberapa jenis tata letak yang dapat digunakan, seperti LinearLayout, RelativeLayout, FrameLayout, dan lainnya. Setiap jenis tata letak memiliki cara kerja dan penggunaan yang berbeda-beda.

Widget view adalah komponen UI yang digunakan untuk menampilkan informasi dan interaksi dengan pengguna dalam aplikasi Android. Contohnya adalah tombol (button), teks (text view), gambar (image view), dan sebagainya. Setiap widget view memiliki karakteristik dan properti tertentu yang dapat disesuaikan dengan kebutuhan aplikasi.

Recycler view adalah salah satu komponen tampilan yang digunakan untuk menampilkan daftar data dalam aplikasi Android. Recycler view memungkinkan penggunaan memori yang efisien dengan hanya menampilkan data yang sedang terlihat dan memungkinkan pengguna untuk menggulir daftar data yang sangat besar dengan cepat. Recycler view juga memiliki kemampuan untuk mengatur ulang tampilan item daftar secara otomatis.

1.2 Tujuan

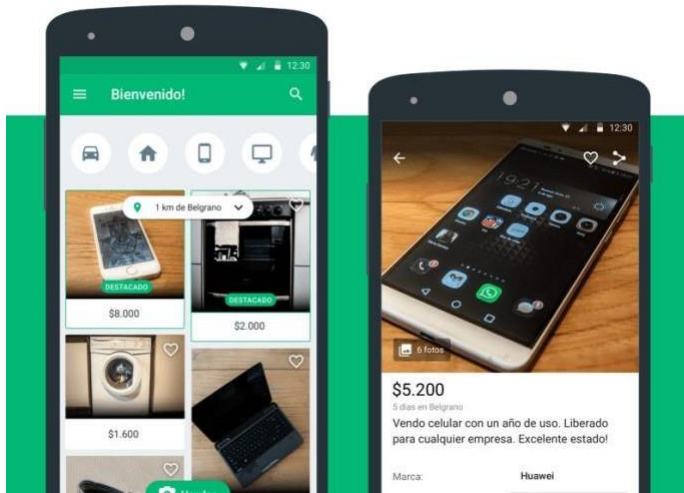
- Membuat Layout dengan Linear Layout dan Constraint Layout
- Mampu menggunakan Widget View (masukan) untuk membuat aplikasi sederhana
- Merepresentasikan data dengan menggunakan komponen recylerview

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 :

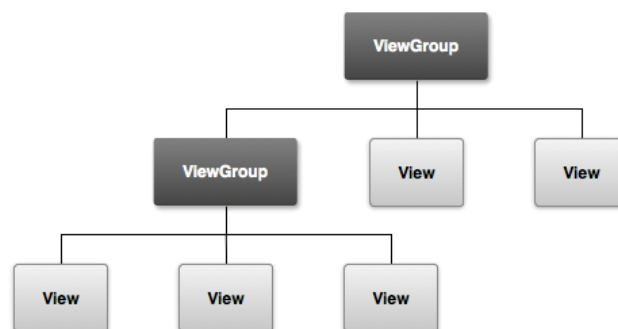
- **TextView**, komponen yang berguna untuk menampilkan teks ke layar.
- **Button**, komponen yang membuat pengguna dapat berinteraksi dengan cara ditekan untuk melakukan sesuatu.
- **ImageView**, Komponen untuk menampilkan gambar.
- **ListView**, komponen untuk menampilkan informasi dalam bentuk list.
- **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 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 viewgroup seperti ini :

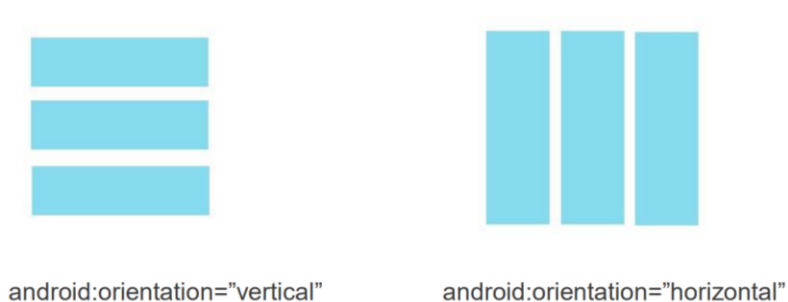
```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:orientation="vertical" >
7.     <TextView android:id="@+id/text"
8.         android:layout_width="wrap_content"
9.         android:layout_height="wrap_content"
10.        android:text="I am a TextView" />
11.     <Button android:id="@+id/button"
12.        android:layout_width="wrap_content"
13.        android:layout_height="wrap_content"
14.        android:text="I am a Button" />
15. </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 berguna untuk menentukan porsi ukuran view dalam sebuah ruang (space) yang tersedia.



Constrain Layout. Apa itu 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.

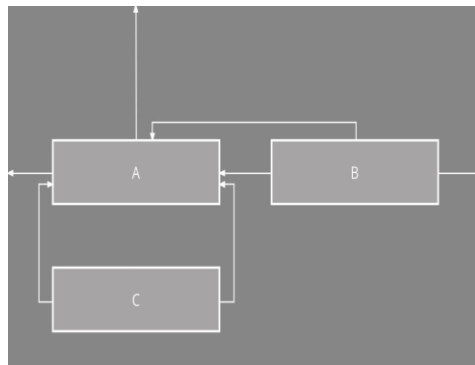
`ConstraintLayout` memiliki kesamaan dengan `RelativeLayout`. Dalam penggunaan semua view yang berada di dalamnya disusun berhubungan antara parent dan view lainnya. Tapi `ConstraintLayout` lebih fleksibel dari `RelativeLayout` dan mudah digunakan dengan dukungan `Layout Editor` pada `Android Studio`.

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.

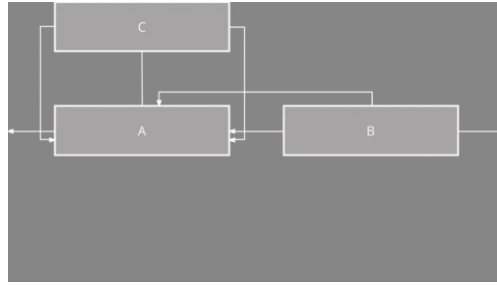
Kenapa gerangan? Karena setelah ditambahkan, view tersebut tidak memiliki constraint yang menghubungkannya dengan parent layout atau view lainnya. Sehingga ketika dijalankan, posisi dari view tersebut akan berada di bagian atas sebelah kiri.

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.

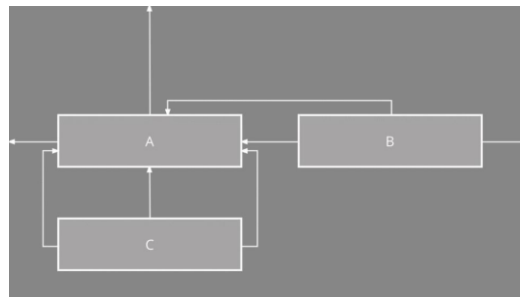
Setiap view setidaknya memiliki satu vertikal dan horizontal constraint. Misal kita memiliki sebuah layout dengan tampilan pada Layout Editor seperti berikut:



Susunan tampilan di atas akan terlihat normal. Tidak ada yang salah di Layout Editor. Tapi jika kita perhatikan seksama, view C diatas hanya memiliki horizontal constraint yang diatur sejajar dengan view A. Sehingga ketika jika kita coba menjalankannya, sama seperti yang disebutkan 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 Widge 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.

Event Handling

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

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

Event Listener

Event listener merupakan antarmuka di kelas View yang berisi method callback tunggal. Method ini akan dipanggil oleh framework Android jika View

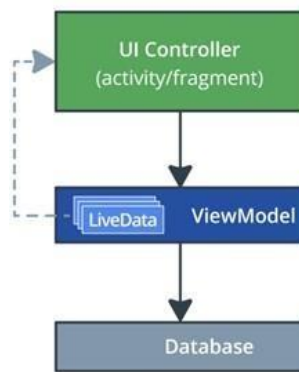
yang telah didaftarkan dengan listener dipicu oleh interaksi pengguna dengan item dalam UI.

Yang juga disertakan dalam antarmuka event listener adalah method callback berikut ini:

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

2.3 Recycler View

`RecyclerView` adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan UI controller, `ViewModel`, dan `LiveData`.



Menampilkan list atau grid data adalah salah satu tugas UI paling umum di Android. Daftar bervariasi dari yang sederhana hingga yang sangat kompleks. Daftar tampilan teks mungkin menampilkan data sederhana, seperti daftar belanja. Daftar yang kompleks, seperti daftar tujuan liburan yang beranotasi, dapat menunjukkan kepada pengguna banyak detail di dalam scrolling grid dengan header. Untuk mendukung semua kasus penggunaan ini, Android menyediakan widget RecyclerView.

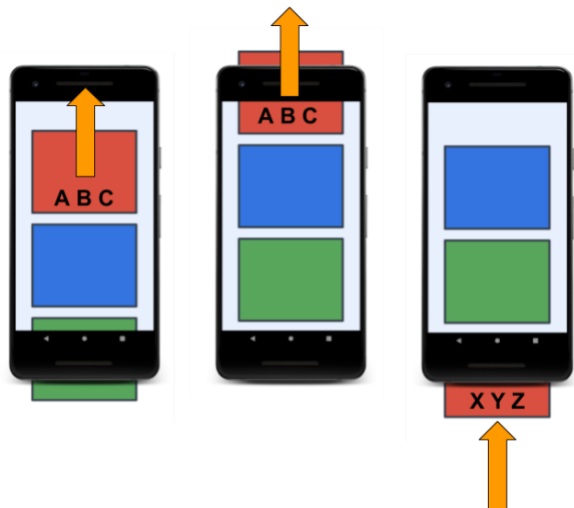


Manfaat terbesar dari RecyclerView adalah sangat efisien untuk daftar besar:

- Secara default, RecyclerView hanya berfungsi untuk memproses atau menggambar item yang saat ini terlihat di layar. Misalnya, jika list memiliki seribu elemen tetapi hanya 10 elemen yang terlihat, RecyclerView hanya berfungsi untuk menggambar 10 item di layar. Ketika pengguna melakukan scroll, RecyclerView mengetahui item baru apa yang seharusnya ada di layar dan tidak cukup berfungsi untuk menampilkan item itu.

- Ketika suatu item scroll dari layar, tampilan item tersebut didaur ulang. Itu berarti item diisi dengan konten baru yang scroll ke layar. Perilaku RecyclerView ini menghemat banyak waktu pemrosesan dan membantu scroll list dengan lancar.
- Ketika suatu item berubah, alih-alih menggambar ulang seluruh daftar, RecyclerView dapat memperbarui satu item itu. Ini adalah keuntungan efisiensi yang sangat besar ketika menampilkan daftar item kompleks!

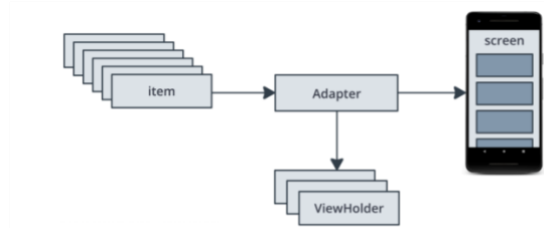
Dalam urutan yang ditunjukkan di bawah ini, kita dapat melihat bahwa satu tampilan telah diisi dengan data, ABC. Setelah itu tampilan bergulir dari layar, RecyclerView menggunakan kembali tampilan untuk data baru, XYZ.



Adapter pattern

Jika kita pernah bepergian antar negara yang menggunakan soket listrik yang berbeda, kita mungkin 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 wadah untuk 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 mengetahui cara menampilkan data di layar.

BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan perbedaan Constraint layout dan linear layout serta kelebihan dan kekurangan dari keduanya!
2. Apa saja manfaat Recycle view?
3. Jelaskan perbedaan antara list view dan recycle view!
4. Komponen apa saja yg diperlukan untuk membuat recycle view?

3.2 Jawaban

1. -Constraint layout lebih fleksibel dan mudah untuk dibuat, dapat langsung menempatkan objek tanpa adanya nested layout. Kelebihan dan kekurangan constraint layout yakni fleksibel dan sangat membantu dalam membangun tampilan aplikasi yg cukup kompleks, sedangkan kekurangannya yaitu tidak bisa digunakan dalam tampilan bertingkat.
-Linear layout menempatkan view secara horizontal atau vertikal, bergantung pada kode yg digunakan. Kelebihan dari linear layout yaitu layouting yg lebih mudah karena arah dan elemen anaknya sudah ditentukan oleh parentnya, sedangkan kekurangannya yaitu tidak bisa menyusun widget dengan bebas karena linear layout membatasi layout hanya pada satu arah saja yaitu horizontal / vertikal.
2. Recycle view bermanfaat untuk memudahkan kita dalam menampilkan kumpulan data dalam jumlah besar secara efisien. Misal, kita menyediakan data dan menentukan tampilan setiap item, dan library recycle view secara dinamis membuat elemen saat diperlukan, sesuai namanya, recycle view mendaur ulang elemen individual tersebut.
3. Recycle view adalah komponen tampilan (widget) yg lebih canggih dibandingkan pendahulunya listview. Apabila listview digunakan, objek dari tiap item akan dibuat semuanya dari awal sampai akhir, sedangkan bila menggunakan recycle view, objek yg dibuat hanya sebatas ukuran layar dan beberapa di atas dan di bawahnya saja.
4. Komponen-komponen yg diperlukan yaitu:
 - Data untuk ditampilkan
 - Mesin virtual recycle view didefinisikan dalam file layout.
 - Layout untuk satu item data.
 - Layout manager, menangani organisasi (layout) komponen UI dalam tampilan.
 - View holder, berisi informasi tampilan untuk menampilkan satu item dan layout item.
 - Adaptor, menghubungkan data ke recycle view.

BAB IV

IMPLEMENTASI

4.1 Implementasi

1. Layout dan Widget View. Buat satu Project pada perangkat komputer anda untuk mengimplementasikan linear, relative, frame, dan table layout dan gunakan beberapa widget view.

- a. Souce Code

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"
    android:orientation="vertical"
    android:background="#F1F6F5"
    tools:context=".MainActivity">

    <!-- Frame Layout -->
    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            android:layout_width="450dp"
            android:layout_height="250dp"
            android:src="@drawable/bg6"
            android:layout_gravity="center"/>

    <!-- Linear Layout -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center">
```

```

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="200dp"
    android:layout_height="90dp"
    app:srcCompat="@drawable/logo_utm"
    android:layout_marginTop="25dp"/>
<TextView
    android:id="@+id/judul"
    android:layout_width="match_parent"
    android:layout_height="38dp"
    android:layout_margin="5dp"
    android:gravity="center"
    android:text="Data Mahasiswa"
    android:textColor="@color/white"
    android:textSize="25sp"
    app:fontFamily="@font/poppins_bold" />
<TextView
    android:id="@+id/subjudul"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Universitas Trunojoyo
Madura"
    android:textColor="@color/white"
    android:textSize="20sp"
    app:fontFamily="@font/poppins_bold" />
</LinearLayout>
</FrameLayout>

<!-- Table Layout -->
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="30dp">
    <TableRow
        android:background="#4B56D2">
        <TextView

```

```

        android:text="NIM"
        android:layout_width="180dp"
        android:layout_column="1"
        android:layout_gravity="center"
        android:textColor="@color/white"
        app:fontFamily="@font/poppins_bold"/>
<TextView
    android:text="NAMA"
    android:layout_width="180dp"
    android:layout_column="2"
    android:layout_gravity="center"
    android:textColor="@color/white"
    app:fontFamily="@font/poppins_bold"/>
</TableRow>

<TableRow
    android:background="@color/white">
    <TextView
        android:text="210441100130"
        android:layout_column="1"
        android:layout_gravity="center"
        android:textColor="#808080"
        app:fontFamily="@font/poppins_medium"/>

        <TextView
            android:text="Cheryl Almirah Azmi"
            android:layout_column="2"
            android:layout_gravity="center"
            android:textColor="#808080"
            app:fontFamily="@font/poppins_medium"/>
    </TableRow >

<TableRow
    android:background="@color/white">
    <TextView
        android:text="210311100010"
        android:layout_column="1"

```



```

        android:layout_gravity="center"
        android:textColor="#808080"
        app:fontFamily="@font/poppins_medium"/>

        <TextView
            android:text="Marsa Rahma Adelia"
            android:layout_column="2"
            android:layout_gravity="center"
            android:textColor="#808080"
            app:fontFamily="@font/poppins_medium"/>
    </TableRow >

    <TableRow
        android:background="@color/white">
        <TextView
            android:text="2106311000080"
            android:layout_column="1"
            android:layout_gravity="center"
            android:textColor="#808080"
            app:fontFamily="@font/poppins_medium"/>

            <TextView
                android:text="Jannatul Firdausiyah"
                android:layout_column="2"
                android:layout_gravity="center"
                android:textColor="#808080"
                app:fontFamily="@font/poppins_medium"/>
        </TableRow >
    </TableLayout>

    <!-- Relative Layout -->
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp">

        <TextView

```

```
        android:id="@+id/tNIM"
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:textSize="18sp"
        android:textColor="@color/black"
        android:text="NIM"
        app:fontFamily="@font/poppins_medium"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"/>
```

<EditText

```
        android:id="@+id/etNIM"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:background="@color/white"
        android:hint="Masukkan NIM"
        android:textSize="15sp"
        android:paddingLeft="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_below="@+id/tNIM"
        android:textColor="#808080"
        app:fontFamily="@font/poppins_medium"/>
```

<TextView

```
        android:id="@+id/tNama"
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:textSize="18sp"
        android:textColor="@color/black"
        android:text="Nama"
        app:fontFamily="@font/poppins_medium"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:layout_below="@+id/etNIM"
```

```
        android:layout_marginRight="20dp"/>
```

```
<EditText
```

```
    android:id="@+id/etNama"
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:layout_below="@+id/tNama"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:background="@color/white"
    android:hint="Masukkan nama"
    android:paddingLeft="10dp"
    android:textSize="15sp"
    android:textColor="#808080"
    app:fontFamily="@font/poppins_medium" />
```

```
<Button
```

```
    android:id="@+id/bSubmit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="submit"
    android:layout_below="@+id/etNama"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="60dp"
    android:layout_marginRight="18dp"
    android:textColor="@color/white"
    app:backgroundTint="#4B56D2"/>
```

```
<Button
```

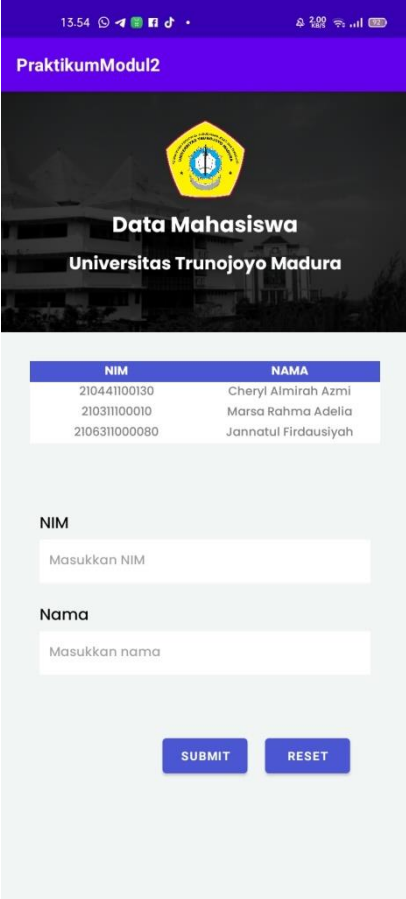
```
    android:id="@+id/bReset"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="reset"
    android:layout_below="@+id/etNama"
    android:layout_toRightOf="@+id/bSubmit"
    android:layout_marginTop="60dp"
```

```

        android:textColor="@color/white"
        app:backgroundTint="#4B56D2"/>
    </RelativeLayout>
</LinearLayout>

```

b. Output



NIM	NAMA
210441100130	Cheryl Almirah Azmi
210311100010	Marsa Rahma Adelia
2106311000080	Jannatul Firdausiyah

NIM

Masukkan NIM

Nama

Masukkan nama

SUBMIT RESET

c. Penjelasan program

Aplikasi ini dibuat dengan beberapa jenis *layout*. *Frame Layout* disini digunakan pada bagian atas (gambar utm, logo, dan judul). *Table layout* digunakan untuk membuat tabel pada data mahasiswa agar terlihat lebih tertata. *Linear layout* disini digunakan pada bagian keseluruhan project dan juga pada bagian penataan logo dan judul. Sedangkan untuk *relative layout* disini digunakan pada bagian form penginputan dan juga button.

2. Recycler View. Buat aplikasi baru dengan menerapkan RecyclerView.

a. Souce Code

build.gradle (:app)

```
implementation 'androidx.recyclerview:recyclerview:1.1.0'
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"
    android:orientation="vertical">

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

recycle_main_template.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"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_margin="8dp"
    android:background="@color/white"
    app:cardElevation="6dp">

    <LinearLayout
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:id="@+id/img_view"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:src="@mipmap/ic_launcher"
            android:layout_marginRight="8dp"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/txt_title"
                android:text="title"
                android:textColor="#000000"
                android:textSize="18sp"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/txt_sub_title"
                android:text="sub title"/>
        </LinearLayout>
    </LinearLayout>
</androidx.cardview.widget.CardView>

```

univ.kt

```

package com.example.praktikummodul2_2

class univ (val imgView: Int, val txtTitle: String, val
txtSubTitle: String)

```

MainActivity.kt

```

package com.example.praktikummodul2_2

```

```

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() {

    lateinit var recyclerView: RecyclerView
    lateinit var adapter: MyAdapter

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

        init()
        recyclerView.layoutManager =
LinearLayoutManager(this)
        recyclerView.adapter = adapter

    }
    private fun init(){
        recyclerView = findViewById(R.id.recycler_view)
        var data = ArrayList<univ>()
        data.add(univ(R.drawable.ub,"Universitas
Brawijaya", "Malang, Jawa Timur"))
        data.add(univ(R.drawable.ugm,"Universitas Gadjah
Mada", "Yogyakarta"))
        data.add(univ(R.drawable.unej,"Universitas
Jember", "Jember, Jawa Timur"))
        data.add(univ(R.drawable.unpad,"Universitas
Padjajaran", "Jatinangor"))
        data.add(univ(R.drawable.ub,"Universitas
Brawijaya", "Malang, Jawa Timur"))
        data.add(univ(R.drawable.ugm,"Universitas Gadjah
Mada", "Yogyakarta"))
    }
}

```

```

        data.add(univ(R.drawable.unej,"Universitas
Jember", "Jember, Jawa Timur"))
        data.add(univ(R.drawable.unpad,"Universitas
Padjajaran", "Jatinangor"))
        data.add(univ(R.drawable.ub,"Universitas
Brawijaya", "Malang, Jawa Timur"))
        data.add(univ(R.drawable.ugm,"Universitas Gadjah
Mada", "Yogyakarta"))
        data.add(univ(R.drawable.unej,"Universitas
Jember", "Jember, Jawa Timur"))
        data.add(univ(R.drawable.unpad,"Universitas
Padjajaran", "Jatinangor"))

        adapter = MyAdapter(data)
    }
}

```

ViewHolder.kt

```

package com.example.praktikummodul2_2
import android.text.Layout
import android.view.LayoutInflater
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import java.util.zip.Inflater

class ViewHolder(inflater: LayoutInflater, parent:
ViewGroup):
RecyclerView.ViewHolder(inflater.inflate(R.layout.recycle
r_view_template,parent,false)) {
    private var imgView: ImageView? = null
    private var txtTitle: TextView? = null
    private var txtSubTitle: TextView? = null

    init {

```



```

        imageView = itemView.findViewById(R.id.img_view)
        txtTitle = itemView.findViewById(R.id.txt_title)
        txtSubTitle =
itemView.findViewById(R.id.txt_sub_title)
    }

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

```

MyAdapter.kt

```

package com.example.praktikummodul2_2

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

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

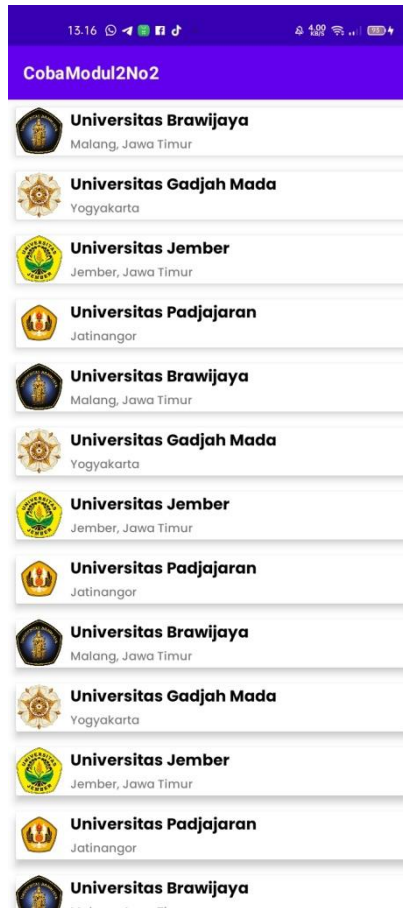
    override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
        holder.bind(data[position])
    }

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

```

```
}  
}
```

b. Output



c. Penjelasan program

Aplikasi ini dibuat dengan menggunakan RecyclerView. RecyclerView memudahkan untuk menampilkan kumpulan data dalam jumlah yang besar secara efisien, lebih modern dari ListView. Untuk membuat project RecyclerView, diperlukan adanya beberapa file lain sebagai pendukung dalam membangun project tersebut seperti ViewHolder dan Adapter.

BAB V

PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa dalam pengembangan aplikasi Android, penggunaan konsep tata letak (layout), widget view, dan recycler view sangat penting untuk membangun antarmuka pengguna yang efisien, responsif, dan mudah digunakan. Dengan memahami cara kerja dan penggunaan dari masing-masing konsep tersebut, pengembang dapat mengoptimalkan kinerja aplikasi dan memberikan pengalaman pengguna yang lebih baik.

Layout merupakan dasar dari tampilan UI pada aplikasi Android. Dengan memilih jenis tata letak yang tepat dan memanfaatkan properti yang disediakan, pengembang dapat membuat antarmuka pengguna yang menarik dan mudah dipahami oleh pengguna. Widget view juga sangat penting dalam membangun tampilan UI yang interaktif dan menarik. Pengembang dapat memanfaatkan properti dan metode yang tersedia pada masing-masing widget view untuk memenuhi kebutuhan aplikasi. Recycler view sangat membantu dalam menampilkan daftar data yang besar dengan cepat dan efisien. Dengan memanfaatkan fitur-fitur yang disediakan pada recycler view, pengembang dapat mengoptimalkan kinerja aplikasi dan memberikan pengalaman pengguna yang lebih baik.

Secara keseluruhan, penggunaan konsep tata letak (layout), widget view, dan recycler view menjadi kunci dalam membangun antarmuka pengguna yang efisien dan responsif pada aplikasi Android. Pengembang perlu memahami dengan baik konsep-konsep tersebut dan memanfaatkan properti dan metode yang tersedia untuk memenuhi kebutuhan aplikasi dan memberikan pengalaman pengguna yang lebih baik.

5.2 Kesimpulan

1. Android disusun dalam bentuk software stack yang terdiri dari aplikasi, sistem operasi, lingkungan run-time, middleware, layanan dan pustaka (library).
2. Aplikasi Android diciptakan dengan menggunakan satu atau lebih komponen bersama, yang dikenal sebagai Activity.
3. Aplikasi manifest file yang mengatur berbagai elemen dalam aplikasi adalah file Manifest. Berkas Manifest berbasis XML ini, menguraikan Activity, Service,
4. Terdapat beberapa IDE pada Android Studio yang nantinya akan digunakan selama proses pembuatan project, yaitu Open edit Run/Debug configurations dialog, Run App, Debug App, Profile App, Attach Debugger to Android Process, Sync Project with Gradles Files, AVD Manager, SDK Manager, dan Search Everywhere.