

MODUL PEMBELAJARAN

APLIKASI KOMPUTASI BERGERAK



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

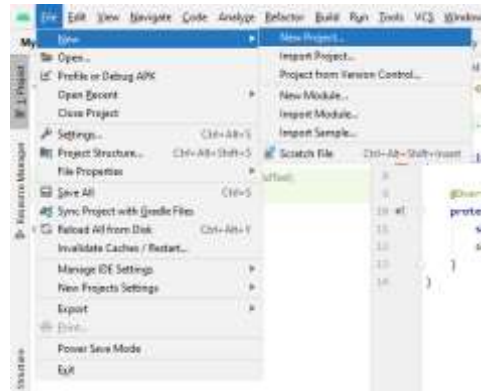
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
2021

1. Layout Antar Muka Bagian I

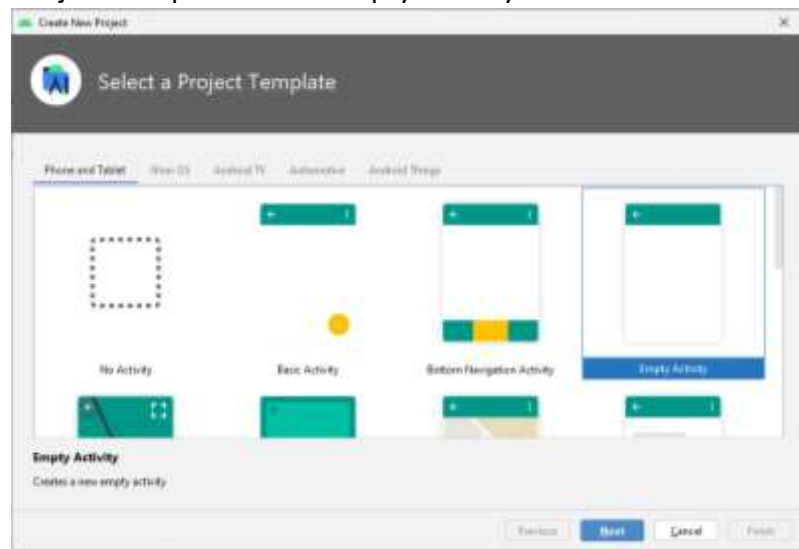
Layout merupakan tampilan dari aplikasi android yang memungkinkan untuk menata komponen – komponen pada aplikasi seperti Foto, Teks, Video maupun komponen lainnya. Fungsi layout mirip dengan kertas, yaitu sebagai media dalam menggambar. Pada layout kita bisa menambahkan komponen aplikasi bahkan, mengatur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

Tahapan Membuat Project Baru

❖ File > New > New Project



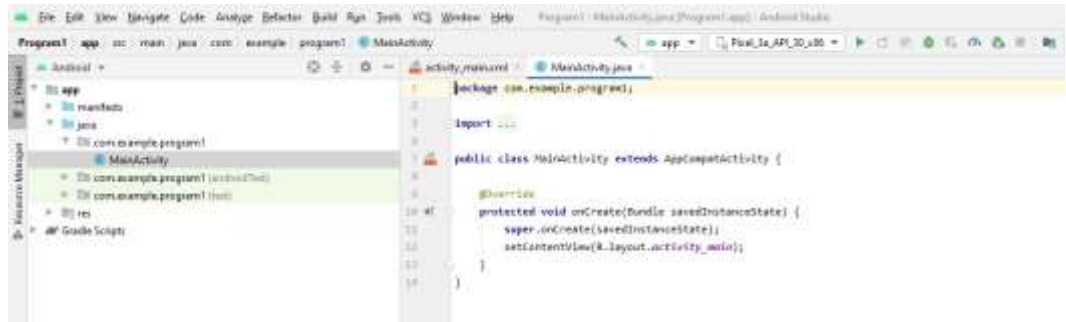
❖ Select a Project Template > Pilih Empty Activity > Next



❖ 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

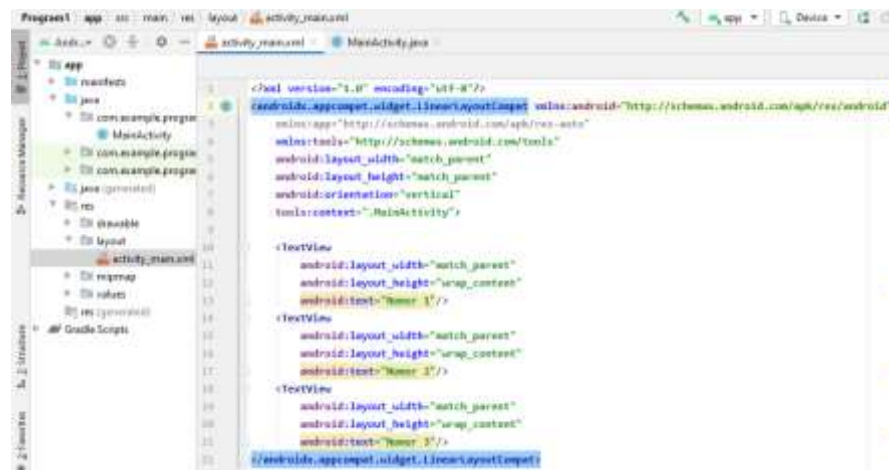


❖ Linear Layout

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 :



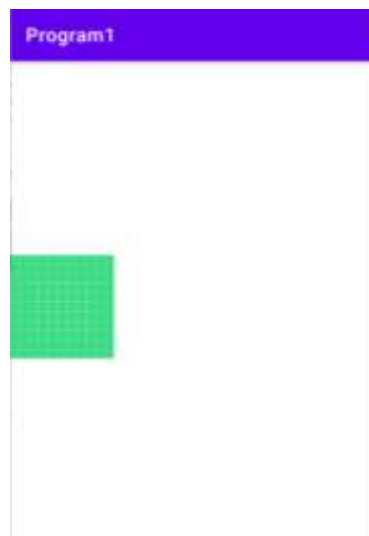
❖ ImageView

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

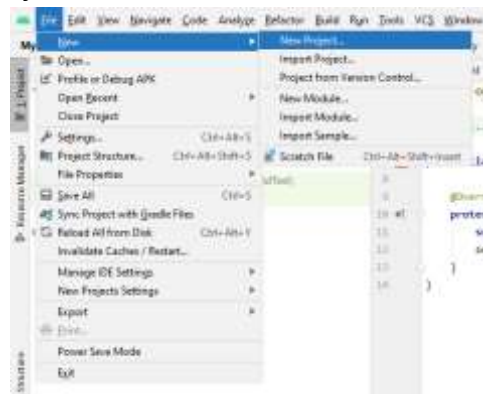
❖ CardView

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.

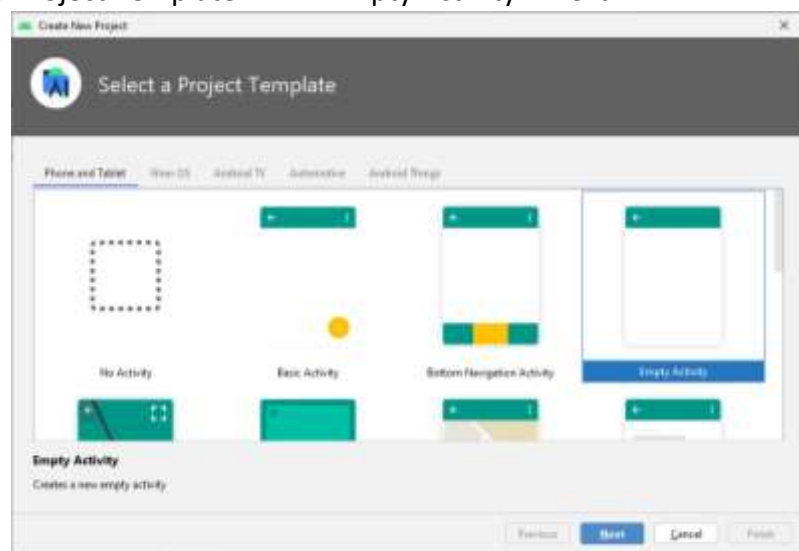
Karena tampilan CardView ini merupakan tampilan dengan tambahan fitur, maka perlu ditambahkan library pada bagian Dependensi pada proyek yang sedang dibuat.

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

❖ File > New > New Project



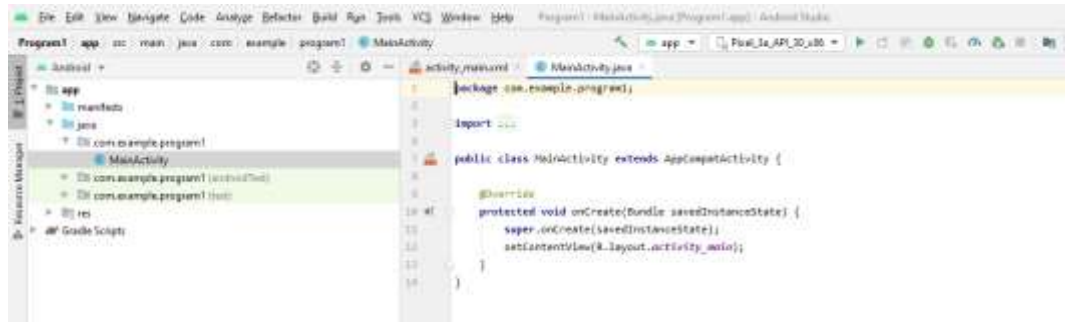
❖ Select a Project Template > Pilih Empty Activity > Next



❖ 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" />

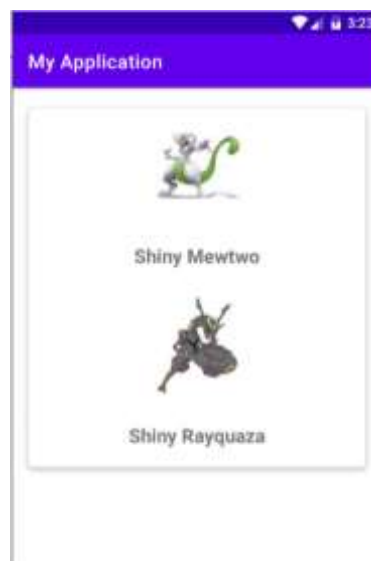
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Shiny Mewtwo"
            android:layout_gravity="bottom|center_horizontal"
            android:layout_marginTop="20dp"
            android:layout_marginBottom="20dp"
            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;

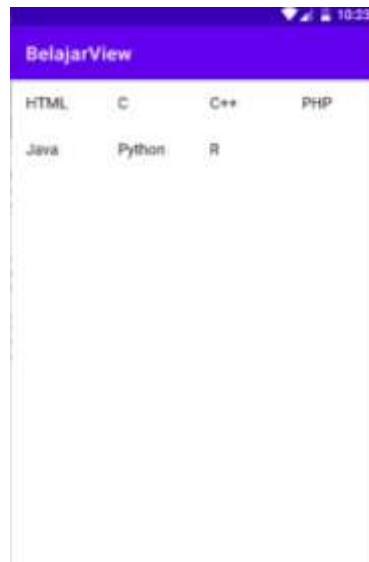
public class MainActivity extends AppCompatActivity {
    GridView grid;
    ArrayAdapter adapter;
    String[] bahasaPemrograman = {"HTML", "C", "C++", "PHP", "Java", "Python", "R"};

    @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.exampleryclerview;

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);

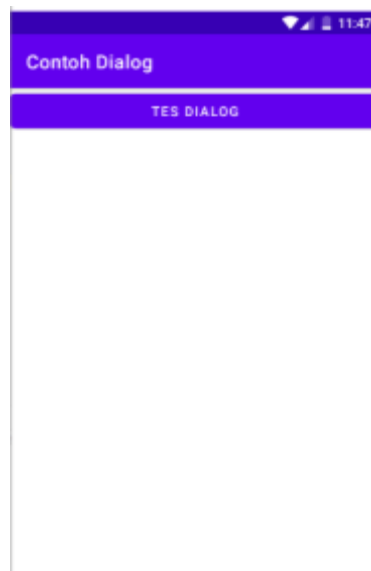
    TesDialog = (Button) findViewById(R.id.button1);
    TesDialog.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            TampilkanDialog();
        }
    });
}

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

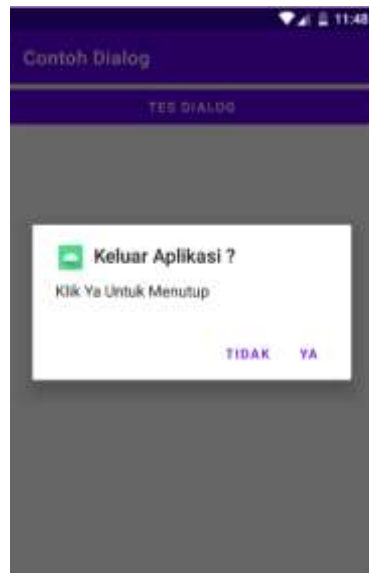
    alertDialogBuilder
        .setMessage("Klik Ya Untuk Menutup")
        .setIcon(R.mipmap.ic_launcher)
        .setCancelable(false)
        .setPositiveButton("Ya", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                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 :



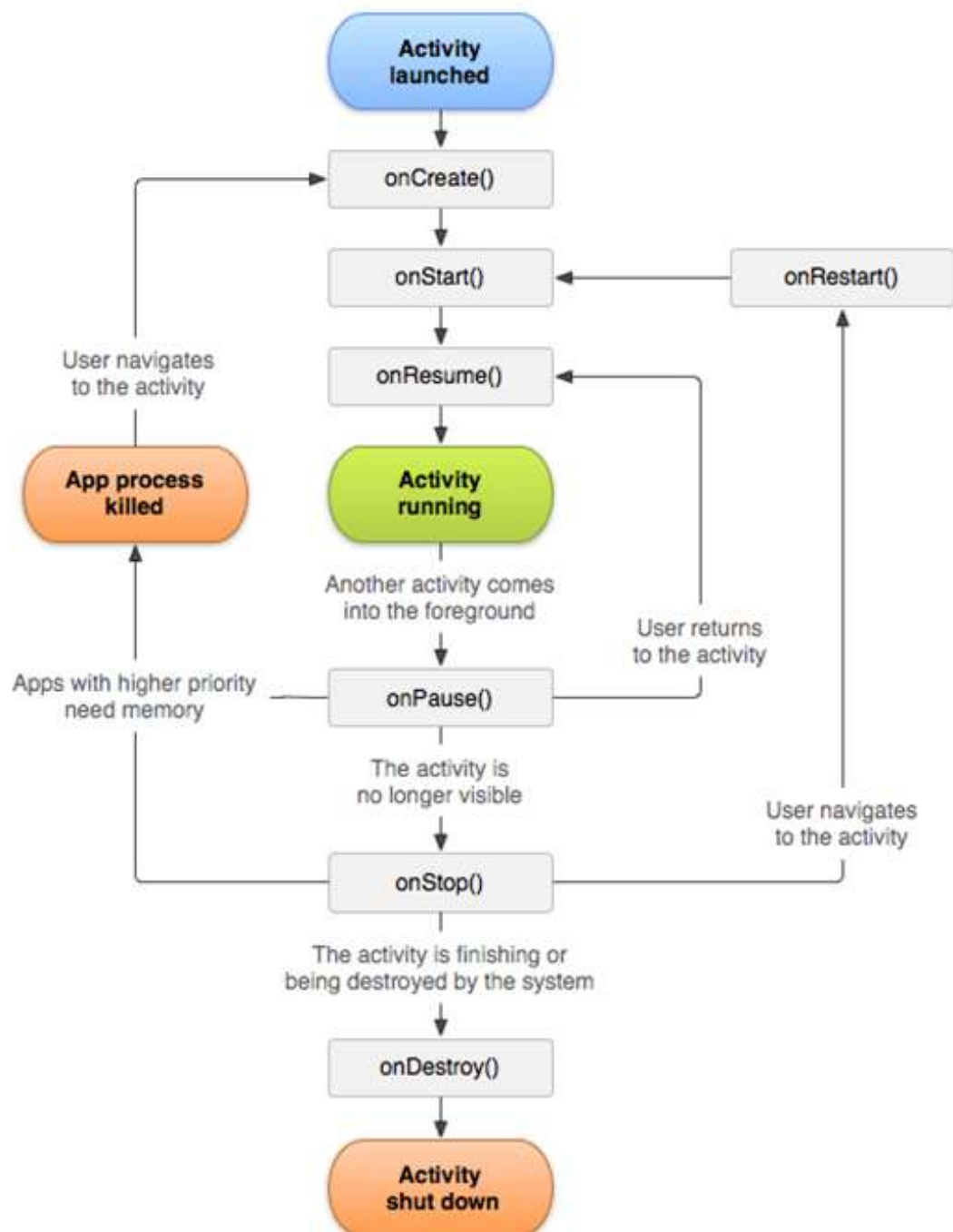
3. Activity Life Cycle Pada Android

❖ Pengenalan Activity

Activity merupakan komponen yang dapat dilihat oleh pengguna, sehingga dapat berinteraksi dengan aplikasi. Activity pada android sama halnya dengan sebuah halaman pada aplikasi website dan sebuah form pada aplikasi berbasis desktop. Activity dapat diciptakan dan dimsunahkan. Activity mempunyai siklus hidup yaitu kondisi yang akan dialami baik saat diciptakan atau dimsunahkan.

❖ Activity Life Cycle

Ketika aplikasi berjalan, maka aplikasi mempunyai siklus hidup. Berikut siklus hidup activity pada android.



Penjelasan setiap activity :

- a. onCreate(), merupakan kondisi saat activity baru diciptakan, biasanya dilakukan inisialisasi pada tahap ini.
- b. onStart(), merupakan activity saat dimulai
- c. onResume(), merupakan activity saat dibuka kembali, biasanya dieksekusi setelah pause.
- d. onPause(), merupakan activity yang akan dipanggil saat ada activity lain yang terbuka.
- e. onStop(), merupakan kondisi saat activity tidak ditampilkan dilayar (saat tombol home ditekan).
- f. onRestart(), merupakan kondisi saat activity kembali dibuka oleh pengguna.
- g. onDestroy(), merupakan kondisi saat activity dimusnahkan pada memory.

❖ Pengenalan Intent

Intent merupakan sebuah class yang ada pada pemrograman android yang berfungsi untuk perpindahan activity. Intent juga merupakan suatu objek pada suatu activity yang dapat berkomunikasi dengan activity lain. Activity yang dipanggil dapat internal activity atau berbeda activity. Untuk internal activity seperti dalam satu package atau berbeda package dalam satu project yang sama.

Jenis Intent terbagi menjadi dua jenis :

- a. Intent Explicit, Intent yang berfungsi untuk mengaktifkan komponen – komponen dalam satu aplikasi yang sama. Contoh berpindah dari satu activity ke activity lain.
- b. Intent Implicit, Intent yang berfungsi memanggil fungsi activity yang sudah ada pada fungsi internal. Contoh memanggil Dial Number, Open Browser, dan lain – lain.

❖ Contoh Program Untuk Activity Life Cycle

Buatlah sebuah project aplikasi baru, atau memodifikasi aplikasi lama juga dapat digunakan. Setelah project baru terbuat (Pembuatan Project Baru, tahapan ada di materi pertama).

Pada saat membuat project aplikasi, siklus activity onCreate sebenarnya telah dibuat sehingga tidak perlu dibuat lagi.

Untuk melihat proses activity life cycle tersebut kita dapat menggunakan Pesan Logcat dan Toast agar terlihat setiap aktifitasnya.

Contoh dengan menggunakan LogCat :

Pada Main Activity menambahkan String status yaitu :

```
String status = "Android : ";
```

Menambahkan log di event onCreate Main Activity yaitu :

```
Log.d(status, "Ini Event onCreate()");
```

Kode keseluruhan untuk Main Activity :

```
package com.example.myapplication;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```

import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
    String status = "Android : ";

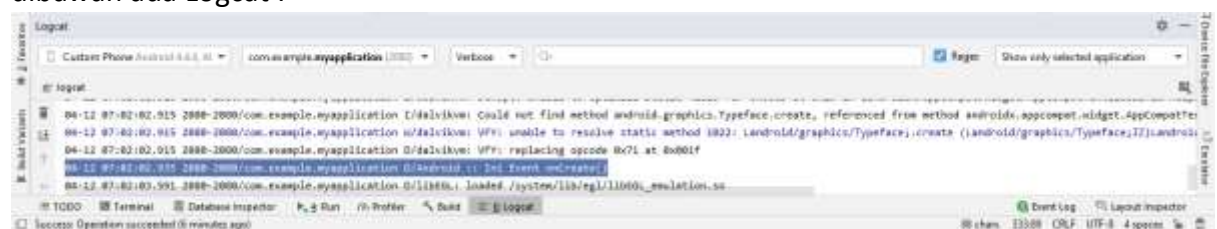
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(status, "Ini Event onCreate()");
    }
}

```

Hasil Program :



Hasil penelusuran saat aplikasi selesai di jalankan, cek pada Android Studio dibawah ada Logcat :



Event onCreate dengan tanda yang telah dibuat sudah terlihat.

Silahkan coba tambahkan untuk event onStart, onResume, onPause, onStop, dan onDestroy. Dengan kode berikut ini :

```

@Override
protected void onStart(){
    super.onStart();
    Log.d(status, "Ini Event onStart()");
}

@Override
protected void onResume(){
    super.onResume();
    Log.d(status, "Ini Event onResume()");
}

@Override
protected void onPause(){

```

```

        super.onPause();
        Log.d(status, "Ini Event onPause()");
    }
    @Override
    protected void onStop(){
        super.onStop();
        Log.d(status, "Ini Event onStop()");
    }
    @Override
    protected void onDestroy(){
        super.onDestroy();
        Log.d(status, "Ini Event onDestroy()");
    }

```

Kemudian, coba jalankan dan cek pada Logcat. Lalu coba dengan setiap event yang telah dibuat.

Karena menggunakan Logcat tidak terlalu terlihat, maka kita ganti **Log.d** dengan menggunakan Toast. Silahkan ganti log.d dengan toast berikut :

```
Toast.makeText(this, "Siklus onCreate", Toast.LENGTH_SHORT).show();
```

Kode untuk event yang lain :

```

@Override
protected void onStart(){
    super.onStart();
    Toast.makeText(this, "Siklus onStart", Toast.LENGTH_SHORT).show();
}
@Override
protected void onResume(){
    super.onResume();
    Toast.makeText(this, "Siklus onResume", Toast.LENGTH_SHORT).show();
}
@Override
protected void onPause(){
    super.onPause();
    Toast.makeText(this, "Siklus onPause", Toast.LENGTH_SHORT).show();
}
@Override
protected void onStop(){
    super.onStop();
    Toast.makeText(this, "Siklus onStop", Toast.LENGTH_SHORT).show();
}
@Override
protected void onDestroy(){
    super.onDestroy();
    Toast.makeText(this, "Siklus onDestroy", Toast.LENGTH_SHORT).show();
}

```

Sehingga kode keseluruhan untuk Main Activity :

```

package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    String status = "Android : ";

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toast.makeText(this, "Siklus onCreate", Toast.LENGTH_SHORT).show();
}
@Override
protected void onStart(){
    super.onStart();
    Toast.makeText(this, "Siklus onStart", Toast.LENGTH_SHORT).show();
}
@Override
protected void onResume(){
    super.onResume();
    Toast.makeText(this, "Siklus onResume", Toast.LENGTH_SHORT).show();
}
@Override
protected void onPause(){
    super.onPause();
    Toast.makeText(this, "Siklus onPause", Toast.LENGTH_SHORT).show();
}
@Override
protected void onStop(){
    super.onStop();
    Toast.makeText(this, "Siklus onStop", Toast.LENGTH_SHORT).show();
}
@Override
protected void onDestroy(){
    super.onDestroy();
    Toast.makeText(this, "Siklus onDestroy", Toast.LENGTH_SHORT).show();
}
}

```

Hasil Aplikasi :



Terlihat hasil dengan muncul notifikasi activity. Pada awal biasanya activity dari onCreate > onStart > onResume.

Tampilan Saat ditekan tombol Home :

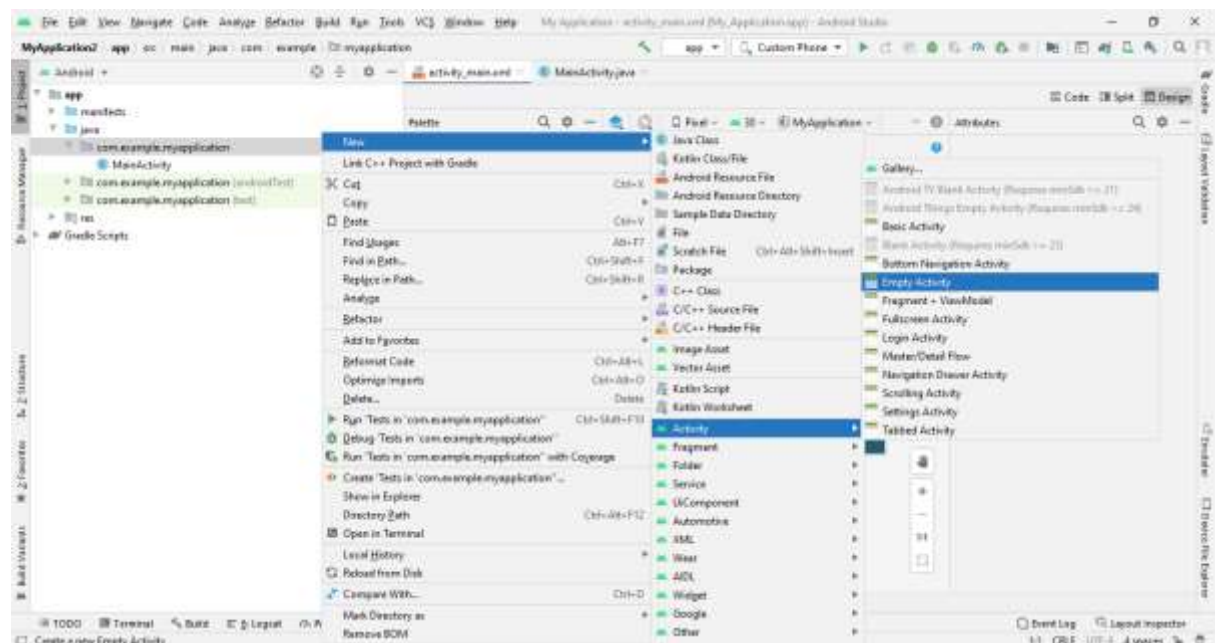


Maka siklusnya berubah.

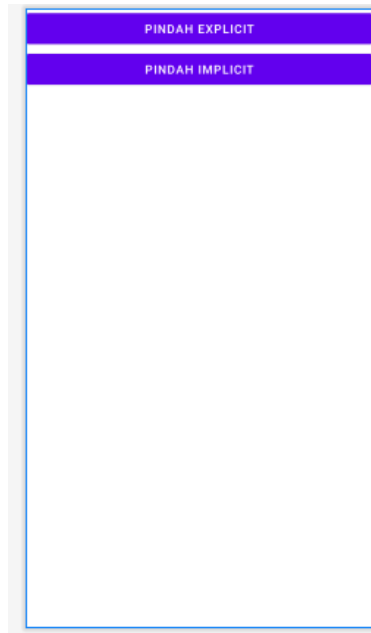
❖ Contoh Program Untuk Intent

Intent yang akan kita gunakan dalam program ini adalah dengan intent Implicit dan Explicit.

Buat Project Baru > Kemudian Tambahkan dengan Activity Baru. Jadi terdapat dua Activity yaitu (Main Activity dan Activity kedua). Berikut Menambahkan Activity baru :



Desain Main Activity (XML) seperti berikut ini :



Berikut 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">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/intentExplicit"
        android:text="Pindah Explicit" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/intentImplicit"
        android:layout_below="@+id/intentExplicit"
        android:text="Pindah Implicit" />

</RelativeLayout>
```

Desain XML Activity kedua :



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=".MainActivity2">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="Selamat Anda Telah Berpindah Activity"
        android:gravity="center"
        android:textSize="40dp"/>

</RelativeLayout>
```

Deklarasi Intent mempunyai format yaitu :

Intent (NAMA INTENT) = new Intent (Activity Pertama, Activity Tujuan);

Memanggil atau memulai Intent dengan format :

startActivity (NAMA INTENT)

Kita akan menambahkan perpindahan intent pada setiap tombol Button yang telah disediakan pada desain Main Activity (XML).

Buka Main Activity Java dengan menambahkan :

Button **implicitIntent**, **explicitIntent**;

Deklarasi untuk setiap Button yang digunakan :

```
explicitIntent = findViewById(R.id.intentExplicit);
```

```
implicitIntent = findViewById(R.id.intentImplicit);
```

Menambahkan Event Onclick pada Button Pertama (PINDAH EXPLICIT) :

```
explicitIntent.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, MainActivity2.class);
        startActivity(intent);
    }
});
```

Menambahkan Event Onclick pada Button Kedua (PINDAH IMPLICIT) :

```
implicitIntent.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String url = "https://unikom.ac.id";
        Intent bukaWeb = new Intent(Intent.ACTION_VIEW);
        bukaWeb.setData(Uri.parse(url));
        startActivity(bukaWeb);
    }
});
```

Pada Intent Implicit kita coba menggunakan aktifitas membuka sebuah website yaitu dengan membuka website UNIKOM.

Keseluruhan kode untuk Main Activity :


```

package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button implicitIntent, explicitIntent;

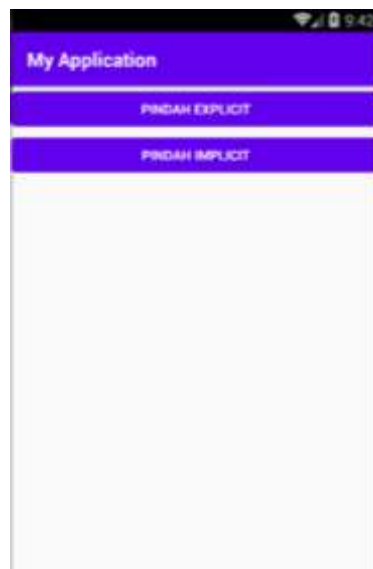
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        explicitIntent = findViewById(R.id.intentExplicit);
        implicitIntent = findViewById(R.id.intentImplicit);
        explicitIntent.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
                MainActivity2.class);
                startActivity(intent);
            }
        });

        implicitIntent.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String url = "https://unikom.ac.id";
                Intent bukaWeb = new Intent(Intent.ACTION_VIEW);
                bukaWeb.setData(Uri.parse(url));
                startActivity(bukaWeb);
            }
        });
    }
}

```

Hasil Tampilan saat dijalankan :



Tampilan saat Button pertama (PINDAH EXPLICIT) ditekan, menghasilkan tampilan :



Tampilan berpindah ke activity yang kedua.

Kemudian, berikut hasil tampilan Button kedua (PINDAH IMPLICIT) saat tombol ditekan, akan merujuk ke browser untuk membuka secara langsung website yang telah dideskripsikan sebelumnya di Main Activity.



Silahkan lakukan praktikum mandiri dengan mencoba materi Activity Life Cycle dan Intent. Kombinasikan dengan materi dari awal yaitu dengan desain form yang lebih baik dan gunakan intent perpindahan Activity dan Intent pemanggilan fungsi lain seperti Dial Number, Open Browser dan fungsi lainnya.

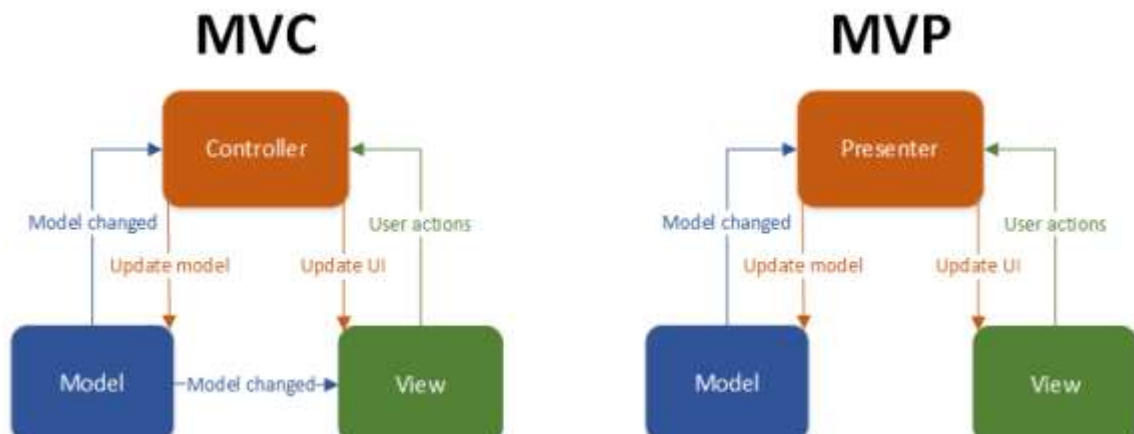
TUGAS 1 :

Buatlah sebuah aplikasi bebas, dalam aplikasi yang anda buat harus meliputi :

- Harus ada layout antarmuka aplikasi. (Layout bebas anda gunakan dapat Linear, Constraint, Relative layout, dll). Dapat menggunakan minimal dua buah layout antarmuka.
- Implementasikan activity life cycle pada setiap activity yang dibuat. Activity minimal menggunakan 3 buah activity.
- Tugas dibuat dalam bentuk laporan dan video pengerjaan program.
- Laporan isinya adalah Kode dan Hasil Screenshoot program yang anda buat dan link video anda. Sedangkan, video berisi proses anda sedang mengerjakan tugas anda. Video diunggah saja di youtube atau Google Drive.
- Waktu pengerjaan adalah 1 Minggu

4. Konsep MVP

MVP (Model View Presenter) merupakan Teknik coding untuk memisahkan antara logika aplikasi dengan tampilan / view dari sebuah aplikasi. Teknik MVP dengan pengkodean biasa yang membedakan adalah pada penempatan dari coding dan tampilan aplikasi. Berikut perbedaan antara Teknik MVC (Model View Control) dan MVP (Model View Presenter) :



Dari gambar diatas yang membedakan adalah pada MVP bagian View tidak mengakses model. Sedangkan pada MVC View dapat mengakses model.

Komponen pada pattern atau model Teknik MVP yaitu Model, View, dan Presenter.

Contoh kita akan membuat sebuah project dengan contoh kasus untuk Login.

Keuntungan dengan menggunakan MVP membuat kode menjadi lebih bersih dan rapi, sehingga memudahkan saat proses maintenance aplikasi. Selain itu, meringankan proses view karena telah dipisahkan antara proses dengan view nya.

Bagian pada MVP (Model, View, Presenter) :

- Model, merupakan bagian yang mengurus set data yang akan digunakan oleh presenter.
- View, merupakan bagian yang mengolah hal yang ditampilkan pada User Interface seperti ShowMessage, SetText, ShowDialog, dan lain – lain.
- Presenter, merupakan bagian yang menghubungkan antara Model dengan View. Setiap action yang diminta oleh user akan diproses oleh presenter.

Kita akan mencoba membuat sebuah aplikasi dengan pattern MVC dan MVP, yaitu sebuah aplikasi login. Pada pembuatan aplikasi dengan MVC dan MVP ini, kita akan membuat urutan pembuatan tampilan login dengan konsep MVC dan MVP.

- Aplikasi dengan menggunakan MVC Pattern

Buat Tampilan Layout :



Kode untuk 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">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="228dp"
        android:id="@+id/editUsername"
        android:hint="Username"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editUsername"
        android:id="@+id/editPassword"
        android:hint="Password"
        android:inputType="textPassword"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editPassword"
        android:id="@+id/btnLogin"
        android:text="LOGIN"/>

</RelativeLayout>
```

Kode Main Activity Java :

```
package com.example.mvc;
```

```

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private EditText username, password;
    private Button tmbLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username = (EditText) findViewById(R.id.editUsername);
        password = (EditText) findViewById(R.id.editPassword);
        tmbLogin = (Button) findViewById(R.id.btnLogin);

        tmbLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String nama = username.getText().toString();
                String pass = password.getText().toString();
                if (nama.isEmpty() || pass.isEmpty()){
                    showErrorLogin();
                } else {
                    if (nama.equals("admin") && pass.equals("admin")){
                        successLogin();
                    } else {
                        errorLogin();
                    }
                }
            }
        });
    }

    public void showErrorLogin(){
        Toast.makeText(this, "Error Username dan Password tidak Valid !",
        Toast.LENGTH_SHORT).show();
    }

    public void successLogin(){
        Toast.makeText(this, "Anda Berhasil Login !", Toast.LENGTH_SHORT).show();
    }

    public void errorLogin(){
        Toast.makeText(this, "Username atau Password Salah !",
        Toast.LENGTH_SHORT).show();
    }
}

```

Hasil Running Aplikasi :



A screenshot of a mobile application interface titled 'MVC'. It features a login form with two input fields: 'Username' and 'Password'. Below the fields is a blue button labeled 'LOGIN'. The status bar at the top shows the time as 11:06.

Saat tombol login ditekan, tapi username dan password tidak diisi :



A screenshot of the MVC login screen after the 'LOGIN' button was pressed with empty fields. A grey error message box at the bottom reads 'Error Username dan Password tidak Valid !'. The status bar shows the time as 11:07.

Saat tombol login ditekan, tetapi username atau password yang dimasukan salah :

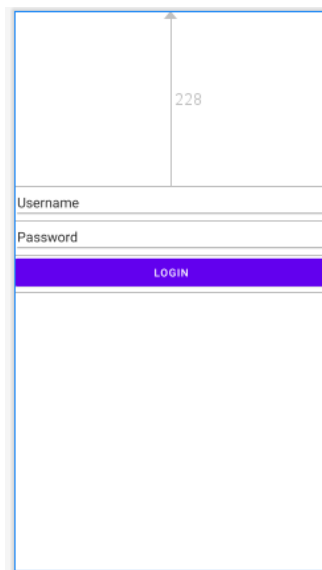


A screenshot of the MVC login screen after the 'LOGIN' button was pressed with the username 'admin' and a masked password. A grey error message box at the bottom reads 'Username atau Password Salah !'. The status bar shows the time as 11:08.

Saat tombol login ditekan, username dan password benar :



- Aplikasi dengan menggunakan MVP Pattern
 - A. Buat Tampilan Layout



Kode untuk 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">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="228dp"
        android:id="@+id/editUsername"
        android:text="Username"/>

    <EditText
        android:layout_width="match_parent"
```

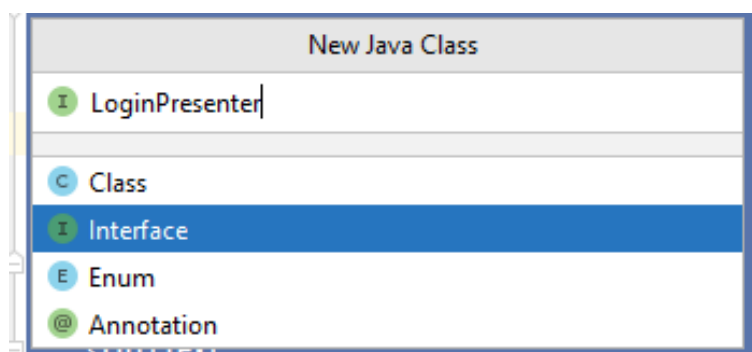
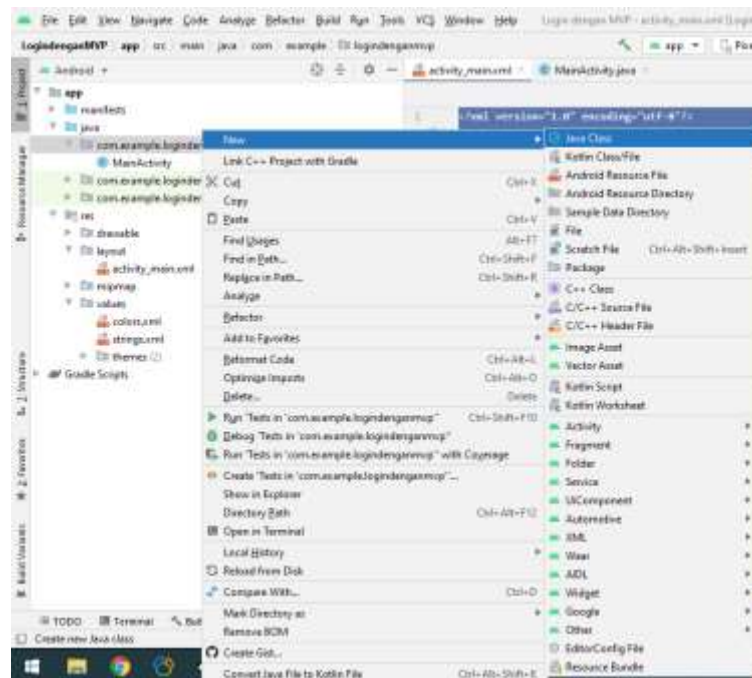
```

        android:layout_height="wrap_content"
        android:layout_below="@id/editUsername"
        android:id="@+id/editPassword"
        android:text="Password"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editPassword"
        android:id="@+id/btnLogin"
        android:text="LOGIN"/>

</RelativeLayout>

```

- B. Buat Interface dengan Klik Kanan pada folder Java yang berisi MainActivity. Klik Kanan > New > Pilih Java Class > Isikan Nama Class dan Pilih Interface.



Fungsi pada class Interface Presenter ini adalah fungsi Data Username dan Password, maka isikan dengan kode XML :

```

package com.example.logindenganmvp;

public interface LoginPresenter {
    void login(String username, String password);
}

```


- C. Tambahkan Interface untuk View, sama dengan Langkah membuat Presenter Login. Kali ini membuat View dengan setiap view Validasi Login, Login Berhasil, dan Error Login. Dengan kode lengkap :

```
package com.example.logindenganmvp;
```

```
public interface LoginView {
    void validasiError();
    void loginBerhasil();
    void loginGagal();
}
```

- D. Membuat implementasi Interface, dimana kita akan membuat class Model yang mengimplementasi LoginPresenter. Tambahkan Classnya dengan pemilihan seperti Interface tetapi pilihannya menjadi Class.

```
package com.example.logindenganmvp;
```

```
import android.text.TextUtils;
```

```
public class LoginImplementasi implements LoginPresenter {
    private LoginView loginView;
    public LoginImplementasi(LoginView loginView){
        this.loginView = loginView;
    }
    @Override
    public void login(String username, String password){
        if (TextUtils.isEmpty(username) ||
            TextUtils.isEmpty(password)){
            loginView.validasiError();
        } else {
            if (username.equals("admin") &&
                password.equals("admin")){
                loginView.loginBerhasil();
            } else {
                loginView.loginGagal();
            }
        }
    }
}
```

- E. Kita edit untuk Main Activity Java dengan mengimplemtasi LoginView. Deklarasikan setiap komponen yang dibutuhkan dimulai dengan Presenter, EditText untuk username dan password, dan Button untuk Login.

Kode Lengkap :

```
package com.example.logindenganmvp;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity implements LoginView{
    private LoginPresenter presenter;
    private EditText nama, password;
    private Button btLogin;
```

```

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

    //View Standar
    nama = (EditText) findViewById(R.id.editUsername);
    password = (EditText) findViewById(R.id.editPassword);
    btLogin = (Button) findViewById(R.id.btnLogin);
    presenter = new LoginImplementasi(this);

    btLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            presenter.login(nama.getText().toString(),
password.getText().toString());
        }
    });
}
// Fungsi - Fungsi yang dipanggil yang ada pada LoginView
@Override
public void validasiError(){
    Toast.makeText(this, "Error Username dan Password tidak Valid !",
Toast.LENGTH_SHORT).show();
}
@Override
public void loginBerhasil(){
    Toast.makeText(this, "Anda Berhasil Login !", Toast.LENGTH_SHORT).show();
}
@Override
public void loginGagal(){
    Toast.makeText(this, "Username atau Password Salah !",
Toast.LENGTH_SHORT).show();
}
}

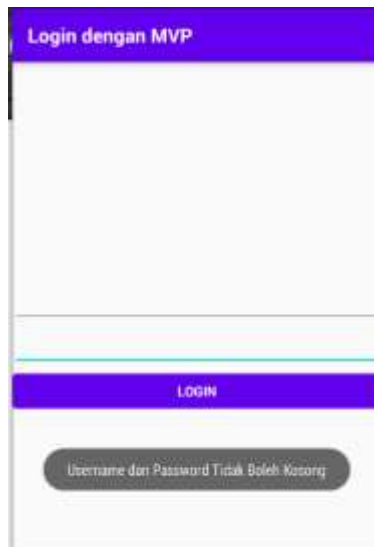
```

Hasil Aplikasi :



The screenshot shows a mobile application interface with a purple header bar containing the text "Login dengan MVP". Below the header, there are two white input fields with light blue borders, labeled "Username" and "Password". At the bottom of the screen, there is a blue button with the text "LOGIN" in white capital letters.

Saat Tombol Login ditekan tanpa mengisi username dan password :



Login dengan MVP

Username

Password

LOGIN

Username dan Password Tidak Boleh Kosong

Saat tombol Login ditekan, tetapi username atau password yang dimasukan salah :



Login dengan MVP

Username

Password

LOGIN

Username atau Password Salah !

Saat tombol login ditekan, user dan password sesuai :



Login dengan MVP

Username

Password

LOGIN

Anda Berhasil Login !

5. Penyimpanan Data Pada Android

Untuk media penyimpanan pada android menggunakan system file yang serupa dengan system file berbasis disk pada platform lain. Sistem ini menyediakan berbagai opsi untuk menyimpan data aplikasi.

- ❖ Penyimpanan khusus aplikasi, menyimpan file yang hanya ditujukan untuk penggunaan aplikasi, baik di direktori khusus dalam penyimpanan internal maupun direktori khusus dalam penyimpanan eksternal. Menggunakan direktori dalam penyimpanan internal untuk menyimpan informasi sensitif yang tidak boleh diakses aplikasi lain.
- ❖ Penyimpanan Bersama, menyimpan file yang ingin dibagikan aplikasi dengan aplikasi lain, termasuk media, dokumen, dan file lainnya.
- ❖ Preferensi, menyimpan data primitif pribadi dalam key-value pair.
- ❖ Database, menyimpan data terstruktur dalam database pribadi menggunakan library persistensi room.

Database Pada Android

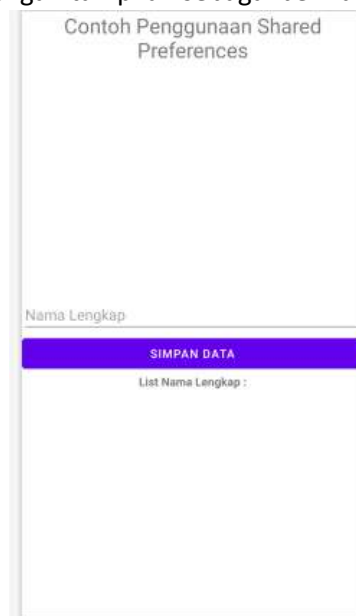
Merupakan kumpulan data yang disimpan secara sistematis di dalam computer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Media penyimpanan pada data pada android mempunyai banyak opsi. Berikut beberapa database pada android yang dapat digunakan :

1. Shared Preferences
2. SQLite
3. Realm

- Shared Preferences, merupakan opsi penyimpanan beberapa data Bersama dengan sebuah aplikasi. Karakter penyimpanan shared preferences bersifat key value storage. Sehingga hanya bisa menyimpan data bertipe primitif (Float, Integer, Long, String, Boolean). Penyimpanan data ini bersifat minimalis dan cocok untuk penyimpanan data yang sedikit.

Dalam menyimpan data shared preferences menggunakan objek ***sharedpreferences.string "Setting"*** nama file yang digunakan oleh SharedPreferences, MODE_PRIVATE untuk akses file. Berikut contoh program dengan menggunakan penyimpanan data Shared Preferences.

Membuat project aplikasi dengan tampilan sebagai berikut :



Berikut kode lengkap XML nya :

```
<?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:id="@+id/txt1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Contoh Penggunaan Shared Preferences"
        android:textAlignment="center"
        android:textSize="25dp" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:hint="Nama Lengkap"
        android:id="@+id/inputNama"
        android:inputType="textPersonName"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/inputNama"
        android:text="SIMPAN DATA"
        android:id="@+id/btnSimpan"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btnSimpan"
        android:gravity="center"
        android:text="List Nama Lengkap :"
        android:textStyle="bold"/>

</RelativeLayout>
```

Kemudian kita modifikasi Main Activity.Java

```
package com.example.menggunakansharedpreferences;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    //Deklarasi Variabel
    private TextView Output;
```

```

private EditText Input;
private Button simpanData;
//Deklarasi Nama Penyimpanan SharedPreferences atau Nama Database
private String MyPREFERENCES = "MyPrefs";
// Deklarasi Nama Field atau Nama Data Field
private String nama = "nameKey";
//Deklarasi Shared Preferences
SharedPreferences sharedPreferences;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Input = findViewById(R.id.inputNama);
    Output = findViewById(R.id.outputNama);
    simpanData = findViewById(R.id.btnSimpan);
    sharedPreferences = getSharedPreferences(MyPREFERENCES,
Context.MODE_PRIVATE);

    simpanData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String dataInput = Input.getText().toString();
            SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.putString(nama,dataInput);
            editor.commit();
            Toast.makeText(getApplicationContext(), "Data Berhasil Disimpan",
Toast.LENGTH_SHORT).show();
            Output.setText("Nama Lengkap :
"+sharedPreferences.getString(nama,null));
        }
    });
}
}

```

Tampilan Aplikasi Saat di Run :



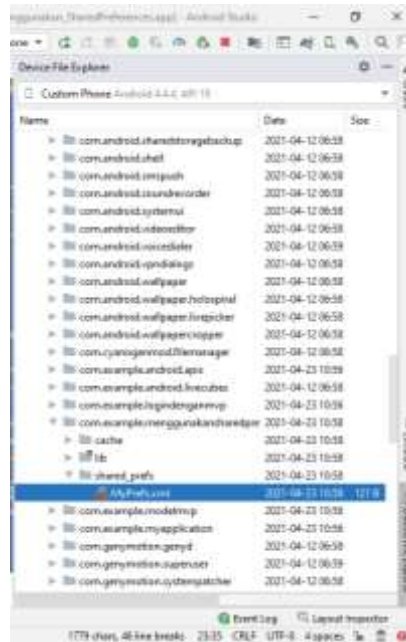
Proses Pengisian Data :



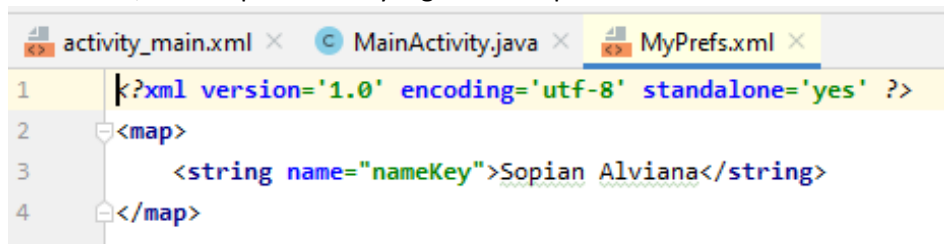
Hasil setelah data disimpan :



Pengecekan Data Yang telah disimpan pada SharedPreferences di Android Emulator. Buka di Pojok Kanan Bawah ada **DEVICE FILE EXPLORER > FOLDER data > FOLDER data > Cari Nama Aplikasi Project (com.example.menggunakansharedpreferences)** atau menyesuaikan nama project saat dibuat > **shared_prefs** > terdapat file XML. File XML bisa secara default atau pada saat deklarasi (dalam contoh ini dengan nama "MyPrefs.XML"). Buka File tersebut, berikut hasilnya :



Saat file dibuka, menampilkan data yang telah diinput :



➤ Realm

Merupakan sebuah library database untuk perangkat mobile yang menggunakan konsep tanpa skema (noSQL). Realm digunakan untuk database sementara android atau IOS. Realm memberikan kemudahan dan performa baca dan tulis yang lebih baik dari database mobile lainnya. Realm hanya bersifat offline atau tidak dapat melakukan sinkronisasi ke server. Untuk proses sinkronisasi Realm menyediakan Realm Mobile Platform. Syarat menggunakan Realm pada Android Studio 1.5.1, JDK 7.0 minimal, SDK terbaru dan menggunakan Android API Level 9 keatas.

Untuk membuat penyimpanan data, kita dapat menggunakan realm dengan menambahkan pada build.gradle (Project) : classpath "io.realm:realm-gradle-plugin:6.1.0"

```
dependencies {
    classpath "com.android.tools.build:gradle:4.1.3"
    classpath "io.realm:realm-gradle-plugin:6.1.0"

    // NOTE: Do not place your application dependencies here; they belong
    // in the individual module build.gradle files
}
```

Tambahkan build.gradle (Module App) : id 'realm-android'

```
plugins {
    id 'com.android.application'
    id 'realm-android'
```

Buat tampilan activity atau desain XML activity sebagai berikut :

Dengan 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">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="NIM"
        android:id="@+id/editNIM"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="NAMA LENGKAP"
        android:id="@+id/editNama"
        android:layout_below="@+id/editNIM"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SIMPAN"
        android:id="@+id/btnSimpan"
        android:layout_below="@+id/editNama"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TAMPIL DATA"
        android:id="@+id/btnTampil"
        android:layout_below="@+id/btnSimpan"/>

</RelativeLayout>
```

Kemudian untuk komponen dan class kita akan menambahkan Class User yang merupakan deskripsi atau pembuatan database Realm : Klik Kanan pada com.example.namaproject > New > Java Class > User. Dengan kode berikut :

```
package com.example.databasesrealm;

import io.realm.RealmObject;
import io.realm.annotations.Ignore;
import io.realm.annotations.PrimaryKey;
import io.realm.annotations.Required;

public class User extends RealmObject {
    @PrimaryKey
    private String nim;
    @Required
    private String nama;

    public String getNim(){return nim;}
    public void setNim (String nim){this.nim = nim;}

    public String getNama(){return nama;}
    public void setNama(String nama){
        this.nama = nama;}
}
```

Tambahkan Activity baru, yaitu activity untuk menampilkan data hasil penyimpanan pada database realm. Dengan desain 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"
    tools:context=".LoadActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:id="@+id/newText"/>

</LinearLayout>
```

Kemudian, modifikasi Main Activity Java dengan kode :

```
package com.example.databasesrealm;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import io.realm.Realm;

public class MainActivity extends AppCompatActivity {
```

```

    Realm realm;
    EditText nim, nama, alamat;

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

        Button button1 = (Button) findViewById(R.id.btnSimpan);
        Button button2 = (Button) findViewById(R.id.btnTampil);
        nim = (EditText) findViewById(R.id.editNIM);
        nama = (EditText) findViewById(R.id.editNama);

        Realm.init(this);
        realm = Realm.getDefaultInstance();

        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                realm.beginTransaction();

                User user = realm.createObject(User.class,
nim.getText().toString());
                user.setNama(nama.getText().toString());

                realm.commitTransaction();
                nim.setText("");
                nama.setText("");
            }
        });
        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this, LoadActivity.class);
                startActivity(i);
            }
        });
    }
}

```

Modifikasi LoadActivity (Activity kedua) dengan kode :

```

package com.example.databasesrealm;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;

import io.realm.Realm;
import io.realm.RealmResults;

public class LoadActivity extends AppCompatActivity {
    Realm realm;
    EditText nim, nama;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentViews(R.layout.activity_load);

TextView textView = (TextView) findViewById(R.id.newText);
Realm realm = Realm.getDefaultInstance();
RealmResults<User> results1 = realm.where(User.class).findAll();
String text = "";
for (User c:results1){
    text = text+c.getNim()+"\n";
    text = text+c.getNama()+"\n";
}
textView.setText(text);
}
}

```

Hasil Running tampilan aplikasi :



Hasil tampil data, sudah diisi data sebelumnya. Maka tampilannya :



➤ SQLite

Merupakan penyimpanan data yang baik untuk data yang terstruktur. Merupakan media penyimpanan untuk aplikasi. Database SQLite dinyatakan sebagai objek SQLiteDatabase dengan kelas SQLiteOpenHelper.

Buat sebuah project dengan tampilan :

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">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="ID"
        android:id="@+id/editNIM"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nama Lengkap"
        android:id="@+id/editNama"
        android:layout_below="@+id/editNIM"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editNama"
        android:id="@+id/btnSimpan"
        android:text="SIMPAN"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btnSimpan"
        android:id="@+id/btnTampil"
        android:text="TAMPIL"/>
    <Button
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:layout_below="@+id/btnTampil"
        android:id="@+id/btnUpdate"
        android:text="UPDATE"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/btnUpdate"
    android:id="@+id/btnDelete"
    android:text="DELETE"/>

</RelativeLayout>

```

Buat sebuah Class DBHelper, Class ini merupakan fungsi membuat database, Tabel serta fungsi CRUD.

```

package com.example.sqlitedatabase;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper {
    //Nama DB
    public static final String DATABASE_NAME = "biodata.db";
    //Nama Tabel
    public static final String TABLE_NAME = "table_mahasiswa";
    //Versi DB
    public static final int DATABASE_VERSION = 1;
    //Tabel Field
    public static final String COL_1 = "ID";
    public static final String COL_2 = "NAMA";

    public DBHelper(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        SQLiteDatabase db = this.getWritableDatabase();
    }
    @Override
    public void onCreate(SQLiteDatabase db){
        db.execSQL("create table table_mahasiswa(id integer primary key
autoincrement, " + "nama text null);");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    //Simpan Data
    public boolean insertData(String nama){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_2, nama);
        long result = db.insert(TABLE_NAME, null, contentValues);
        if (result == -1)
            return false;
        else
            return true;
    }
}

```

```

//Mengambil Data
public Cursor getAllData(){
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor res = db.rawQuery("SELECT * FROM table_mahasiswa", null);
    return res;
}

//Merubah Data
public boolean updateData(String id, String nama){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_1, id);
    contentValues.put(COL_2, nama);
    db.update(TABLE_NAME, contentValues, "ID=?", new String[]{id});
    return true;
}

//Menghapus Data
public int deleteData(String id){
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(TABLE_NAME, "ID = ?", new String[]{id});
}
}

```

Modifikasi Main Activity Java :

```

package com.example.sqlitedatabase;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    DBHelper myDb;
    EditText edID, edNama;
    Button btnAddData;
    Button btnViewAll;
    Button btnUpdate;
    Button btnDelete;

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

        myDb = new DBHelper(this);
        edID = (EditText) findViewById(R.id.editNIM);
        edNama = (EditText) findViewById(R.id.editNama);
        btnAddData = (Button) findViewById(R.id.btnSimpan);
        btnViewAll = (Button) findViewById(R.id.btnTampil);
        btnUpdate = (Button) findViewById(R.id.btnUpdate);
        btnDelete = (Button) findViewById(R.id.btnDelete);

        addData();
        viewAll();
        updateData();
    }
}

```

```

        deleteData();
    }

    //Hapus
    public void deleteData(){
        btnDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Integer deletedRows = myDb.deleteData(edID.getText().toString());
                if (deletedRows > 0)
                    Toast.makeText(MainActivity.this, "Data Dihapus",
Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(MainActivity.this, "Data Gagal Dihapus",
Toast.LENGTH_LONG).show();
            }
        });
    }

    //Update
    public void updateData(){
        btnUpdate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                boolean isUpdate = myDb.updateData(edID.getText().toString(),
                    edNama.getText().toString());
                if (isUpdate == true)
                    Toast.makeText(MainActivity.this, "Data Berhasil Dirubah",
Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(MainActivity.this, "Data gagal Dirubah",
Toast.LENGTH_LONG).show();
            }
        });
    }

    //Simpan
    public void addData(){
        btnAddData.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                boolean isInserted = myDb.insertData(edNama.getText().toString());
                if (isInserted == true)
                    Toast.makeText(MainActivity.this, "Data Berhasil Disimpan",
Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(MainActivity.this, "Data Gagal Disimpan",
Toast.LENGTH_LONG).show();
            }
        });
    }

    //Tampil
    public void viewAll(){
        btnViewAll.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Cursor res = myDb.getAllData();
                if (res.getCount() == 0){
                    showMessage("Error", "Nothing Found");
                    return;
                }
            }
        });
    }

```



```

        StringBuffer buffer = new StringBuffer();
        while (res.moveToNext()){
            buffer.append("ID : " + res.getString(0)+"\n");
            buffer.append("NAMA : " + res.getString(1)+"\n");
        }
        showMessage("Data", buffer.toString());
    }
});
}
//Alert Dialog
public void showMessage(String title, String Message){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(Message);
    builder.show();
}
}

```

Hasil Run Aplikasi :



Saat Data Disimpan :



Silahkan coba untuk hasil aplikasi dengan menekan tombol yang lain.