

MODUL PEMBELAJARAN

APLIKASI KOMPUTASI BERGERAK



DOSEN PENGAMPU :
Sopian Alviana, S.Kom., M.Kom

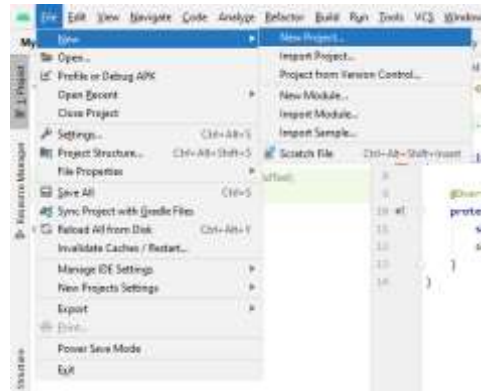
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
2021

1. Layout Antar Muka Bagian I

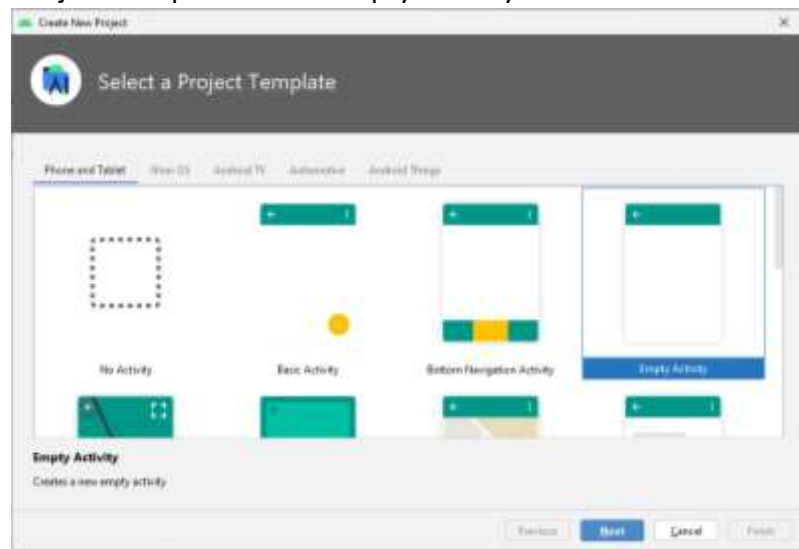
Layout merupakan tampilan dari aplikasi android yang memungkinkan untuk menata komponen – komponen pada aplikasi seperti Foto, Teks, Video maupun komponen lainnya. Fungsi layout mirip dengan kertas, yaitu sebagai media dalam menggambar. Pada layout kita bisa menambahkan komponen aplikasi bahkan, mengatur tampilan menjadi lebih indah dan nyaman di mata pengguna aplikasi.

Tahapan Membuat Project Baru

❖ File > New > New Project



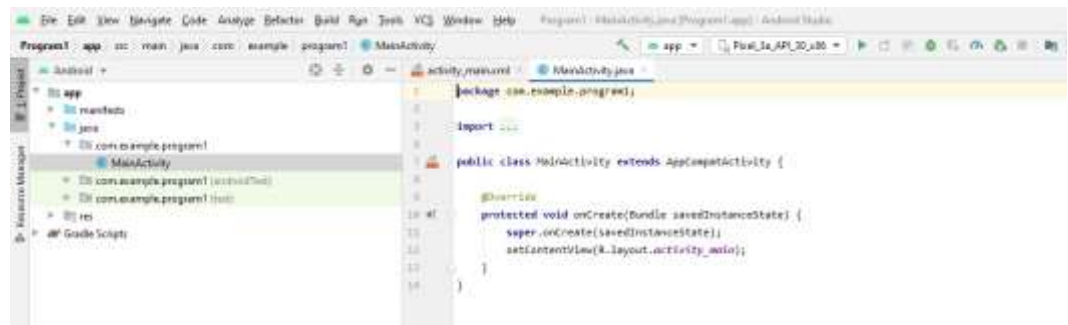
❖ Select a Project Template > Pilih Empty Activity > Next



❖ Configure Your Project

- Name : Isi Nama Project Aplikasi
- Package Name : Mengikuti Nama Project
- Save Location : Lokasi Project yang dibuat
- Language : Bahasa Pemrograman yang digunakan (Java / Kotlin)
- Minimum SDK : Minimum SDK Android yang digunakan untuk aplikasi

Jika sudah diisi > Klik Finish

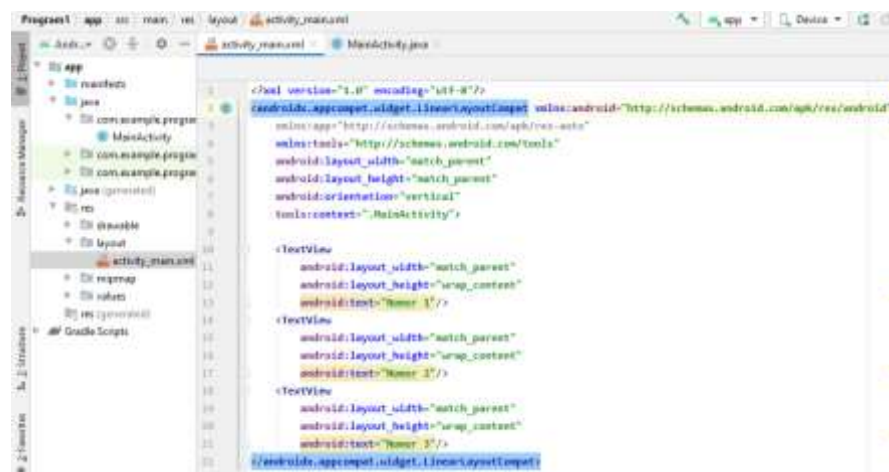


➤ Linear Layout

Merupakan layout yang berfungsi untuk menampilkan komponen – komponen aplikasi secara vertical atau horizontal.

Linear Layout pada android studio diawali dengan Tag pembuka <LinearLayout> dan diakhiri dengan tag </LinearLayout> pada file Activity.XML

Kode XML :



Hasil Program :



Silahkan coba dengan mengganti orientasi dari vertical menjadi horizontal. Dan modifikasi Teks dari program yang dibuat.

➤ Relative Layout

Merupakan layout yang berfungsi untuk menampilkan komponen – komponen aplikasi secara bebas tidak hanya vertical atau horizontal.

Relative Layout pada android studio diawali dengan Tag pembuka <RelativeLayout> dan diakhiri dengan tag </RelativeLayout> pada file Activity.XML

Kode XML :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:text="Nomor 1"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="Nomor 2"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:text="Nomor 3"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:text="Nomor 4"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Center"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"
        android:text="Nomor 5"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:text="Nomor 6"/>
    <TextView
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Nomor 7"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:text="Nomor 8"/>
</RelativeLayout>

```

Hasil Program :



➤ Frame Layout

Merupakan layout yang berfungsi untuk menampilkan komponen – komponen aplikasi secara tumpang tindih.

Frame Layout pada android studio diawali dengan Tag pembuka <FrameLayout> dan diakhiri dengan tag </FrameLayout> pada file Activity.XML

Kode XML :

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_gravity="center"
        android:background="@color/design_default_color_primary"/>

    <Button
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:text="Contoh Button"/>

```

Hasil Program :



➤ **TextView**

Komponen untuk menampilkan teks pada layar. TextView berisi property dari ukuran, style, dan warna. Berikut property untuk TextView

Kode XML :

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Contoh"
    android:textSize="12dp"
    android:textColor="@color/design_default_color_primary"
    android:textStyle="bold"
    android:id="@+id/TextView1"/>
```

TextView"

Hasil Program :



➤ **EditText**

EditText merupakan kustomisasi dari TextView yang dapat dikonfigurasi atau disunting sendiri.

Kode XML :

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Contoh TextView"
    android:textSize="12dp"
    android:textColor="@color/design_default_color_primary"
    android:textStyle="bold"
    android:id="@+id/TextView1"/>
```

Hasil Program :



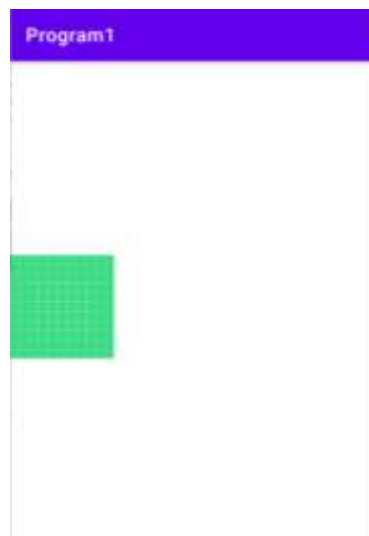
➤ **ImageView**

ImageView merupakan komponen yang digunakan untuk menampilkan gambar atau Ikon.

Kode XML :

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:src="@drawable/ic_launcher_background"/>
```

Hasil Program :



➤ **Button**

Button merupakan komponen yang digunakan untuk Tombol.

Kode XML :

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Contoh Button"/>
```

Hasil Program :



➤ Event Handling

Event Listener merupakan antar muka pada kelas View yang berisi metode pemanggilan Kembali. Metode ini akan dipanggil android Ketika view yang terdaftar dengan pendengar dipicu oleh interaksi pengguna.

Method	Description
onClick()	This method is called when the user touches or focuses on the item using navigation-keys or trackball or presses on the "enter" key or presses down on the trackball.
onLongClick()	This method is called when the user touches and holds the item or focuses on the item using navigation-keys or trackball and presses and holds "enter" key or presses and holds down on the trackball (for one second).
onFocusChange()	This method is called when the user navigates onto or away from the item, using the navigation-keys or trackball.
onKey()	This method is called when the user is focused on the item and presses or releases a hardware key on the device.
onTouch()	This method is called when the user performs a touch event, including a press, a release, or any movement gesture on the screen.
onCreateContextMenu()	This method is called when a Context Menu is being built.

Contoh Kode XML Event On Click :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
    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="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Contoh Event On Click"
        android:textSize="15dp"
```



```

        android:textAlignment="center"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tampilkan Data"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/hasilButton"/>
</androidx.appcompat.widget.LinearLayoutCompat>

```

Hasil Program :



Agar Button Tampilkan Data dapat menampilkan hasil data maka lakukan modifikasi pada MainActivity.Java, sebagai berikut :

```

public class MainActivity extends AppCompatActivity {
    Button button1; //Deklarasi Button
    TextView textView; //Deklarasi TextView

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button1 = (Button) findViewById(R.id.buton) //Memanggil Button
        textView = (TextView) findViewById(R.id.hasilButton) // Menampilkan Hasil

        //Event On Click
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                textView.setText("Anda Melakukan Klik Tombol");
            }
        });
    }
}

```

Hasil Program :



➤ Gaya dan Tema Aplikasi

Gaya dan tema di Android memungkinkan Anda memisahkan detail desain aplikasi dari struktur dan perilaku UI, mirip dengan stylesheet di desain web. Gaya adalah kumpulan atribut yang menentukan tampilan untuk view Tunggal. Gaya dapat menentukan atribut seperti warna font, ukuran font, warna latar belakang, dan banyak lagi.

Tema adalah jenis gaya yang diterapkan ke seluruh aplikasi, aktivitas, atau hierarki tampilan — bukan hanya tampilan individual. Saat Anda menerapkan gaya sebagai tema, setiap tampilan di aplikasi atau aktivitas akan menerapkan setiap atribut gaya yang didukungnya. Tema juga dapat menerapkan gaya ke elemen non-tampilan, seperti status bar dan latar belakang jendela. Gaya dan tema dideklarasikan di `res/values/` bernama `Styles.XML`

❖ Gaya dan Tema Terang (Light)

Mengatur gaya dan tema dengan mode terang dengan melakukan modifikasi pada bagian `Res > Values > Styles` untuk Android Studio versi lama. Sedangkan untuk versi android terbaru dapat dimodifikasi pada `Res > Themes > themes.xml` (untuk terang) dan `themes.xml (night)` untuk mode gelap. Untuk versi android studio terbaru sudah mendukung dua versi tema tidak perlu dibuat ulang.

❖ Gaya dan Tema Gelap (Dark)

Mengatur gaya dan tema dengan mode gelap dengan melakukan modifikasi pada bagian `Res > Values > Styles` untuk Android Studio versi lama. Sedangkan untuk versi android terbaru dapat dimodifikasi pada `Res > Themes > themes.xml` (untuk terang) dan `themes.xml (night)` untuk mode gelap. Untuk versi android studio terbaru sudah mendukung dua versi tema tidak perlu dibuat ulang.

Latihan :

Dari beberapa tutorial diatas dan contoh program. Silahkan anda berlatih dengan memodifikasi atau membuat program baru dengan beberapa jenis Layout, berbeda komponen dan tema. Serta mencoba membuat aplikasi dengan kombinasi berbagai komponen dan layout yang telah dijelaskan diatas. Good Luck selamat belajar.

2. Layout Antar Muka Bagian II

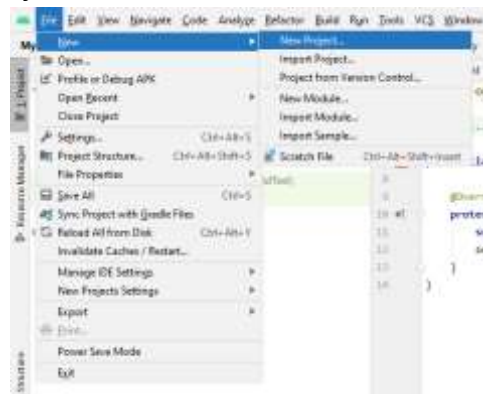
➤ CardView

Merupakan tampilan antar muka untuk menampilkan dalam bentuk kartu. Atau tampilan dengan memiliki drop shadow (elevation) dan corner radius. CardView merupakan subclass dari Frame Layout. Jika kita melihat daftar aplikasi pada Play Store, tampilan yang digunakan adalah merupakan tampilan dalam bentuk CardView.

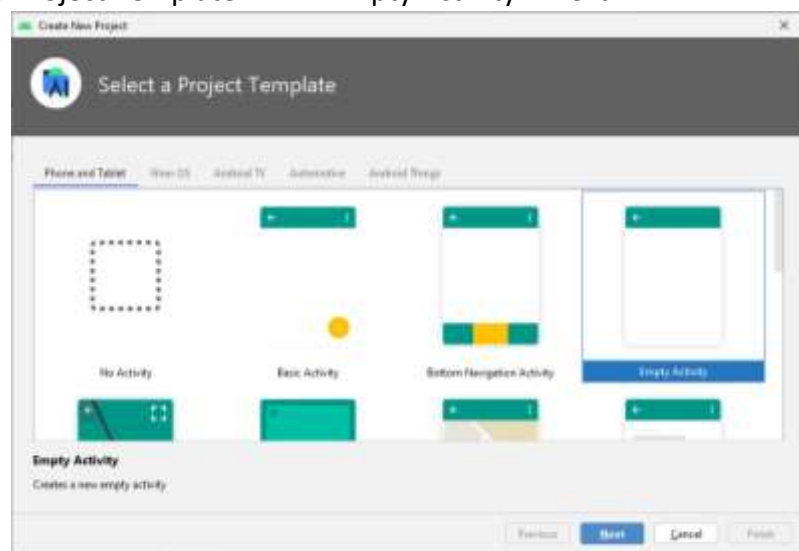
Karena tampilan CardView ini merupakan tampilan dengan tambahan fitur, maka perlu ditambahkan library pada bagian Dependensi pada proyek yang sedang dibuat.

Penambahan library tersebut ada pada bagian **build.gradle (Module:app)**. Tahapan Membuat Project Baru

❖ File > New > New Project



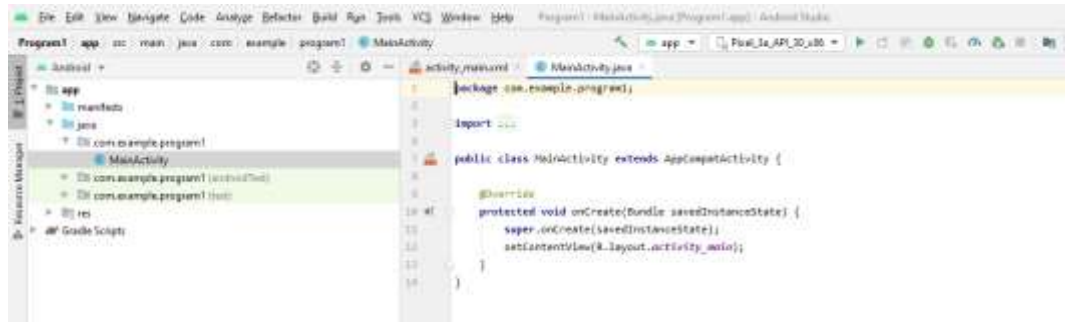
❖ Select a Project Template > Pilih Empty Activity > Next



❖ Configure Your Project

- Name : Isi Nama Project Aplikasi
- Package Name : Mengikuti Nama Project
- Save Location : Lokasi Project yang dibuat
- Language : Bahasa Pemrograman yang digunakan (Java / Kotlin)
- Minimum SDK : Minimum SDK Android yang digunakan untuk aplikasi

Jika sudah diisi > Klik Finish



Tambahkan terlebih dahulu File gambar yang akan kita gunakan ke Android Studio. Yaitu dengan copy gambar kemudian klik kanan pada bagian **Drawable Android Studio** dan paste file tersebut. File gambar ini akan digunakan pada Imageview dengan memanggil pada bagian **android:src**.

Buka halaman **activity_main.xml**, karena kita akan menggunakan tampilan (View dengan format CardView) maka kita rubah layout dari Constrain Layout menjadi Relative Layout dan tambahkan widget CardView dengan kode :

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardElevation="5dp"
    app:cardCornerRadius="5dp"
    android:layout_margin="5dp"
    app:cardBackgroundColor="@color/white"
    app:cardMaxElevation="10dp"
    app:cardPreventCornerOverlap="true"
    app:cardUseCompatPadding="true">
```

```
</androidx.cardview.widget.CardView>
```

tambahkan 1 buah ImageView dan TextView seperti berikut, didalam kode CardView :

```
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="center"
    android:src="@drawable/shinymewtwo"
    android:layout_margin="10dp"
    android:id="@+id/img"
    android:contentDescription="@string/app_name" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Shiny Mewtwo"
    android:layout_gravity="bottom|center_horizontal"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="20dp"
    android:textSize="20sp"
    android:textStyle="bold"
/>
```

Sehingga keseluruhan XML main activity menjadi :

```

<?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"
    tools:context=".MainActivity">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardElevation="5dp"
        app:cardCornerRadius="5dp"
        android:layout_margin="5dp"
        app:cardBackgroundColor="@color/white"
        app:cardMaxElevation="10dp"
        app:cardPreventCornerOverlap="true"
        app:cardUseCompatPadding="true">
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:src="@drawable/shinymewtwo"
            android:layout_margin="10dp"
            android:id="@+id/img"
            android:contentDescription="@string/app_name" />

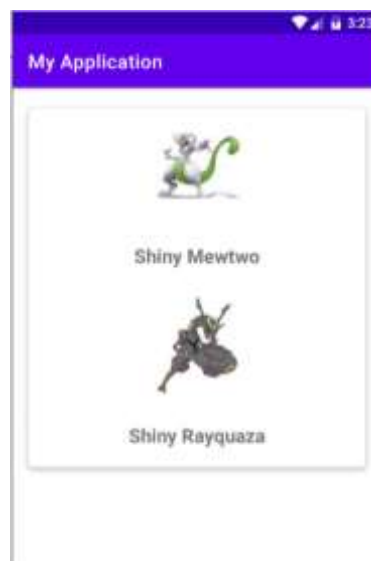
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Shiny Mewtwo"
            android:layout_gravity="bottom|center_horizontal"
            android:layout_marginTop="20dp"
            android:layout_marginBottom="20dp"
            android:textSize="20sp"
            android:textStyle="bold"
            />
    </androidx.cardview.widget.CardView>
</LinearLayout>

```

Berikut hasil tampilan dengan CardView :



Contoh dengan penambahan Gambar dan TextView :



➤ ListView

Merupakan tampilan antar muka untuk menampilkan dalam bentuk list (daftar). ListView merupakan salah satu widget yang menampilkan data atau nilai dalam bentuk daftar. Data atau nilai yang ditampilkan dapat dari sebuah array atau database yang sudah ditentukan. Untuk tahapan kali ini kita akan menampilkan data dari masukan berupa array. Seperti biasa, silahkan buat project baru atau modifikasi project sebelumnya.

Kemudian pada tampilan Main.XML gunakan layout LinearLayout karena tampilannya akan berupa list (daftar). Kemudian tambahkan widget ListView dengan kode :

```
<ListView  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:id="@+id/ListView"/>
```

Android:id="@+id/ListView"/> merupakan penamaan atau id untuk Komponen / Widget ListView yang kita gunakan.

Setelah Menyusun untuk tampilan ListView, kita mulai memasukan data melalui array di Activity Java (MainActivity.Java). Deklarasikan ListView, Array Adapter, dan Array untuk datanya seperti berikut :

```
ListView list; // Deklarasi ListView
ArrayAdapter adapter; //Inisialisasi Array Adapter
String[] bahasaPemrograman = {"HTML", "C", "C++", "PHP", "Java"}; //Data Array yang
dumsukan
```

Kode untuk menampilkan data pada activity :

```
list = (ListView)findViewById(R.id.ListView); //Memasukan data kedalam List yang
telah dideklarasikan
adapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1,
bahasaPemrograman); //Mengisi Array dengan Data
list.setAdapter(adapter); //memanggil data array dan menampilkan pada List
```

Hasil Keseluruhan kode pada Main Activity.Java :

```
package com.example.belajarview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {
    ListView list; // Deklarasi ListView
    ArrayAdapter adapter; //Inisialisasi Array Adapter
    String[] bahasaPemrograman = {"HTML", "C", "C++", "PHP", "Java"}; //Data Array
    yang dumsukan

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

        list = (ListView)findViewById(R.id.ListView); //Memasukan data kedalam
        List yang telah dideklarasikan
        adapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1,
bahasaPemrograman); //Mengisi Array dengan Data
        list.setAdapter(adapter); //memanggil data array dan menampilkan pada List
    }
}
```

Hasil Tampilan :



➤ GridView

Merupakan tampilan antar muka untuk menampilkan dalam bentuk Grid. GridView merupakan salah satu widget yang menampilkan data atau nilai dalam bentuk Grid atau kotak – kotak seperti bentuk rak lemari. Data atau nilai yang ditampilkan dapat dari sebuah array atau database yang sudah ditentukan. Untuk tahapan kali ini kita akan menampilkan data dari masukan berupa array. Seperti biasa, silahkan buat project baru atau modifikasi project sebelumnya. Tampilan atau menampilkan data dalam bentuk Grid sama dengan ListView hanya berbeda dalam bentukan tampilan data saja. Kita masih akan menggunakan data atau program yang dibuat dengan menampilkan data pada ListView. Kita akan modifikasi tampilan data tersebut kedalam bentuk GridView.

Merubah XML dari ListView menjadi GridView :

```
<GridView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numColumns="4"
    android:id="@+id/GridView"/>
```

Atribut ListView dan GridView sama hanya ada penambahan jumlah kolom pada Grid dengan ***android:numColumns = "Nilai atau Jumlah Kolom Yang diinginkan"*** contoh ini dengan jumlah kolom 4.

Perubahan pada Main Activity Java :

```
GridView grid;
ArrayAdapter adapter;
String[] bahasaPemrograman = {"HTML", "C", "C++", "PHP", "Java", "Python", "R"};
```

Kode untuk menampilkan Grid :

```
grid = (GridView)findViewById(R.id.GridView);
adapter = new ArrayAdapter<this, android.R.layout.simple_list_item_1,
    bahasaPemrograman>;
grid.setAdapter(adapter);
```

Hasil Keseluruhan pada Main Activity Java :

```
package com.example.belajarview;
```



```

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;

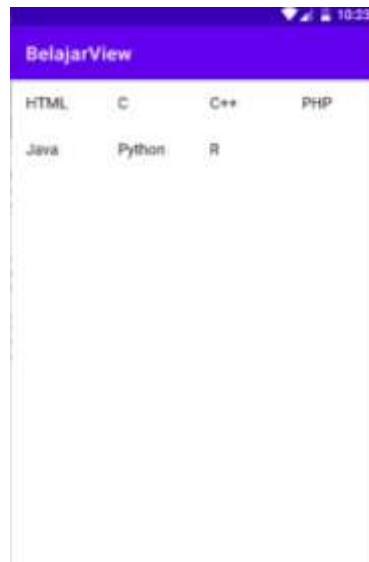
public class MainActivity extends AppCompatActivity {
    GridView grid;
    ArrayAdapter adapter;
    String[] bahasaPemrograman = {"HTML", "C", "C++", "PHP", "Java", "Python", "R"};

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

        grid = (GridView) findViewById(R.id.gridView);
        adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
            bahasaPemrograman);
        grid.setAdapter(adapter);
    }
}

```

Hasil Tampilan :



Tampilan jika kita merubah jumlah Kolom menjadi 2 :



➤ RecyclerView

Merupakan tampilan antar muka untuk menampilkan dalam bentuk list atau Grid. RecyclerView merupakan salah satu widget yang menampilkan data atau nilai dalam bentuk mirip dengan ListView akan tetapi lebih fleksibel. Data atau nilai yang ditampilkan dapat dari sebuah array atau database yang sudah ditentukan. RecyclerView berfungsi sebagai *container* untuk menampilkan sejumlah data besar yang dapat dilakukan *scroll* secara efisien.

Dalam membuat tampilan dengan RecyclerView ada hal yang perlu diperhatikan yaitu :

1. Layout
2. Class dan Adapter

Membuat layout untuk tampilan data yaitu dengan xml (Club_List_Row) untuk menampung data dalam baris:

```
<?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="wrap_content"
    android:orientation="vertical">
    <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:textSize="16dp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/genre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/title" />

    <TextView
        android:id="@+id/year"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"/>

```

```
</RelativeLayout>
```

Membuat Class (Club.Java)

```

package com.example.exampleryclerview;

public class Club {
    private String title, genre, year;

    public Club() {
    }

    public Club(String title, String genre, String year) {
        this.title = title;
        this.genre = genre;
        this.year = year;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String name) {
        this.title = name;
    }

    public String getYear() {
        return year;
    }

    public void setYear(String year) {
        this.year = year;
    }

    public String getGenre() {
        return genre;
    }

    public void setGenre(String genre) {
        this.genre = genre;
    }
}

```

Membuat Adapter :

```

package com.example.exampleryclerview;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class ClubAdapter extends RecyclerView.Adapter<ClubAdapter.MyViewHolder> {
    private List<Club> clubsList;
}

```

```

public class MyViewHolder extends RecyclerView.ViewHolder {
    public TextView title, year, genre;

    public MyViewHolder(View view) {
        super(view);
        title = (TextView) view.findViewById(R.id.title);
        genre = (TextView) view.findViewById(R.id.genre);
        year = (TextView) view.findViewById(R.id.year);
    }
}

public ClubAdapter(List<Club> clubsList) {
    this.clubsList = clubsList;
}

@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.club_list_row, parent, false);

    return new MyViewHolder(itemView);
}

@Override
public void onBindViewHolder(MyViewHolder holder, int position) {
    Club club = clubsList.get(position);
    holder.title.setText(club.getTitle());
    holder.genre.setText(club.getGenre());
    holder.year.setText(club.getYear());
}

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

```

Tampilan Main Activity XML :

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/recycler_view"
        android:scrollbars="vertical"/>

</RelativeLayout>

```

Kode Untuk Main Activity Java :

```

package com.example.examplerecyclerview;

import androidx.appcompat.app.AppCompatActivity;

```

```
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private List<Club> clubList = new ArrayList<>();
    private RecyclerView recyclerView;
    private ClubAdapter mAdapter;

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

        recyclerView = (RecyclerView) findViewById(R.id.recycler_view);

        mAdapter = new ClubAdapter(clubList);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.setAdapter(mAdapter);

        prepareClubData();
    }
    private void prepareClubData() {
        Club club = new Club("Manchester City", "Liga Premier", "Inggris");
        clubList.add(club);
        club = new Club("Inter Milan", "Serie A", "Italia");
        clubList.add(club);
        club = new Club("Bayern Munchen", "Bundesliga", "Jerman");
        clubList.add(club);

        mAdapter.notifyDataSetChanged();
    }
}
```

Tampilan Program :



➤ Dialog

Membuat fungsi dialog dalam android studio yaitu fungsi yang menampilkan sebuah dialog atau Message Box dalam aplikasi Desktop:

Kode 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">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:id="@+id/button1"
        android:text="TES DIALOG"/>

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

Kode Main Activity Java

```
package com.example.contohdialog;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button TesDialog;
```

```

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

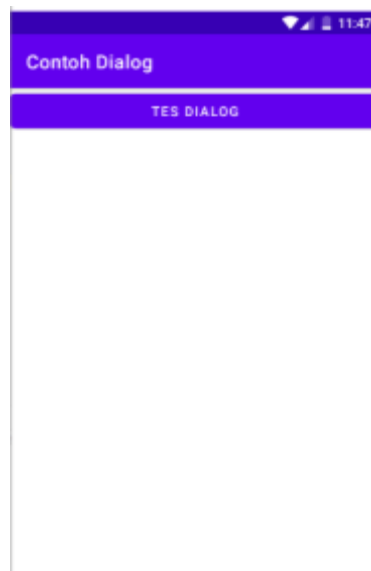
    TesDialog = (Button) findViewById(R.id.button1);
    TesDialog.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            TampilkanDialog();
        }
    });
}

private void TampilkanDialog(){
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
    alertDialogBuilder.setTitle("Keluar Aplikasi ?");

    alertDialogBuilder
        .setMessage("Klik Ya Untuk Menutup")
        .setIcon(R.mipmap.ic_launcher)
        .setCancelable(false)
        .setPositiveButton("Ya", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                MainActivity.this();
            }
        })
        .setNegativeButton("Tidak", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });
    AlertDialog alertDialog = alertDialogBuilder.create();
    alertDialog.show();
}
}

```

Tampilan Aplikasi :



Tampilan Setelah di Klik Tombol :

