LAPORAN PRAKTIKUM PBP

“ANDROID: INTENT”

PRAKTIKUM KE-9

LAB Logo

Description automatically generatedC2

Disusun Oleh :

**Muhamad Nur Baihaqi**

**24060120140048**

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

**2022**

1. Overview



Pada praktikum ini, akan dibuat sebuah aplikasi yang di dalamnya terdapat 3 tombol untuk berpindah activity. Tombol pertama digunakan untuk berpindah activity secara langsung tanpa ada data yang ditransfer. Tombol kedua digunakan untuk berpindah activity dengan data yang nantinya akan dikirimkan ketika terjadi proses pemindahan activity sehingga data dapat tampil di activity berikutnya. Tombol ketiga digunakan untuk melakukan pemindahan activity secara implicit yang merupakan activity bawaan dari sistem yaitu ke activity dial. Perpindahan activity dilakukan dengan menggunakan **intent**.

1. Layout dan Komponen
   1. Main Activity

Graphical user interface, application

Description automatically generated

Berikut adalah tampilan dari main activity. Digunakan sebuah layout bernama *linear layout* bertipe *vertical* yang digunakan untuk *wrapping* berbagai komponen-komponen tombol ke arah vertikal dari aplikasi sehingga tersusun secara rapih.

Setiap tombol diberikan ID untuk menandai tombol tersebut ketika digunakan dalam proses *onclick* nantinya. ID dideklarasikan pada activity\_main.xml. Selain id, kita juga bisa mendeklarasikan atribut-atribut lain seperti margin, text dan lain sebagainya di dalam activity\_main.xml. Agar lebih dinamis, semua string text disimpan dalam string.xml yang terletak dalam folder res/values/strings.xml. Hal ini bertujuan untuk memudahkan pengubahan serta penandaan string apabila diperlukan pengubahan dalam string yang digunakan.

Berikut adalah kode dari layout activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="409dp"

android:layout\_height="729dp"

android:layout\_marginTop="64dp"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<Button

android:id="@+id/btn\_move\_activity"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/btn\_move" />

<Space

android:layout\_width="match\_parent"

android:layout\_height="32dp" />

<Button

android:id="@+id/btn\_move\_activity\_data"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/btn\_move\_with\_data" />

<Space

android:layout\_width="match\_parent"

android:layout\_height="32dp" />

<Button

android:id="@+id/btn\_dial\_number"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/dial\_number" />

<Space

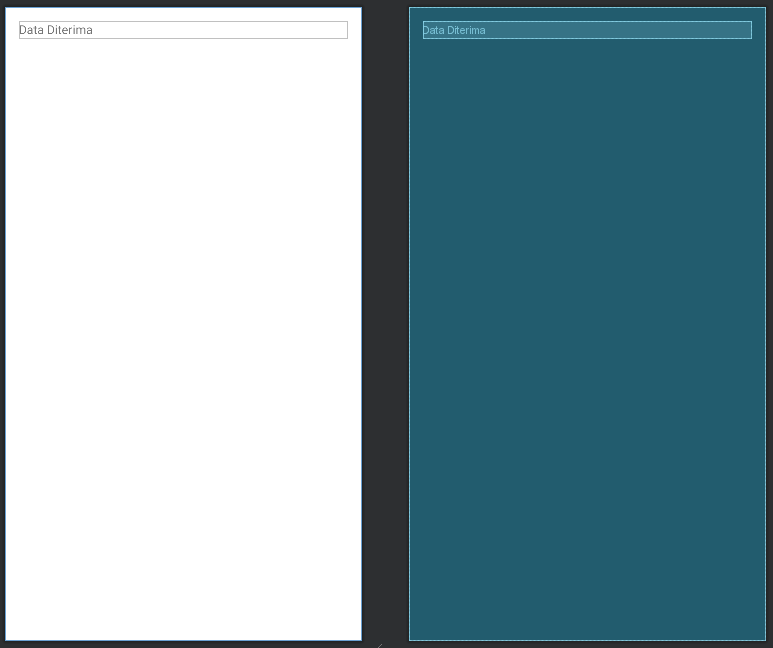
android:layout\_width="match\_parent"

android:layout\_height="32dp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

* 1. Move With Data Activity



Activity ini adalah activity yang digunakan untuk menerima data/extra yang dikirimkan oleh Main Activity. Data yang dikirimkan adalah data berupa nilai string serta integer. Data ini nantinya akan ditampilkan pada bagian dengan placeholder “Data Diterima”.

Berikut adalah kode xml dari MoveActivityWithData.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=".MoveWithDataActivity"

android:padding="16dp">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/tv\_data\_received"

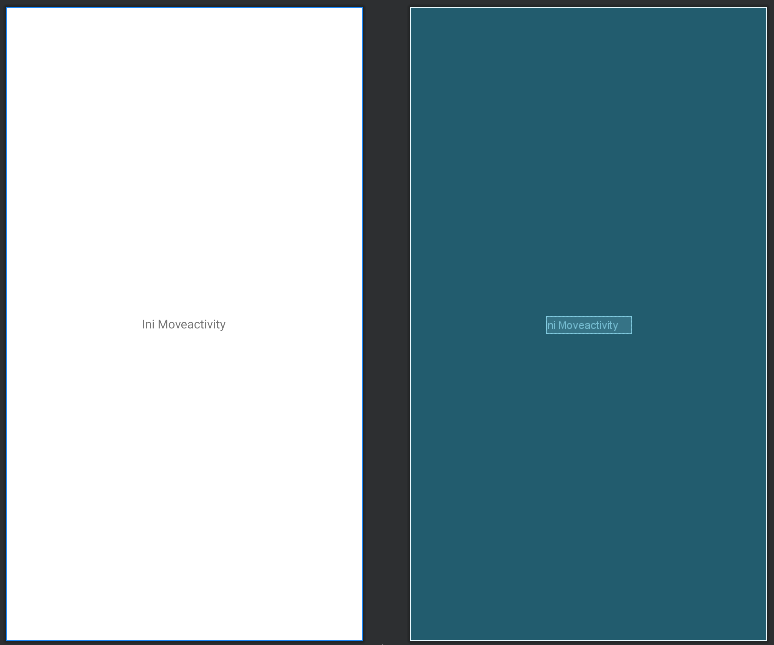
android:text="@string/data\_received"/>

</RelativeLayout>

Digunakan sebuah layout yang bernama RelativeLayout. Layout ini merupakan layout yang relatif terhadap parentnya sesuai dengan namanya. Layout bertipe tidak memiliki constraint seperti pada constraintlayout maupun linearlayout. Layout ini memiliki width dan height yang langsung menyesuaikan parent nya.

Dalam layout activity ini, hanya terdapat text view dengan id “tv\_data\_received” yang merupakan placeholder untuk data yang nanti dikirimkan oleh Main Activity.

* 1. Move Activity



Activity ini adalah activity sederhana yang terdiri dari constraint layout yang menampung sebuah textview. Tidak ada proses perpindahan extra dari Main Activity ke Activity ini. Berikut adalah layout xml dari activity ini:

<?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=".MoveActivity">

<TextView

android:id="@+id/tvvMoveActivity"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/this\_is\_moveactivity"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

1. Application Process
   1. Button 1

package id.baihaqi.myintentapp;

…

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button btnMoveActivity = findViewById(R.id.btn\_move\_activity);

btnMoveActivity.setOnClickListener(this);

}

@Override

public void onClick(View v) {

switch ( v.getId()) {

case R.id.btn\_move\_activity:

Intent moveIntent = new Intent(MainActivity.this,

MoveActivity.class);

startActivity(moveIntent);

break;

}

}

}

Button pertama digunakan untuk berpindah activity dari Main Activity ke Move Activity. MainActivity.java mengimplementasikan interface view.OnClickListener untuk dapat melakukan proses listening onclick ketika button ditekan (dalam state onclick).

Pertama, kita ambil ID dari button pertama untuk kita simpan dalam variable dengan kelas button pada ActivityMain.java dengan kode berikut:

Button btnMoveActivity = findViewById(R.id.btn\_move\_activity);

btnMoveActivity.setOnClickListener(this);

Setelah itu, kita perlu melakukan overriding method onclick default dari view MainActivity tersebut dan melakukan proses pengecekan button mana yang ditekan dengan ID yang telah kita simpan pada variabel Button tadi.

@Override

public void onClick(View v) {

switch ( v.getId()) {

case R.id.btn\_move\_activity:

Intent moveIntent = new Intent(MainActivity.this,

MoveActivity.class);

startActivity(moveIntent);

break;

}

}

}

Kemudian, kita deklarasikan sebuah objek Intent. Intent adalah jembatan penghubung yang digunakan untuk menghubungkan satu activity dengan activity lain. Untuk membuat intent, kita perlu memberikan argumen berupa activity asal dan activity tujuan. Setelah objek intent dibuat, kita bisa memulai intent dengan startActivity(namaIntent). Jangan lupa memberikan break dalam setiap case dari switch function yang digunakan.

Ketika aplikasi dijalankan dan tombol pertama diklik, maka method onClick akan berjalan dan melakukan pengecekan id tombol yang diklik. Karena id tombol yang diklik merupakan id tombol pertama, maka switch case akan menjalankan proses pembuatan intent serta perpindahan activity sesuai yang dideklarasikan pada switch case button tersebut. Selanjutnya perpindahan activity pun terjadi.

* 1. Button 2

Tombol kedua digunakan untuk melakukan perpindahan activity dari MainActivity ke MoveWithDataActivity. Perpindahan ini disertai dengan perpindahan data yang dikirimkan dari MainActivity ke MoveWithDataActivity. Oleh karena itu, diperlukan sebuah fungsi/method yang digunakan untuk melakukan handling data dari kedua activity yang saling berinteraksi tersebut.

MainActivity.java

package id.baihaqi.myintentapp;

…

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

…

Button btnMoveWithDataActivity = findViewById(R.id.btn\_move\_activity\_data);

btnMoveWithDataActivity.setOnClickListener(this);

}

}

Pertama, kita ambil id button kedua yang telah dibuat melalui layout xml dalam MainActivity.java. Setelah itu, pada bagian switch-case, buat case untuk button kedua:

@Override

public void onClick(View v) {

switch ( v.getId()) {

case R.id.btn\_move\_activity:

Intent moveIntent = new Intent(MainActivity.this,

MoveActivity.class);

startActivity(moveIntent);

break;

case R.id.btn\_move\_activity\_data:

Intent moveWithDataIntent = new Intent(MainActivity.this,

MoveWithDataActivity.class);

startActivity(moveWithDataIntent);

break;

}

}

}

Buat objek intent moveWithDataIntent dengan asal activity berupa MainActivity.this dan tujuan activity adalah MoveWithDataActivity.class. Setelah objek intent dibuat, kita akan berpindah ke MoveWithDataActivity.java

MoveWithDataActivity.java

…

public class MoveWithDataActivity extends AppCompatActivity {

public static final String EXTRA\_AGE = "extra\_age";

public static final String EXTRA\_NAME = "extra\_name";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_move\_with\_data);

TextView tvDataReceived = findViewById(R.id.tv\_data\_received);

String name = getIntent().getStringExtra(EXTRA\_NAME);

int age = getIntent().getIntExtra(EXTRA\_AGE, 0);

String text = "Name: " + name + ", \nYour Age: " + age;

tvDataReceived.setText(text);

}

}

Buat sebuah static final string yang digunakan untuk melakukan penamaan data. Data yang telah dinamai antara lain EXTRA\_AGE dan EXTRA\_NAME. Data yang telah dinamai kemudian dapat digunakan sebagai variabel konstanta dalam variabel lain yang membutuhkan nilainya. Berikut contoh penggunaan data yang telah dilakukan penamaan:

String name = getIntent().getStringExtra(EXTRA\_NAME);

int age = getIntent().getIntExtra(EXTRA\_AGE, 0);

…

Untuk mendapatkan data, perlu digunakan sebuah method getIntent() dan dilanjutkan dengan method getStringExtra dengan argumen dengan nama data/extra yang ingin diambil value-nya. Nama extra tersebut disesuaikan dengan nama extra/data yang telah kita buat pada bagian sebelumnya. Setleah itu, kita bise melakukan method setText() pada TextView.

MainActivity.java

@Override

public void onClick(View v) {

switch ( v.getId()) {

case R.id.btn\_move\_activity:

Intent moveIntent = new Intent(MainActivity.this,

MoveActivity.class);

startActivity(moveIntent);

break;

case R.id.btn\_move\_activity\_data:

Intent moveWithDataIntent = new Intent(MainActivity.this,

MoveWithDataActivity.class);

moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME,

"Joe Mama");

moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE,

99);

startActivity(moveWithDataIntent);

break;

}

}

Pada bagian MainActivity.java, kita bisa menggunakan method putExtra dan menambahkan nama extra serta nilai extra yang akan dikirimkan. Setelah itu, kita bisa melakukan startActivity()

* 1. Button 3

Button ketiga digunakan untuk memanggil implicit intent yang merupakan intent yang dapat dipanggil tanpa harus membuat objek intentnya seperti pada bagian-bagian sebelumnya. Ada banyak sekali intent implisit yang biasanya merupakan bawaan sistem android. Seperti pada button-button sebelumnya, kita perlu mengambil id button ketiga serta menambahkan proses yang ingin dilakukan ketika button ketiga diklik pada switch-case di ActivityMain.java

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

…

Button btnDialPhone = findViewById(R.id.btn\_dial\_number);

btnDialPhone.setOnClickListener(this);

}

@Override

public void onClick(View v) {

switch ( v.getId()) {

…

**case R.id.btn\_dial\_number:**

**String phoneNumber = "081353000166";**

**Intent dialPhoneNumber = new Intent(Intent.ACTION\_DIAL, Uri.parse("tel:"+phoneNumber));**

**startActivity(dialPhoneNumber);**

**break; }**

}

}

Deklarasi pembuatan intent implisit cukup berbeda dari intent sebelumnya karena kita tidak perlu membuat layout activity yang akan dituju, kita hanya perlu memberikan argumen berupa Intent.<NAMA\_INTENT\_IMPLISIT> yang dilakukan serta uri yang dikirimkan. Dalam hal ini, intent ACTION\_DIAL dipanggil yang merupakan intent dari aplikasi telepon bawaan android. URI yang dimasukkan akan tampil sebagai nomor telepon yang default terisi ketika intent dijalankan

1. Link GitHub

<https://github.com/baihaqi2193/intent-app>