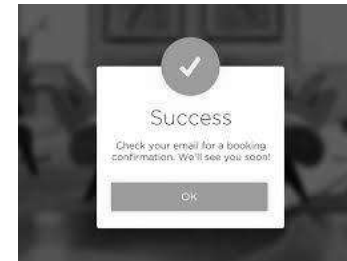




NOTIFIKASI

- Bagian informasi yang akan muncul atau sengaja dilihat dibuat nuncul selama class tersebut berjalan.



3

DIALOG

- Jendela kecil atau popup yang muncul di depan layar aplikasi dan juga kita dapat memodifikasi bagaimana muncul konformasi seperti (Yes Or No).

4

Kotak Dialog

Android secara primitif memiliki 2 bentuk dialog.

1. AlertDialog, dan
2. Toast



- Sebuah **AlertDialog** menampilkan layar dialog dalam bentuk floating(mengambang) dan menunggu pengguna untuk menekan tombol untuk ditutup/diberhentikan.
- Sebuah Toast menampilkan pesan dialog dalam bentuk flash dan akan hilang dalam waktu kurang lebih 4 detik.

5

Kotak Peringatan(Alert)

AlertDialog biasanya merupakan layar yang tampil menonjol(modal window) yang

1. Menyajikan pesan singkat ke pengguna, biasanya ditampilkan dengan cara sebuah layar kecil yang muncul dengan mengaburkan tampilan dibelakangnya. Dan,
2. Mengumpulkan jawaban sederhana (biasanya dengan menekan tombol yang harus dipilih)



Catatan :

Tampilan modal akan menunggu input dari pengguna. Cara ini bisa dihentikan dengan cara menekan tombol yang harus dipilih oleh pengguna

6

Toast

- Toast adalah pesan singkat yang berisi pesan sederhana yang disampaikan kepada user. Kemunculannya dalam waktu singkat(3-4 detik) dan tidak memiliki fokus objek seperti halnya Alert Dialog



7

contoh

```
public void toastClick(View v) {  
    Toast t = Toast.makeText(this, "Toast Short", Toast.LENGTH_SHORT);  
    t.show();  
    Toast t2 = Toast.makeText(this, "Toast Long", Toast.LENGTH_LONG);  
    t2.show();  
    Toast t3 = Toast.makeText(this, "Toast ada di tengah", Toast.LENGTH_SHORT);  
    t3.setGravity(Gravity.CENTER|Gravity.CENTER, 0, 0);  
    t3.show();  
}
```

8

Toast

- Parameter pemanggilan Toast :

Toast.makeText(context, message, duration).show()

- **Context** : Sebuah referensi untuk melihat activity sedang aktif pada saat ini
- **Message** : Pesan yang akan disampaikan pada Toast
- **Duration** : Waktu yang akan digunakan untuk menampilkan pesan, LENGTH.SHORT dan LENGTH.LONG

9

Toast

Mengubah posisi tampilan Toast :

- secara default Toast ditampilkan pada posisi bawah tengah pada layar. Tetapi kita bisa mengubah penempatan posisi dari Toast dengan beberapa metode, seperti dibawah ini:

`void setGravity(int gravity, int xOffset, int yOffset)`

- mengatur lokasi dimana pemberitahuan akan ditampilkan pada layar.

`void setMargin(float horizontalMargin, float verticalMargin)`

- mengatur tepi dari pesan.

10

Toast

Mengubah posisi tampilan Toast :

- Metode berikut menggunakan nilai offset berdasarkan resolusi pixel pada layar device. Misalnya, pada layar device dengan resolusi 320x480 pixel

`void setGravity(int gravity, int xOffset, int yOffset)`

- Gravity : penempatan secara keseluruhan. Biasanya nilainya berisi: Gravity.CENTER, Gravity.TOP, Gravity.BOTTOM, ...
- xOffset : diasumsikan, nilai penempatan Gravity.CENTER pada layar resolusi 320x480, maka jangkauan xOffset adalah -160,...,0,...,160(kiri, tengah, kanan)
- yOffset : jangkauannya meliputi -240,...,0,...,240(atas, tengah,bawah)



11

Toast

Mengubah posisi tampilan Toast :

- Metode kedua untuk menempatkan Toast adalah setMargin. Layar dianggap memiliki titik pusat/tengah dimana pusat garis horisontal dan vertikal bertemu. Ada 50% dari layar untuk setiap sisi dari titik pusat (atas,bawah,kiri,kanan). Margin dinyatakan sebagai nilai antara : -50,...,0,...,50

`void setMargin(float horizontalMargin, float verticalMargin)`

- catatan :
- Pasangan margin(-50, -50) mewakili sudut kiri atas dari layar, (0,0) adalah pusat dan (50,50) adalah pojok kanan bawah

12

Toast

- Contoh Mengubah posisi tampilan Toast :**
- Dengan menggunakan metode setGravity(...) Gravity.CENTER dan x serta y offset dari :
- 0,0 (tengah)
- 160, -240 (kiri-atas)
- 160, 240 (kanan-bawah)

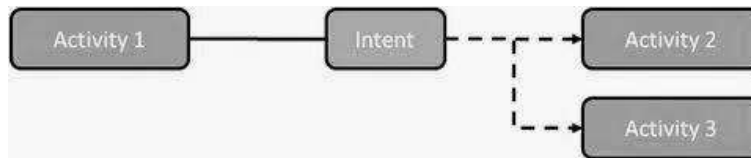


13

INTENT

- Pembuatan intent dilakukan pada activity untuk memanggil activity lainnya

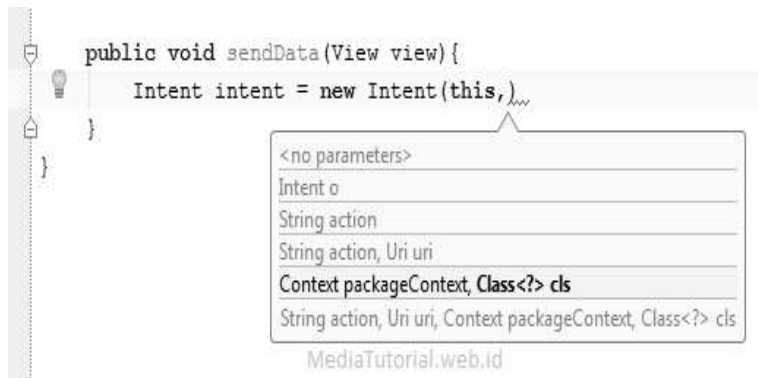
```
1 Intent intent = new Intent(this, MenuKedua.class);  
2 startActivity(intent);
```



14

IMPLICIT INTENT

memanggil aplikasi dengan memberikan parameter **action** dan **data untuk action**



16

Pengaktifan Activity

Implicit

Intent yang penggunaannya untuk memanggil fungsi yang sudah terdapat di internal Android dengan kata lain Implicit Intent adalah intent yang memanggil aplikasi dengan memberikan parameter action dan data untuk action.

Explicit

Intent yang penggunaannya untuk memanggil activity yang lain. Bisa juga diartikan intent Explicit adalah intent yang secara eksplisit mendefinisikan komponen yang harus dipanggil oleh Android System dengan menggunakan Class Java sebagai identifier-nya

15

```
1. // Create the text message with a string  
2. Intent sendIntent = new Intent();  
3. sendIntent.setAction(Intent.ACTION_SEND);  
4. sendIntent.putExtra(Intent.EXTRA_TEXT,  
    textMessage);  
5. sendIntent.setType(HTTP.PLAIN_TEXT_TYPE); //  
    "text/plain" MIME type  
6.  
7. // Verify that the intent will resolve to an  
    activity  
8. if  
    (sendIntent.resolveActivity(getPackageManager())  
    != null) {  
9.     startActivity(sendIntent);  
10. }
```

17

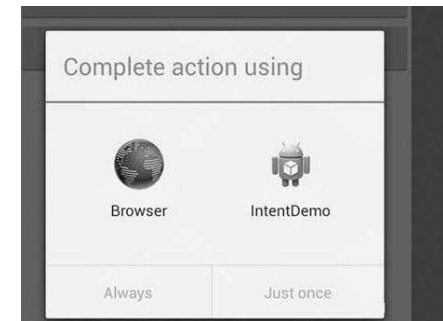
contoh

Intent i = new Intent(Intent.ACTION_VIEW,Uri.parse("http://www.uad.ac.id"));

Kita memakai parameter **action** nya adalah **Intent.ACTION_VIEW**

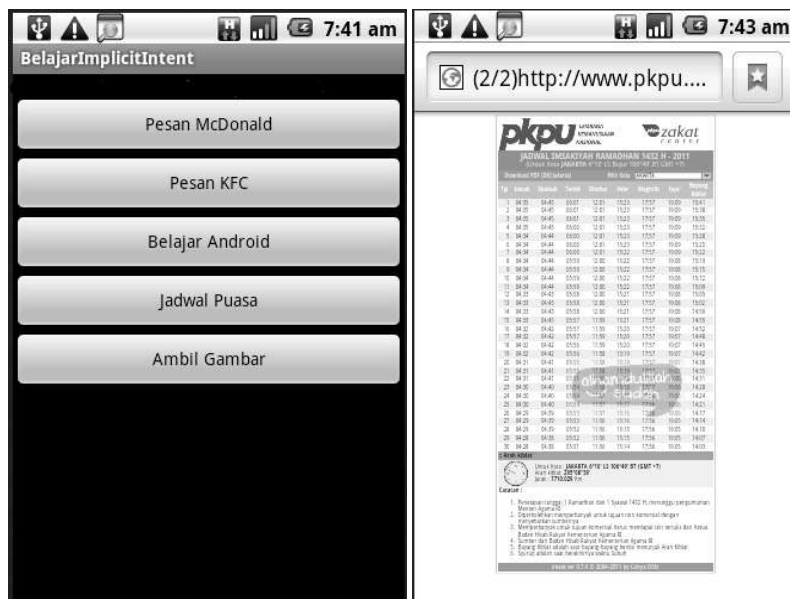
dan **data** action nya adalah **Uri.parse("http://www.uad.ac.id")**

Intent pada slide sebelumnya akan memanggil **SEMUA APLIKASI/KOMPONEN** pada Android yang sudah teregister pada system Android dengan action VIEW tersebut. Nanti, Android System akan menampilkan semuanya kemudian pengguna bisa memilih aplikasi mana yang akan dipakai.



18

19



20

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text=" Belajar Implicit Intent" />
    <Button
        android:layout_height="wrap_content"
        android:layout_width="match_parent" android:text="Pesan McDonald"
        android:id="@+id/btnMcDonald">
    </Button>
    <Button
        android:layout_height="wrap_content"
        android:layout_width="match_parent" android:text="Pesan KFC"
        android:id="@+id/btnKFC">
    </Button>
    <Button
        android:text="Belajar Android" android:layout_height="wrap_content"
        android:layout_width="match_parent" android:id="@+id/btnBelajarAndroid">
    </Button>
    <Button
        android:text="Jadwal Puasa" android:layout_height="wrap_content"
        android:id="@+id/btnJadwalPuasa" android:layout_width="match_parent">
    </Button>
    <Button
        android:text="Ambil Gambar" android:layout_height="wrap_content"
        android:id="@+id/btnKamera" android:layout_width="match_parent">
    </Button>
</LinearLayout>
```

21

Disini pada masing-masing button dipasang **setOnClickListener**. Agar pada saat kita mengklik atau sentuh button tersebut akan mengeksekusi method `onClick` didalamnya.

```
Button btnMcDonald = (Button)findViewById(R.id.btnMcDonald);
Button btnKFC = (Button)findViewById(R.id.btnKFC);
Button btnBelajarAndroid = (Button)findViewById(R.id.btnBelajarAndroid);
Button btnJadwalPuasa = (Button)findViewById(R.id.btnJadwalPuasa);
Button btnKamera = (Button)findViewById(R.id.btnKamera);

btnMcDonald.setOnClickListener(new Button.OnClickListener(){
    public void onClick(View v){
        // getRequest(txtResult,txtUrl);
        CallIntent(v);
    }
});

btnKFC.setOnClickListener(new Button.OnClickListener(){
    public void onClick(View v){
        // getRequest(txtResult,txtUrl);
        CallIntent(v);
    }
});

btnBelajarAndroid.setOnClickListener(new Button.OnClickListener(){
    public void onClick(View v){
        // getRequest(txtResult,txtUrl);
        CallIntent(v);
    }
});

btnJadwalPuasa.setOnClickListener(new Button.OnClickListener(){
    public void onClick(View v){
        // getRequest(txtResult,txtUrl);
        CallIntent(v);
    }
});

btnKamera.setOnClickListener(new Button.OnClickListener(){
    public void onClick(View v){
        // getRequest(txtResult,txtUrl);
        CallIntent(v);
    }
});
```

22

Method `CallIntent` berfungsi untuk menjalankan Suatu Activity berdasarkan button yang diklik.

```
public void CallIntent(View view) {
    Intent intent = null;
    switch (view.getId()) {
        case R.id.btnKFC:
            //akan melakukan Call ke nomor 14022
            intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:14022"));
            startActivity(intent);
            break;
        case R.id.btnMcDonald:
            //akan melakukan Call ke nomor 14045
            intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:14045"));
            startActivity(intent);
            break;
        case R.id.btnBelajarAndroid:
            //akan memanggil browser dan menampilkan website http://uad.ac.id
            intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://uad.ac.id"));
            startActivity(intent);
            break;
        case R.id.btnJadwalPuasa:
            //akan memanggil browser dan menampilkan website http://www.pkpu.or.id/imsyak/
            intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.pkpu.or.id/imsyak/"));
            startActivity(intent);
            break;
        case R.id.btnKamera:
            //akan memanggil fungsi Camera android
            intent = new Intent("android.media.action.IMAGE_CAPTURE");
            startActivityForResult(intent, 0);
            break;
        default:
            break;
    }
}
```

23

EXPLICIT INTENT

Didefinisikan secara langsung ketika akan mengaktifkan suatu komponen dengan memanggil class name-nya

```
public void sendData(View view){
    Intent intent = new Intent(this,DisplayActivity.class);
    EditText edittext_datanama = (EditText) findViewById(R.id.txtNama);
    String dataNama = edittext_datanama.getText().toString();
    intent.putExtra(DATA_NAMA,dataNama);
    startActivity(intent);
}
```

MediaTutorial.web.id

24

Kita membuat Intent dengan menggunakan dua parameter.

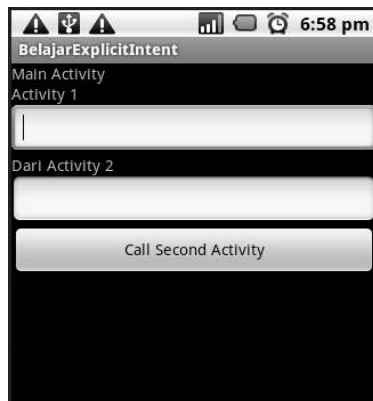
Parameter yang kedua itulah (`DisplayActivity.class`) yang dimaksud Class Java.

Class inilah yang digunakan sebagai identifier.

Explicit intent ini kebanyakan digunakan interaksi antar activity dalam satu aplikasi.

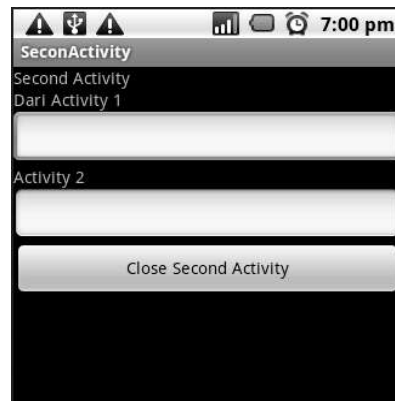
Lalu bagaimana cara mengirim dan menangkap data yang dikirim ?

25



```
//method for call SecondActivity
```

```
public void callIntent(){
    Intent i = new Intent(this, SecondActivity.class);
    startActivity(i);
}
```



26



```
public void callIntent(){
    Intent i = new Intent(this, SecondActivity.class);
    EditText text1 = (EditText) findViewById(R.id.editTextdr1);
    String dataactivity_1 = text1.getText().toString();
    //untuk mengirim data gunakan putExtra
    i.putExtra("value1", dataactivity_1 );
    startActivity(i);
}
```

27



Untuk mengirimkan data gunakan perintah putExtra, Kemudian di secondactivity.javanya ditambahkan :

```
Bundle extras = getIntent().getExtras();

if (extras == null) {
    return;
}

String value1 = extras.getString("value1");

if (value1 != null ) {
    EditText text1 = (EditText)
findViewById(R.id.editText1);

    text1.setText(value1);
}
```

28

Untuk membaca data yang dikirim dari SecondActivity pada saat Activity tersebut ditutup. Maka pada activityutama perlu menambahkan kode :

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (resultCode == RESULT_OK && requestCode ==
REQUEST_CODE) {
        if (data.hasExtra("return_value1")) {
            EditText text2 = (EditText)
findViewById(R.id.editTextdr2);

            text2.setText(data.getExtras().getString("return_value1"));
        }
    }
}
```

29

Pengaktifan Activity

Secara explicit activity diaktifkan dengan menyebutkan nama Class pada intent, dan Activity class tersebut berada pada project yang sama.

Secara implicit activity diaktifkan melalui Action pada intent, dan URI. Pada umumnya Activity class berada pada aplikasi eksternal. Activity class yang diaktifkan melalui intent filter terhadap Action dan URI.

30

KESIMPULAN

Intent adalah mekanisme untuk melakukan Action pada komponen dalam satu aplikasi yang sama maupun aplikasi lain. Pemanfaatannya sebagai berikut :

- Untuk berpindah halaman dari satu Activity ke Activity lain dengan atau tanpa data
- Untuk menjalankan background Service misal dibutuhkan sebuah proses service untuk mengambil lokasi pengguna, download file atau sinkronisasi ke server
- Untuk menyampaikan sebuah objek dari komponen Broadcast misal jika ingin mengetahui jika device Android sudah selesai booting setelah diaktifkan.

• **Dalam pemanggilan Intent dibedakan menjadi dua :**

- **Explicit Intent**
Untuk mengaktifkan komponen-komponen dalam satu aplikasi yang sama
- **Implicit Intent**
Untuk mengaktifkan komponen dari aplikasi lain.

32

Mengaktifkan Activity secara Explicit



```
public class MenuActivity extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    public void segitigaOnClick(View v) {  
        Intent intent = new Intent();  
        intent.setClass(this, SegitigaActivity.class);  
        startActivity(intent);  
    }  
  
    public void selesaiOnClick(View v) {  
        finish();  
    }  
}
```

¹ Secara explicit activity diaktifkan dengan menyebutkan nama Class, dan Class tersebut berada pada project yang sama.

Secara program anda dapat mengakhiri suatu Activity dengan memanggil metode finish().

31