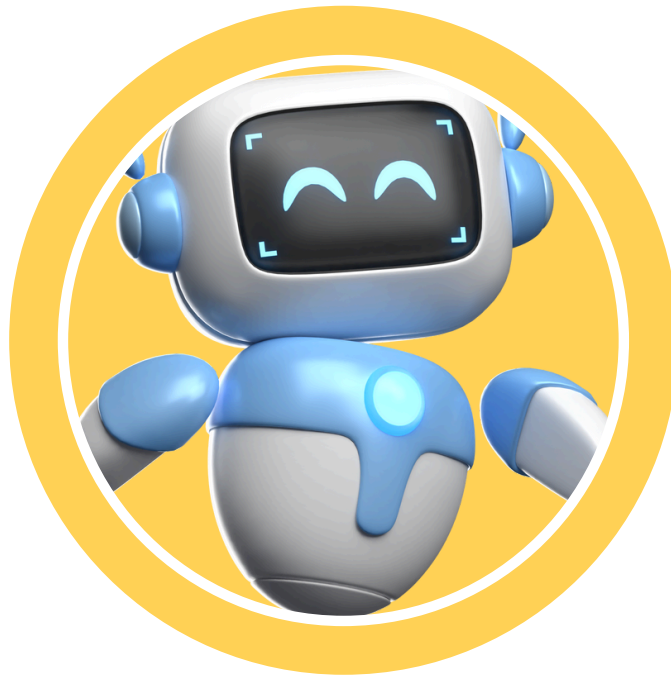




STMIK KAPUTAMA

PRAKTIKUM PEMROGRAMAN BERGERAK



Disusun Oleh:

Kristina Annatasia Br Sitepu, M.Kom
Ratih Puspadini, ST., M.Kom

PERTEMUAN 3 LAYOUT DESAIN

3.1 TUJUAN PEMBELAJARAN :

- a. Mahasiswa mampu memahami layout pada pemrograman android.
- b. Mahasiswa mampu membuat aplikasi Layout sederhana dan melakukan kompilasi ke emulator.

3.2 ALAT DAN BAHAN :

- a. Laptop/PC
- b. Android Studio
- c. Sistem Operasi Windows

3.3 MATERI

Aplikasi android dibangun oleh dua komponen utama yaitu layout dan business logic. Secara umum user interface (UI) aplikasi android berasal dari tampilan/layout file yang berekstensi xml (*main_android.xml*). File xml inilah yang handle tampilan aplikasi android yang dapat dilihat oleh pengguna aplikasi.

Layout merupakan tempat dimana kita meletakkan komponen yang dibutuhkan dalam aplikasi android sehingga tata letak dari komponen tersebut menjadi lebih teratur dan menarik. Android layout berisikan tata letak komponen dalam bentuk textView,buttonView, imageView dll yang dibuat dalam bentuk XML file.

Sementara business logic adalah berupa java file yang berfungsi untuk handle aspek event-driven dari aplikasi. Selain java bahasa pemrograman lain yang dapat digunakan untuk business logic dari aplikasi android adalah Kotlin.

Pada android studio telah disediakan fasilitas yang sangat membantu pada pengembangan android layout yaitu android layout editor. Layar ini memiliki dua mode yaitu **mode design** dan **mode text**. Mode design memungkinkan kita mendesign User Interface dengan menggunakan drag and drop dari android view. Sementara pada mode text user harus memasukkan sintaks-sintaks XML.

Secara umum setiap komponen atau elemen di xml memiliki **atribut** wajib dan tambahan. Atribut yang wajib dimiliki setiap elemen di xml adalah atribut *android:layout_width* dan atribut *android:layout_height*. Kemudian **nilai dari atribut** tersebut dapat berupa "*match_parent*" dan "*wrap_content*".

```

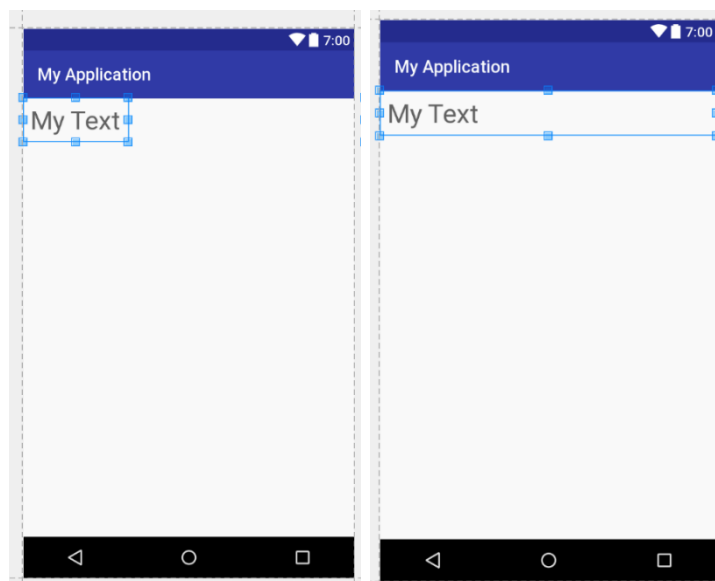
<TextView
    android:text="status"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="78dp"
    android:id="@+id/textView2" />

```

"*android:layout_width*" adalah tagging yang berisi tentang lebar layar dari elemen tersebut. Sedangkan "*android:layout_height*" adalah tagging yang berisi tentang tinggi layar dari elemen tersebut.

"*match_parent*" berarti ukuran dimensi sebuah View disesuaikan dengan ukuran layar secara horizontal jika pada *layout_width* dan vertikal jika pada *layout_height*.

"*wrap_content*" berarti ukuran dimensi sebuah View disesuaikan dengan ukuran konten didalamnya baik secara horizontal jika pada *layout_width* dan vertikal jika pada *layout_height*.



```

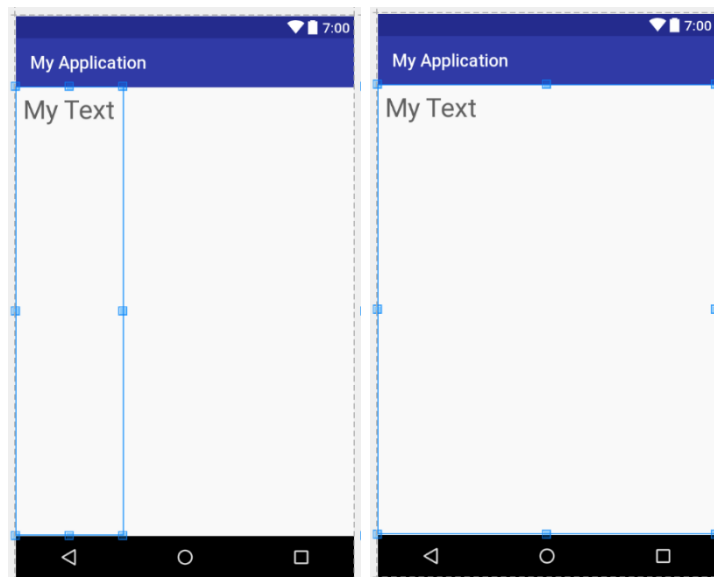
android:layout_width="wrap_content"
android:layout_height="wrap_content"

```

```

android:layout_width="match_parent"
android:layout_height="wrap_content"

```



`android:layout_width="wrap_content"` `android:layout_width="match_parent"`
`android:layout_height="match_parent"` `android:layout_height="match_parent"`

Untuk me-load atau mengakses XML Layout pada aplikasi yang kita jalankan, maka pada file java diterapkan syntax `setContentView(R.layout.namafile_xmlLayout)`.

```

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

```

Tujuan utama dari layout desain adalah untuk membuat antarmuka pengguna yang intuitif, mudah diakses, dan efektif dalam menyampaikan informasi atau berinteraksi dengan aplikasi. Layout yang baik mempertimbangkan faktor seperti responsivitas, aksesibilitas, kemudahan penggunaan, serta estetika visual.

1. Component Types (Jenis Komponen)

Di Android, komponen-komponen UI adalah elemen dasar yang digunakan untuk membuat antarmuka pengguna (UI). Komponen ini bisa digunakan untuk menampilkan informasi, mengambil input dari pengguna, atau memicu berbagai interaksi dalam aplikasi. Berikut adalah beberapa jenis komponen yang sering digunakan dalam pengembangan aplikasi Android:

- **TextView** : Digunakan untuk menampilkan teks statis pada layar.
 Contoh : Menampilkan judul atau informasi.
 Kode XML : `<TextView android:text="Hello World" />`

- **EditText** : Komponen input yang memungkinkan pengguna mengetik teks, biasanya digunakan untuk form atau pengisian data.
 Contoh : Kolom untuk input nama atau email.
 Kode XML : `<EditText android:hint="Enter your name" />`
- **Button** : Komponen yang dapat diklik untuk memicu suatu aksi atau event.
 Contoh : Tombol "Submit", "Login", atau "Next".
 Kode XML : `<Button android:text="Submit" />`
- **ImageView** : Digunakan untuk menampilkan gambar pada layar.
 Contoh : Logo aplikasi atau foto pengguna.
 Kode XML : `<ImageView android:src="@drawable/my_image" />`
- **CheckBox** : Sebuah kotak centang untuk memilih atau tidak memilih suatu opsi.
 Contoh : Pilihan "Ingat saya" pada layar login.
 Kode XML : `<CheckBox android:text="Remember Me" />`
- **RadioButton** : Digunakan untuk memilih satu opsi di antara beberapa pilihan dalam sebuah grup.
 Contoh : Pilihan "Jenis Kelamin" (Pria/Wanita).
 Kode XML :

```

<RadioGroup>
    <RadioButton android:text="Male" />
    <RadioButton android:text="Female" />
</RadioGroup>

```
- **ProgressBar**: Menampilkan status proses seperti saat sedang mengunduh file atau melakukan loading data.
 Kode XML : `<ProgressBar android:indeterminate="true" />`
- **RecyclerView** : Komponen yang sangat efisien untuk menampilkan daftar atau grid item yang banyak dan bisa di-scroll.
 Contoh : Daftar produk, daftar pesan, atau galeri foto.
 Kode XML : `<androidx.recyclerview.widget.RecyclerView />`

Setiap komponen memiliki properti yang dapat dikustomisasi melalui XML atau kode Java untuk menyesuaikan dengan kebutuhan aplikasi.

2. Hierarchy of Screen Elements (Hirarki Elemen Layar)

Pada Android, layout didefinisikan sebagai hirarki dari elemen-elemen UI. Hirarki ini menentukan bagaimana komponen UI disusun dan bagaimana tampilannya di layar.

- ViewGroup dan View:

View adalah elemen dasar UI yang bisa dilihat oleh pengguna, misalnya Button, TextView, ImageView.

ViewGroup adalah turunan dari View yang bertindak sebagai container untuk View lainnya. Contohnya, LinearLayout, RelativeLayout, dan ConstraintLayout.

Dalam hierarki layout, ada hubungan induk (parent) dan anak (child):

- Sebuah ViewGroup (misalnya LinearLayout) bisa memiliki beberapa komponen View di dalamnya sebagai anak.
- Parent-Child Relationship ini mengatur bagaimana tata letak komponen ditampilkan, seperti apakah komponen ditumpuk secara vertikal (atas ke bawah), horizontal (kiri ke kanan), atau mengikuti aturan constraint.

Contoh layout hirarki sederhana dalam XML:

```
<LinearLayout
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <!-- TextView sebagai child dari LinearLayout -->
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World" />

  <!-- Button sebagai child dari LinearLayout -->
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="Click Me" />
    </LinearLayout>
```

Dalam contoh di atas, LinearLayout adalah parent, dan TextView serta Button adalah child. Tata letaknya vertikal karena android:orientation diset ke "vertical".

3. Komponen Aplikasi : Service dan Activity

Android adalah sistem operasi berbasis komponen, di mana aplikasi terdiri dari berbagai komponen yang bekerja sama untuk memberikan pengalaman pengguna yang lengkap. Dua komponen penting adalah Activity dan Service.

- Activity

Activity adalah salah satu komponen dasar yang mewakili satu layar pada aplikasi. Setiap kali pengguna berinteraksi dengan aplikasi, biasanya mereka akan berpindah antar berbagai Activity.

Contoh: Layar login, layar beranda, layar detail produk.

Setiap Activity memiliki siklus hidup (lifecycle) seperti (materi pertemuan 2):

- onCreate(): Dipanggil saat Activity pertama kali dibuat.
- onStart(): Dipanggil saat Activity akan mulai terlihat di layar.
- onResume(): Dipanggil saat Activity mulai berinteraksi dengan pengguna.

Contoh kode Java dasar untuk Activity:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Menyambungkan layout XML
    }
}
```

Dalam kode di atas, MainActivity adalah sebuah Activity yang menampilkan layout yang didefinisikan di activity_main.xml.

- Service

Service adalah komponen yang berjalan di background untuk melakukan operasi jangka panjang tanpa interaksi dengan UI.

Misalnya, Service digunakan untuk memutar musik di latar belakang, mengunduh file, atau sinkronisasi data dengan server.

Contoh penggunaan:

- Aplikasi musik: Service berjalan untuk memutar musik bahkan ketika pengguna tidak sedang di dalam aplikasi.
- Aplikasi sinkronisasi: Mengunduh data dari server di background tanpa harus menunggu di layar.

4. Struktur XML

Layout dalam aplikasi Android biasanya ditulis dalam XML (Extensible Markup Language). Struktur XML ini menggambarkan bagaimana antarmuka pengguna disusun. Setiap komponen didefinisikan menggunakan tag XML yang spesifik, misalnya <TextView>, <Button>, <LinearLayout>, dll.

Berikut adalah contoh struktur XML sederhana untuk sebuah layout:

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:id="@+id/helloText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World" />

  <Button
    android:id="@+id/myButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click Me" />
</LinearLayout>
```


Penjelasan:

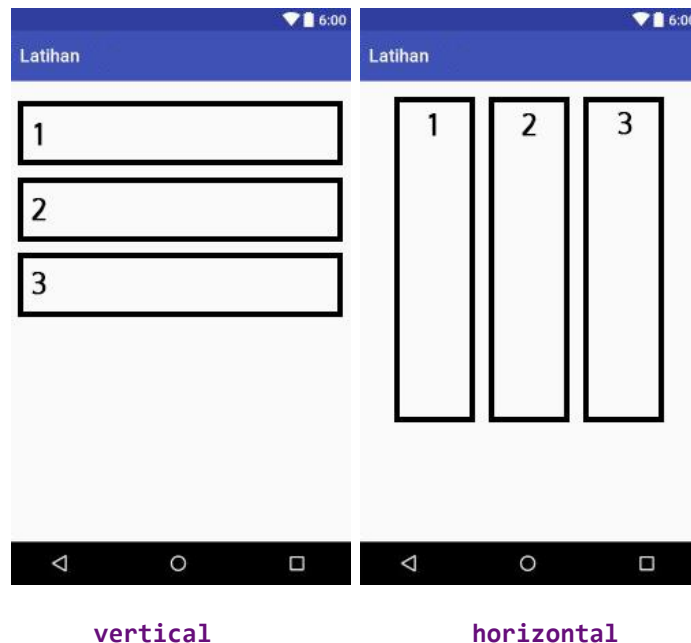
- ✓ `xmlns:android`: Namespace untuk properti Android.
- ✓ `layout_width` dan `layout_height`: Menentukan ukuran layout komponen. Bisa diatur sebagai `match_parent` (mengambil seluruh ruang yang tersedia) atau `wrap_content` (hanya sebesar konten yang ada).
- ✓ `id`: Digunakan untuk mengidentifikasi komponen agar bisa diakses dari kode Java.

5. Jenis-jenis layout

Pada bagian ini, kita akan membahas bagaimana menggunakan layout desain di Android, serta contoh implementasi setiap jenis layout. Semua contoh implementasi ditulis dalam XML untuk layout dan Java untuk logika pemrograman.

1) LinearLayout

LinearLayout merupakan komponen view bertipe viewgroup yang menjadi parent dari semua sub komponen view di dalamnya. **LinearLayout** akan membuat tampilan dengan konsep linear, dimana elemen atau komponen tampilan akan menempati posisi linear dengan elemen lainnya baik secara vertikal maupun horizontal. Anda bisa menetapkan arah layout dengan atribut `android:orientation`.



Perlu diperhatikan, bahwa dalam satu file xml, hanya ada satu root tagging, tidak diperbolehkan ada dua root tagging. Artinya semua elemen tagging anak mengakui atribut properties yang dimiliki oleh root tagging sebagai induknya.

Catatan : root tagging adalah tagging utama atau disebut tagging induk, dimana semua tagging lainnya berada di dalamnya. Contoh pada linear layout, sebuah tagging dimulai dengan `<LinearLayout>` dan diakhiri dengan `</LinearLayout>` dan semua tagging lainnya berada dalam block tagging tersebut.

```
<LinearLayout>  
<  
<  
>  
>  
<  
>  
</LinearLayout>
```

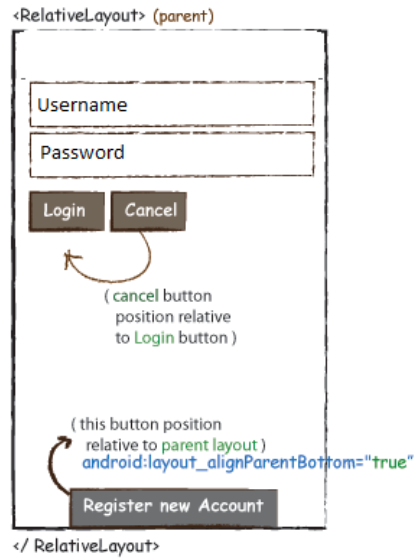
Block root

A diagram illustrating the structure of a LinearLayout XML block. On the left, the XML tags are listed: <LinearLayout>, <, <, >, >, <, >, and </LinearLayout>. A large right-facing curly bracket groups these tags together, with the text 'Block root' positioned to the right of the bracket.

2) RelativeLayout

RelativeLayout membuat posisi elemen didalamnya tergantung elemen lainnya. Tampilan RelativeLayout memungkinkan kita membuat elemen yang satu relative dengan posisi elemen yang lain tanpa aturan sesuai orientasi (horizontal atau vertical) seperti yang diterapkan pada Linear Layout. Ada 4 dasar relative layout yaitu :

- Di bawah (below)
- Di atas (above)
- Samping kiri (toLeftOf)
- Samping kanan (toRightOf)
- alignTop/alignBottom
- alignParentTop/alignParentBottom



3) FrameLayout

FrameLayout digunakan untuk menampilkan satu elemen UI pada satu waktu, dan elemen-elemen UI lainnya akan ditumpuk di atas elemen pertama jika ada lebih dari satu.

Contoh Implementasi: FrameLayout

Layout XML (activity_main.xml):

```

<FrameLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <!-- Background ImageView -->
  <ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/sample_image"
    android:scaleType="centerCrop" />

  <!-- Text Overlay di atas ImageView -->
  <TextView
    android:id="@+id/text_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
  
```

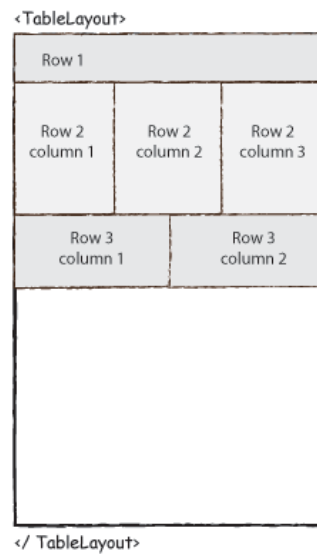
```

    android:text="This is a FrameLayout"
    android:textSize="18sp"
    android:textColor="#FFFFFF"
    android:layout_gravity="center" />
</FrameLayout>

```

4) Table Layout

Layout yang digunakan untuk membangun user interface (tampilan antar muka) aplikasi android dengan berdasarkan Baris dan Kolom.



5) AbsoluteLayout (Deprecated)

AbsoluteLayout memungkinkan Anda menentukan posisi elemen UI berdasarkan koordinat x dan y. Namun, layout ini telah usang (deprecated) dan sebaiknya tidak digunakan lagi karena tidak responsif.

Contoh Implementasi:

```

<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```
    android:text="Click Me"
    android:layout_x="50dp"
    android:layout_y="100dp"/>
</AbsoluteLayout>
```

Penjelasan:

layout_x dan layout_y: Menentukan posisi elemen UI pada layout menggunakan koordinat x dan y.

Kekurangan: Layout ini tidak responsif dan tidak mendukung berbagai ukuran layar.

6) ScrollView

ScrollView adalah container yang memungkinkan Anda menggulung (scroll) konten yang lebih besar dari layar.

Contoh Implementasi:

```
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

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

        <TextView
            android:text="This is a very long text. This is a very long text. This is a very
long text..."/>
        <!-- Komponen lainnya -->

    </LinearLayout>
</ScrollView>
```

Penjelasan:

- ✓ ScrollView memungkinkan pengguna menggulung konten yang melebihi batas layar.
- ✓ Di dalam ScrollView, hanya ada satu elemen parent, seperti LinearLayout di atas.

7) TableRow

TableRow adalah elemen dari TableLayout yang merepresentasikan satu baris dalam tabel.

Contoh Implementasi:

```
<TableLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="wrap_content">

  <TableRow>
    <TextView
      android:text="Nama"/>
    <EditText
      android:hint="Masukkan Nama"/>
  </TableRow>

  <TableRow>
    <Button
      android:text="Submit"/>
  </TableRow>
</TableLayout>
```

Penjelasan:

TableRow mengatur komponen dalam satu baris. Di atas, TextView dan EditText ada dalam satu baris, dan Button di baris lain.

8) Ticker (Marquee Text)

Ticker, atau teks marquee, adalah teks yang berjalan (bergerak) secara horizontal dalam tampilan.

Contoh Implementasi:

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="This is a scrolling text"
  android:ellipsize="marquee"
  android:marqueeRepeatLimit="marquee_forever"
  android:singleLine="true"
  android:focusable="true"
  android:focusableInTouchMode="true"/>
```

Penjelasan:

ellipsize="marquee": Membuat teks berjalan (scrolling) seperti efek marquee.

marqueeRepeatLimit="marquee_forever": Teks akan terus berulang tanpa henti.

9) Kombinasi Layout

Kombinasi layout digunakan untuk membuat tampilan yang lebih kompleks dengan menggabungkan beberapa jenis layout di dalam satu layout.

Contoh Implementasi:

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:text="Form Input"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

  <ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```

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

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TableRow>
            <TextView
                android:text="Nama:"/>
            <EditText
                android:hint="Masukkan Nama"/>
        </TableRow>

        <TableRow>
            <TextView
                android:text="Alamat:"/>
            <EditText
                android:hint="Masukkan Alamat"/>
        </TableRow>

    </TableLayout>

</LinearLayout>

</ScrollView>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit"/>
</LinearLayout>

```


Penjelasan:

Menggabungkan LinearLayout, ScrollView, dan TableLayout untuk membuat form input yang panjang dan bisa digulung.

Kombinasi layout membantu mengatur komponen UI yang lebih kompleks.

3.4 PRAKTIKUM

1) Buat Proyek Baru:

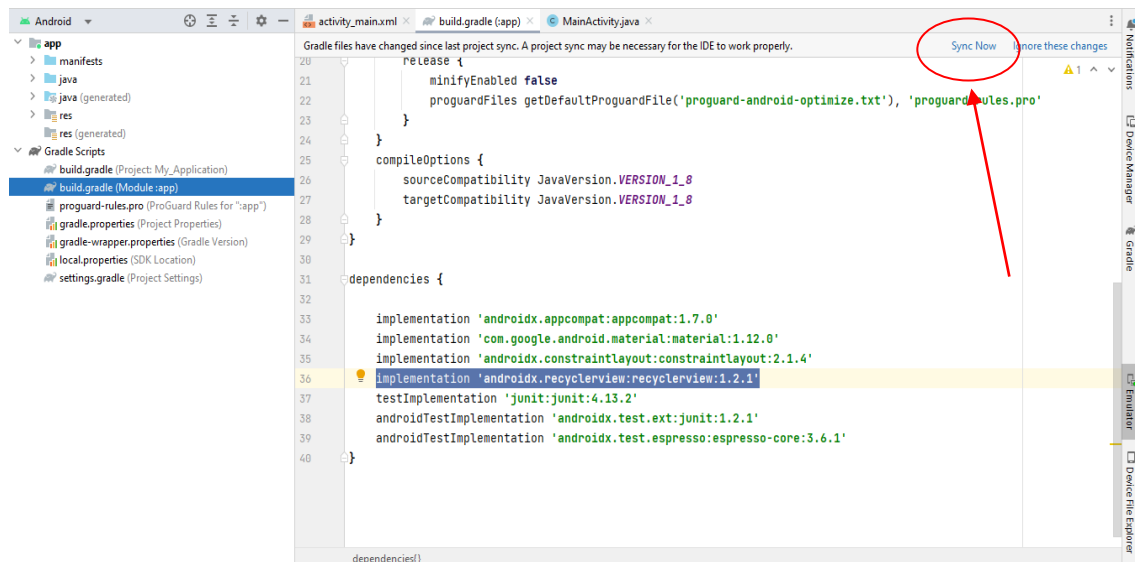
- Buka Android Studio, pilih "Start a new Android Studio project".
- Pilih template "Empty Activity" dan klik "Next".
- Masukkan nama proyek : KatalogBarang_NPMMasing-masing

2) Tambahkan Dependensi

- Buka file build.gradle (Module: app).
- Pastikan Anda memiliki dependensi untuk RecyclerView. Tambahkan baris berikut jika belum ada:

```
dependencies {  
    implementation 'androidx.recyclerview:recyclerview:1.2.1'  
}
```

Klik "Sync Now" untuk menyinkronkan dependensi.



3) Buat Model Data

Buat model data untuk produk. Buat file baru bernama Product.java di dalam folder java/com.example.katalogbarang.

```

package com.example.katalogbarang;

public class Product {
    private String name;
    private String description;
    private int imageResource;

    public Product(String name, String description, int imageResource) {
        this.name = name;
        this.description = description;
        this.imageResource = imageResource;
    }

    public String getName() {
        return name;
    }

    public String getDescription() {
        return description;
    }

    public int getImageResource() {
        return imageResource;
    }
}

```

4) Buat Layout Item

Buat layout untuk item produk. Buat file baru bernama product_item.xml di dalam folder res/layout.

```

<?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:orientation="vertical"

```

```
android:padding="16dp">

<ImageView
    android:id="@+id/product_image"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:scaleType="centerCrop" />

<TextView
    android:id="@+id/product_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:textStyle="bold"
    android:paddingTop="8dp" />

<TextView
    android:id="@+id/product_description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp" />
</LinearLayout>
```

5) Buat Adapter RecyclerView

Buat adapter untuk RecyclerView. Buat file baru bernama ProductAdapter.java di dalam folder java/com.example.katalogbarang.

```
package com.example.katalogbarang;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
```

```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class ProductAdapter extends
RecyclerView.Adapter<ProductAdapter.ProductViewHolder> {

    private List<Product> productList;
    private Context context;

    public ProductAdapter(List<Product> productList, Context context) {
        this.productList = productList;
        this.context = context;
    }

    @NonNull
    @Override
    public ProductViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.product_item, parent,
false);
        return new ProductViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ProductViewHolder holder, int position) {
        Product product = productList.get(position);
        holder.productName.setText(product.getName());
        holder.productDescription.setText(product.getDescription());
        holder.productImage.setImageResource(product.getImageResource());
    }

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

```

```

    }

    static class ProductViewHolder extends RecyclerView.ViewHolder {
        ImageView productImage;
        TextView productName;
        TextView productDescription;

        public ProductViewHolder(@NonNull View itemView) {
            super(itemView);
            productImage = itemView.findViewById(R.id.product_image);
            productName = itemView.findViewById(R.id.product_name);
            productDescription = itemView.findViewById(R.id.product_description);
        }
    }
}

```

6) Buat Layout Activity Utama

Edit layout untuk activity utama. Buka activity_main.xml di folder res/layout dan ubah isinya menjadi:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp" />
</RelativeLayout>

```

7) Implementasi di MainActivity

Edit file MainActivity.java untuk mengatur RecyclerView dan menampilkan data produk.

```
package com.example.katalogbarang;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private ProductAdapter productAdapter;
    private List<Product> productList;

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

        recyclerView = findViewById(R.id.recycler_view);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        // Inisialisasi daftar produk
        productList = new ArrayList<>();
        productList.add(new Product("Produk 1", "Deskripsi Produk 1",
R.drawable.product1));
        productList.add(new Product("Produk 2", "Deskripsi Produk 2",
R.drawable.product2));
        productList.add(new Product("Produk 3", "Deskripsi Produk 3",
R.drawable.product3));
        // Tambahkan produk lain sesuai kebutuhan
    }
}
```

```
productAdapter = new ProductAdapter(productList, this);
recyclerView.setAdapter(productAdapter);
}
}
```

8) Tambahkan Gambar Produk

- Siapkan gambar produk dalam format PNG atau JPG dan letakkan dalam folder res/drawable.
- Pastikan nama file gambar sesuai dengan yang digunakan dalam MainActivity.java (misal: product1.png, product2.png, dll.).

9) Jalankan Aplikasi

- Hubungkan perangkat Android Anda atau gunakan emulator.
- Klik tombol "Run" di Android Studio untuk menjalankan aplikasi.

10) Tambahkan Halaman Detail Produk (Opsional)

Jika Anda ingin menambahkan halaman detail produk ketika item diklik, Anda dapat membuat Activity baru untuk detail produk dan memanggilnya dari onBindViewHolder di ProductAdapter.

Contoh Pengaturan Halaman Detail Produk

1. Buat Activity Detail:

- Buat Activity baru bernama ProductDetailActivity.
- Tambahkan layout untuk detail produk (misal: activity_product_detail.xml).

2. Edit ProductDetailActivity.java:

```
package com.example.katalogbarang;

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

public class ProductDetailActivity extends AppCompatActivity {
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_product_detail);

    TextView productName = findViewById(R.id.product_name);
    TextView productDescription = findViewById(R.id.product_description);
    ImageView productImage = findViewById(R.id.product_image);

    Intent intent = getIntent();
    productName.setText(intent.getStringExtra("name"));
    productDescription.setText(intent.getStringExtra("description"));
    productImage.setImageResource(intent.getIntExtra("imageResource", 0));
}
}
```

Edit Layout activity_product_detail.xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/product_image"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:scaleType="centerCrop" />

    <TextView
        android:id="@+id/product_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:textStyle="bold">
```



```
        android:padding="16dp" />

<TextView
    android:id="@+id/product_description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:padding="16dp" />
</LinearLayout>
```

Panggil Halaman Detail dari ProductAdapter:

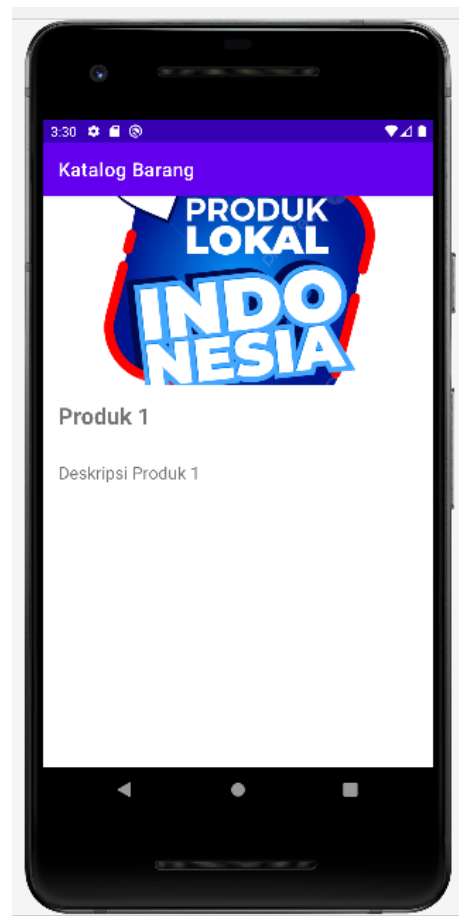
```
@Override
public void onBindViewHolder(@NonNull ProductViewHolder holder, int position) {
    Product product = productList.get(position);
    holder.productName.setText(product.getName());
    holder.productDescription.setText(product.getDescription());
    holder.productImage.setImageResource(product.getImageResource());

    holder.itemView.setOnClickListener(v -> {
        Intent intent = new Intent(context, ProductDetailActivity.class);
        intent.putExtra("name", product.getName());
        intent.putExtra("description", product.getDescription());
        intent.putExtra("imageResource", product.getImageResource());
        context.startActivity(intent);
    });
}
```

11) Output dari project tersebut adalah :



Tampilan awal saat project dijalankan



Tampilan saat salah satu produk di klik

Catatan : Silahkan edit deskripsi produk sesuai dengan jenis produk yang anda tampilkan.

3.5 TUGAS

1. Demokan project tersebut hingga muncul di emulator!
2. Tugas dikumpul dalam format pdf dengan nama file : NPM_Tugaske?
3. Didalam file tersebut dituliskan :
 - NPM
 - Nama Lengkap
 - Program Studi
 - Kelas