

# **BAB I      II**

## **Android: Intent**

### **1. Tujuan**

Mahasiswa mampu membuat Aplikasi Sederhana dengan memanfaatkan Intent

### **2. Tools**

Android Studio

### **3. Dasar Teori**

Intent adalah mekanisme untuk melakukan sebuah action dan komunikasi antar komponen aplikasi misal activity, services, dan broadcast receiver. Ada tiga penggunaan umum intent dalam aplikasi Android yaitu:

- Memindahkan satu activity ke activity lain dengan atau tidak membawa data.
- Menjalankan background service, misalnya melakukan sinkronisasi ke server dan menjalankan proses berulang (periodic/scheduler task).
- Mengirimkan obyek broadcast ke aplikasi yang membutuhkan. Misal, ketika aplikasi membutuhkan proses menjalankan sebuah background service setiap kali aplikasi selesai melakukan booting. Aplikasi harus bisa menerima obyek broadcast yang dikirimkan oleh sistem Android untuk event booting tersebut.

Intent memiliki dua bentuk yaitu:

#### **1. Explicit Intent**

Adalah tipe Intent yang digunakan untuk menjalankan komponen dari dalam sebuah aplikasi. Explicit intent bekerja dengan menggunakan nama kelas yang dituju misal : `com.pbp.activity.DetailActivity`. Umumnya intent ini digunakan untuk mengaktifkan komponen pada satu aplikasi.

#### **2. Implicit Intent**

Adalah tipe intent yang tidak memerlukan detail nama kelas yang ingin diaktifkan. Model ini memungkinkan komponen dari aplikasi lain bisa merespon request intent yang dijalankan.

Penggunaan tipe intent ini umumnya diperuntukkan untuk menjalankan fitur/fungsi dari komponen aplikasi lain. Contohnya ketika kita membutuhkan fitur untuk mengambil foto. Daripada membuat sendiri fungsi kamera, lebih baik kita menyerahkan proses tersebut

pada aplikasi kamera bawaan dari peranti atau aplikasi kamera lain yang telah terinstal sebelumnya di peranti.

Hal yang sama misalnya ketika kita membutuhkan fungsi berbagi konten. Kita bisa memanfaatkan intent untuk menampilkan aplikasi mana saja yang bisa menangani fitur tersebut.

Implementasi implicit intent ini akan sangat memudahkan bagi pengembang agar tetap fokus pada proses bisnis inti dari aplikasi yang dikembangkan.

#### 4. Langkah Praktikum

##### a. Membuat Proyek Baru

Buat proyek baru dengan ketentuan sebagai berikut:

<b>Nama Project</b>	MyIntentApp
<b>Target &amp; Minimum Target SDK</b>	Api Level >21
<b>Tipe Activity</b>	Empty Activity
<b>Language</b>	Java

##### b. Intent Sederhana

###### 1. Tambahkan kode berikut pada **activity\_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/btn_move_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/move_activity" />

</LinearLayout>
```

###### 2. Tambahkan juga resource string-nya. Tambahkan semua string yang akan digunakan

di project ini. Buka berkas **strings.xml** yang ada di folder **res/value** dan tambahkan kode berikut ini.

```
<resources>
    <string name="app_name">MyIntentApp</string>
    <string name="move_activity">Pindah Activity</string>
    <string name="move_with_data">Pindah Activity dengan
Data</string>
    <string name="dial_number">Dial a Number</string>
    <string name="data_received">Data Diterima</string>
    <string name="this_is_moveactivity">Ini MoveActivity</string>
</resources>
```

3. Kemudian masuk ke **MainActivity**, tambahkan beberapa baris kode yang berfungsi untuk memperkenalkan button yang sudah ditambahkan di layout seperti ini: (Jangan lupa **import class Button**)

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

    Button btnMoveActivity = findViewById(R.id.btn_move_activity);
}
```

4. Lalu tambahkan beberapa baris yang berfungsi untuk menambahkan event onClick pada button btnMoveActivity seperti ini.

```
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);
    }

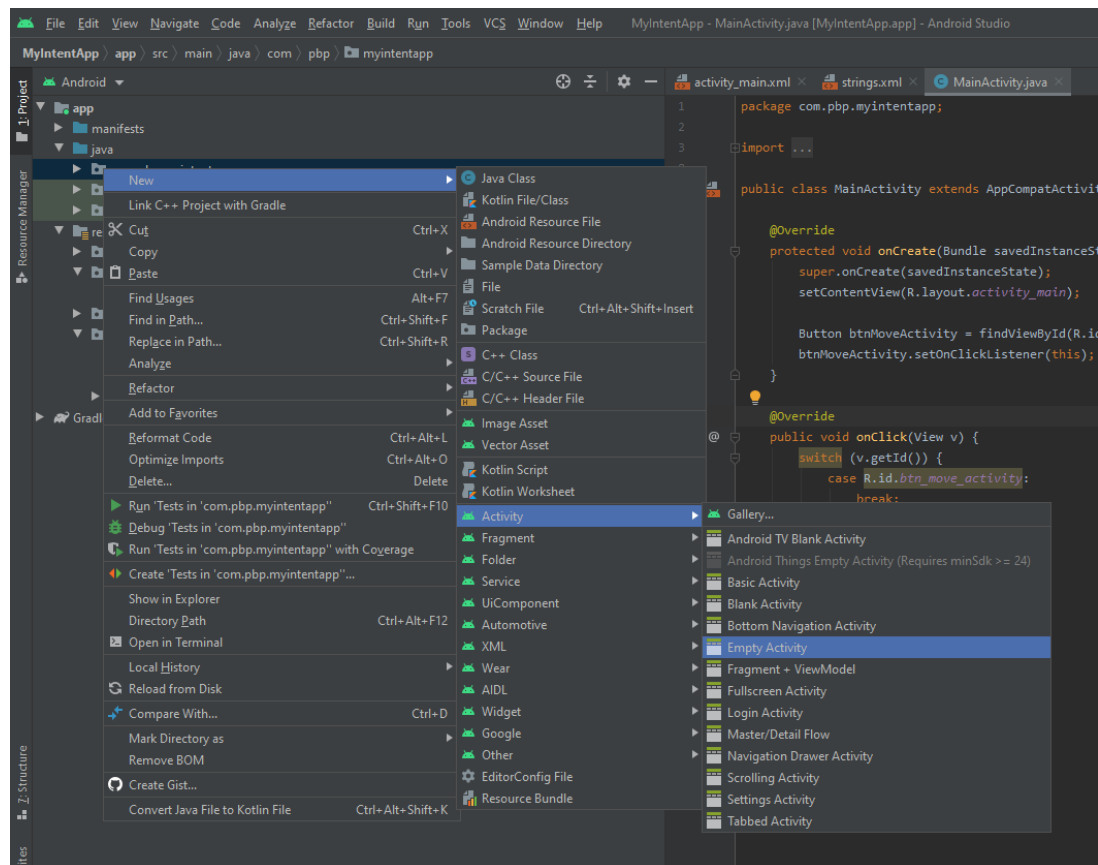
    @Override
```

```

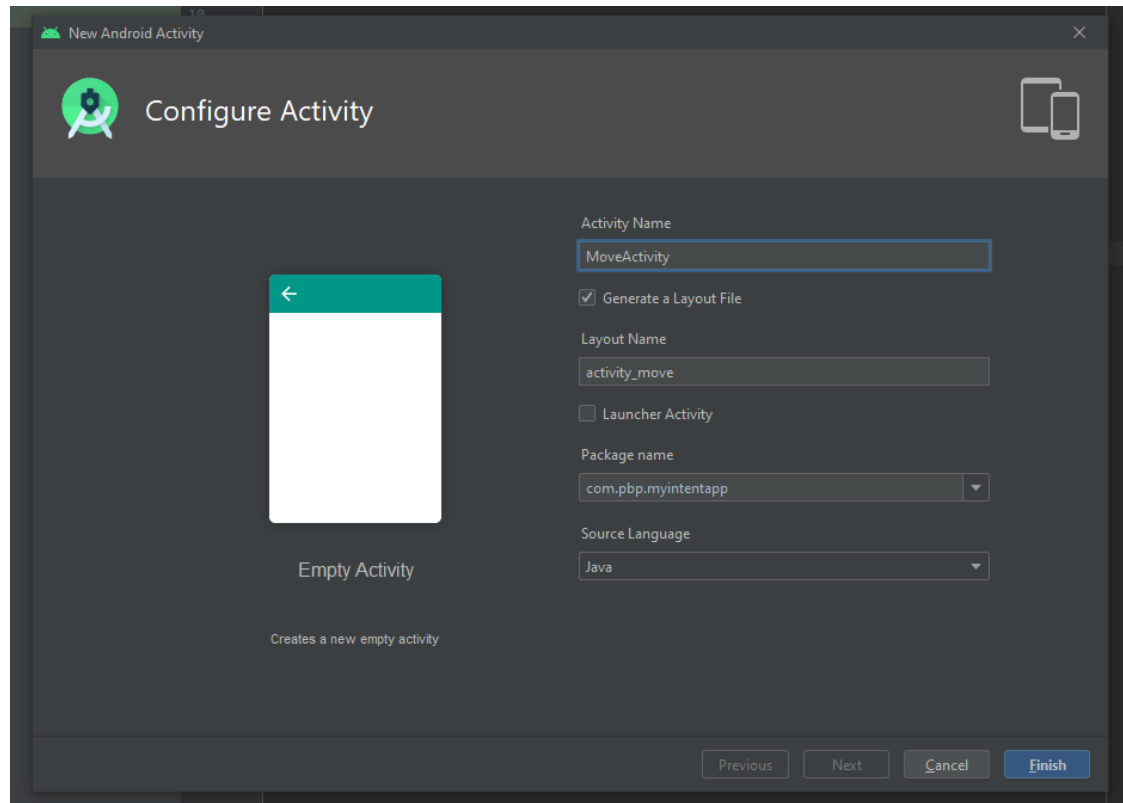
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.btn_move_activity:
                    break;
            }
        }
    }
}

```

5. Button **btnMoveActivity** akan memiliki fungsi untuk berpindah Activity ke Activity lain. Sekarang kita buat Activity baru dengan cara sebagai berikut: **Klik kanan** di package utama aplikasi **package name** → **New** → **Activity** → **Empty Activity**.



Lalu isikan MoveActivity pada dialog. Jangan lupa untuk memastikan bahasa yang digunakan sudah sesuai dengan yang anda gunakan sebelumnya



6. Untuk menandakan bahwa perpindahan activity berhasil, silakan tambahkan satu **TextView** dan kondisikan **activity\_move.xml** menjadi seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/this_is_moveactivity" />

</RelativeLayout>
```

7. Setelah Activity tujuan sudah berhasil diciptakan, sekarang saatnya menambahkan suatu Intent pada method **onClick()** di **MainActivity.java** menjadi sebagai berikut. (Jangan lupa **import class Intent**)

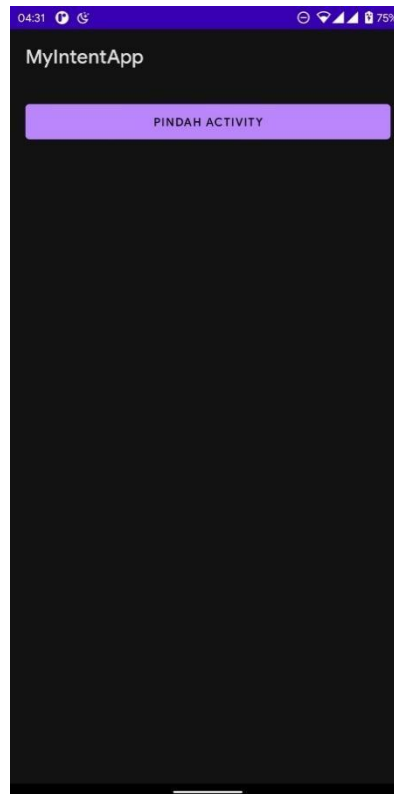
```
@Override
public void onClick(View v) {
```

```

switch (v.getId()) {
    case R.id.btn_move_activity:
        Intent moveIntent = new Intent(MainActivity.this,
        MoveActivity.class);
        startActivity(moveIntent);
        break;
}
}

```

8. Selesai! Langkah pertama untuk memindahkan satu Activity ke Activity lain sudah selesai, sekarang silakan jalankan aplikasi Anda dengan mengklik tombol pada menu bar. Seharusnya sekarang anda sudah bisa memindahkan Activity dengan mengklik tombol 'Pindah Activity'.



### c. Mengirim Data pada Intent

Membuat sebuah Intent yang di dalamnya akan membawa data ke Activity tujuan.\

1. Buka **activity\_main.xml**, kemudian tambahkan satu tombol lagi di bawah tombol sebelumnya. Kode **activity\_main.xml** kita pun, jadi seperti ini.

```

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

```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp">

...

<Button
    android:id="@+id/btn_move_activity_data"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:text="@string/move_with_data" />
</LinearLayout>

```

2. Setelah selesai dengan penambahan pada berkas **activity\_main.xml**, maka lanjutkan dengan menambahkan beberapa baris berikut di **MainActivity.java**.

```

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);
    }

    @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:
                break;
        }
    }
}

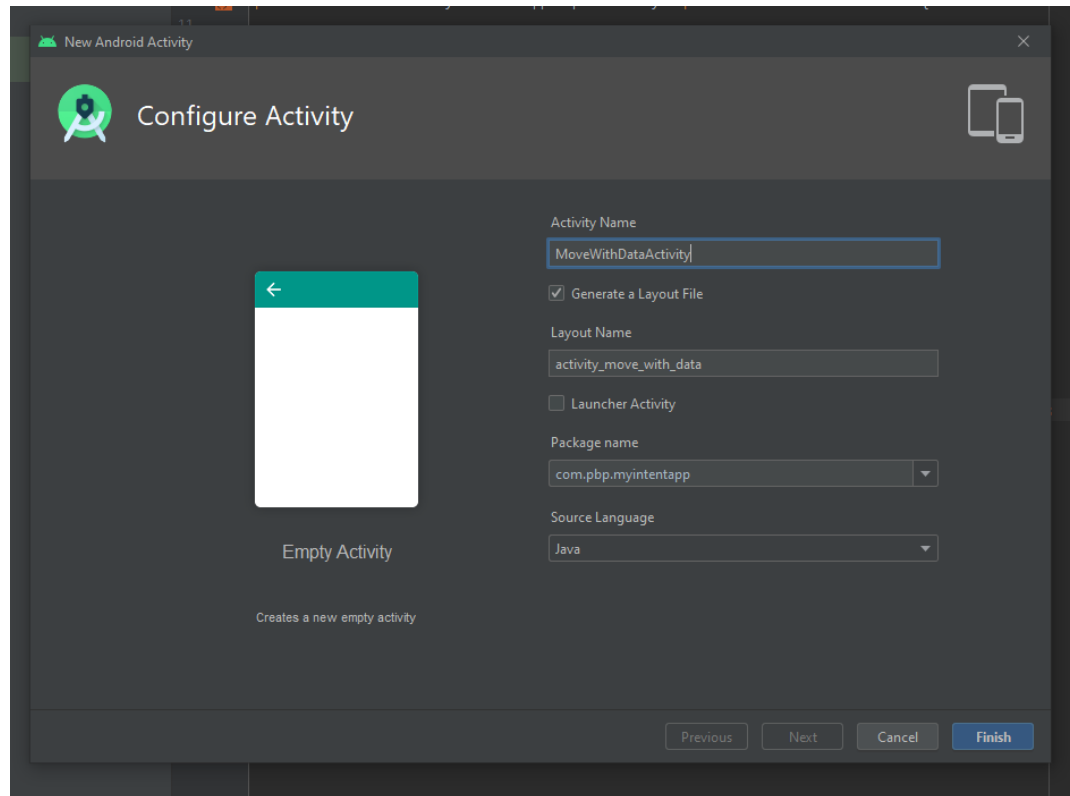
```

```

    }
}
}

```

3. Selanjutnya, buat Activity baru lagi seperti cara sebelumnya dan beri nama **MoveWithDataActivity**.



4. Lalu pada layout **activity\_move\_with\_data.xml** kita ubah layoutnya menjadi **RelativeLayout** dan tambahkan sebuah **TextView** ber-ID untuk menampilkan data yang dikirimkan dari **Activity** asal.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
    <TextView
        android:id="@+id/tv_data_received"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/data_received" />
</RelativeLayout>

```



5. Selanjutnya pada **MoveWithDataActivity.java** kita kondisikan **TextView** seperti berikut ini :

```
public class MoveWithDataActivity extends AppCompatActivity {

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

        TextView tvDataReceived = findViewById(R.id.tv_data_received);
    }
}
```

6. Selanjutnya untuk menerima data dari Activity asal, kondisikan **MoveWithDataActivity** seperti berikut :

```
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);
    }
}
```

7. Dan sekarang saatnya kita menambahkan obyek Intent pada MainActivity.java seperti baris yang ditebalkan.

```
@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,
"Praktikum PBP Go Brrrrr");

            moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
            startActivity(moveWithDataIntent);
            break;
    }
}

```

8. Maka MainActivity.java akan jadi seperti ini:

```

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);

        Button btnMoveWithDataActivity =
findViewById(R.id.btn_move_activity_data);
        btnMoveWithDataActivity.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;

```

```

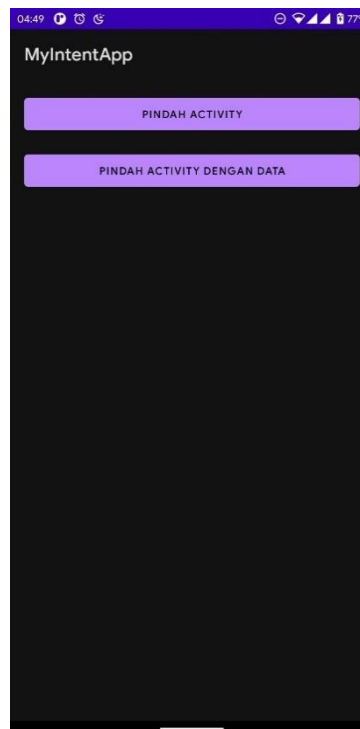
        case R.id.btn_move_activity_data:
            Intent moveWithDataIntent = new
            Intent(MainActivity.this, MoveWithDataActivity.class);

            moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "
            Praktikum PBP Go Brrrrr");

            moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
            startActivity(moveWithDataIntent);
            break;
        }
    }
}

```

9. Silakan jalankan aplikasi Anda, kemudian coba klik pada tombol Pindah Activity dengan Data. Seharusnya Anda sudah bisa memindahkan satu Activity ke Activity lain dengan membawa data.



#### d. Intent Implicit

1. Buka kembali **activity\_main.xml** tambahkan satu tombol lagi sebagai berikut:

```

...

<Button

```

```
android:id="@+id/btn_dial_number"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:text="@string/dial_number" />
```

2. Sekarang, buka kembali **MainActivity.java** dan lanjutkan menginisialisasi button yang sudah ditambahkan:

```
@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);
}
```

3. Kemudian dengan melakukan casting tombol Dial Number, set listener dan menambahkan action ketika tombol diklik. (Jangan lupa import class Uri)

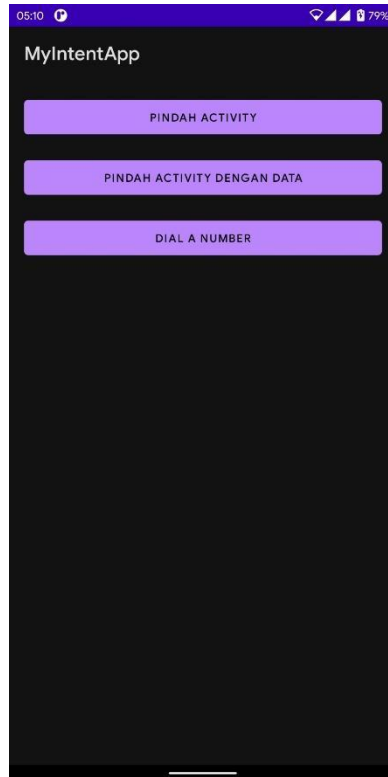
```
@Override
public void onClick(View v) {
    switch (v.getId()) {

        ...

        case R.id.btn_dial_number:
            String phoneNumber = "081210841382";
            Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL,
            Uri.parse("tel:"+phoneNumber));
            startActivity(dialPhoneIntent);
            break;

    }
}
```

4. Setelah selesai, silakan jalankan lagi aplikasi Anda. Yeay, Anda sudah bisa men-dial sebuah nomor telepon melalui aplikasi bawaan dari peranti anda. Selamat!



## 5. Tugas

1. Membuat Aplikasi sesuai dengan yang ada di modul
2. Membuat Video yang berisi tentang Penjelasan kode dan praktikum minggu ini