

Tutorial 5

Android Broadcast Receiver

Android BroadcastReceiver adalah komponen aktif Android yang mendengarkan system-wide broadcast events atau intents.

Ketika salah satu dari peristiwa ini terjadi, aplikasi akan beraksi dengan membuat pemberitahuan status bar atau melakukan sebuah tugas.

Tidak seperti activity, BroadcastReceiver android tidak memiliki antarmuka pengguna apa pun. Penerima broadcast umumnya diterapkan untuk mendelegasikan tugas ke service tergantung pada jenis intent yang diterima.

Berikut adalah beberapa system wide generated intents:

android.intent.action.BATTERY_LOW	:	Mengindikasikan kondisi baterai lemah di perangkat.
android.intent.action.BOOT_COMPLETED	:	Disiarkan hanya sekali setelah sistem selesai melakukan booting.
android.intent.action.CALL	:	Melakukan sebuah panggilan ke seseorang.
android.intent.action.DATE_CHANGED	:	Perubahan tanggal
android.intent.action.REBOOT	:	Ketika mereboot sistem.
android.net.conn.CONNECTIVITY_CHANGE	:	Ketika jaringan perangkat atau WiFi berubah.

Untuk mengatur Broadcast Receiver di aplikasi Android, kita perlu melakukan:

1. Creating a BroadcastReceiver

Buatlah sebuah project baru dengan nama ApplicationBroadcastReceiver, lalu buatlah sebuah Activity baru dengan nama **ActivityReceiver.java** lalu isikan baris kodingan berikut:

```
package com.example.tutorial5;

import ...

public class ActivityReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context, text: "Action: " + intent.getAction(), Toast.LENGTH_SHORT).show();
    }
}
```

BroadcastReceiver merupakan sebuah kelas abstrak fungsi onReceive() sebagai fungsi abstraknya. Fungsi onReceive() dipanggil pertama kali saat registrasi Broadcast Receiver kapan saja sebuah event muncul.

Objek Intent akan dikirimkan bersamaan dengan semua data tambahan. Sebuah objek Context juga tersedia dan dapat digunakan untuk memulai sebuah activity atau service menggunakan perintah **context.startActivity(objIntent);** atau **context.startService(objService);**

2. Registering a BroadcastReceiver

Untuk meregistrasikan Broadcast Receiver, kita bisa memilih melalui file manifest atau menuliskannya di badan program.

File Manifest

```
<receiver android:name=".ActivityReceiver" >
    <intent-filter>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    </intent-filter>
</receiver>
```

Dengan menggunakan Intent Filter, kita bisa memberitahu sistem semua Intent yang cocok untuk subelement yang kita buat agar dikirimkan ke broadcast receiver tertentu.

Badan Program

```
IntentFilter filter = new IntentFilter();
filter.addAction(getPackageName()+"android.net.conn.CONNECTIVITY_CHANGE");

ActivityReceiver myReceiver = new ActivityReceiver();
registerReceiver(myReceiver, filter);
```

Contoh:

Kita akan mencoba membuat Broadcast Receiver untuk mengecek pergantian koneksi internet di perangkat mobile. Pertama kita perlu membuat izin network akses, tambahkan baris program berikut di file manifest:

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Lakukan modifikasi pada file **activity_main.xml** dengan baris program berikut:



```
<?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:gravity="center"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btnExplicitBroadcast"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Explicit Broadcast" />
</LinearLayout>
```

Pada **MainActivity.java**, lakukan perubahan kodingan berikut:

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btnExplicitBroadcast;
    // Membuat variabel beritipe ConnectionReceiver
    ConnectionReceiver cr;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnExplicitBroadcast = findViewById(R.id.btnExplicitBroadcast);
        btnExplicitBroadcast.setOnClickListener(this);
        // Membuat variabel or menjadi variabel objek
        cr= new ConnectionReceiver();
    }

    @Override
    protected void onResume() {
        super.onResume();
        // Membuat objek Intent
        IntentFilter filter = new IntentFilter();
        // Menentukan action Intent Filter
        filter.addAction("android.net.conn.CONNECTIVITY_CHANGE");
        // Mendaftarkan Broadcast Receiver
        registerReceiver(cr, filter);
    }

    public void broadcastIntent() {
        Intent intent = new Intent();
        intent.setAction("IntentA");
        // Mengatur komponen nilai untuk Broadcast Receiver
        intent.setComponent(new ComponentName(getPackageName(),
            ds: "com.example.applicationbroadcastreceiver.ConnectionReceiver"));
        getApplicationContext().sendBroadcast(intent);
    }

    // Override fungsi onStop
    @Override
    protected void onStop() {
        super.onStop();
        unregisterReceiver(cr);
    }

    // Override fungsi onClick
    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btnExplicitBroadcast:
                broadcastIntent();
                break;
        }
    }
}

```

Buat sebuah kelas java baru dengan nama **ConnectionReceiver.java**, lalu isikan baris program berikut:

```
public class ConnectionReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // Mengambil nilai aksi dari Intent Filter
        String action = intent.getAction();
        // Melakukan pengecekan jenis Intent
        // Jika Intent merupakan Explicit
        if (action.equals("IntentA")) {
            Toast.makeText(context, text: "Explicit: ", Toast.LENGTH_SHORT).show();
        }
        // Jika Intent merupakan Implicit
        if (("android.net.conn.CONNECTIVITY_CHANGE").equals(action)) {
            Toast.makeText(context, text: "Implicit: Connection Changed",
                Toast.LENGTH_SHORT).show();
        }
    }
}
```

Bagian `context.getSystemService(Context.CONNECTIVITY_SERVICE)` digunakan untuk mengembalikan objek dari kelas `ConnectivityManager` yang digunakan untuk mengakses properti jaringan perangkat.

Fungsi `activeNetwork.getType()` mengembalikan tipe jaringan (mobile data atau WiFi data) yang digunakan.

Registrasikan Broadcast Receiver yang telah dibuat di file manifest.

```
<receiver android:name=".ConnectionReceiver"
    android:enabled="true">
    <intent-filter>
        <action android:name="IntentA" />
    </intent-filter>
</receiver>
```

Jalankan program!

