Praktikum ke : 13 dan 14

Judul Praktikum : Service

Alokasi Waktu : 2 x 50 menit

1. Tujuan Instruksional Khusus

a. Mahasiswa dapat memahami konsep dasar dari service pada android
b. Mahasiswa dapat mengimpementasikan service pada pemrograman berbasis android

2. Teori

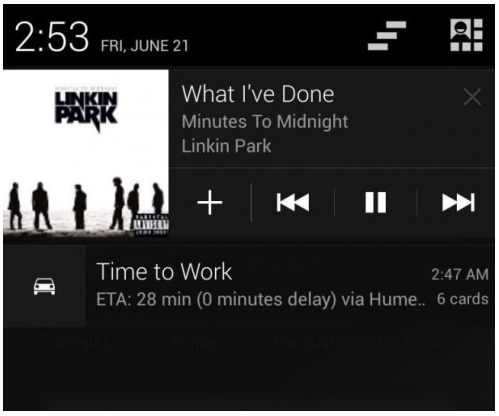
Kita telah belajar mengenai activity dan implementasinya. Activity dan fragment adalah dua komponen yang memberikan pengalaman kepada pengguna secara langsung. Pengguna dapat melihat dan berinteraksi diatasnya.

Service berada pada sisi yang lain, komponen ini tidak memiliki antarmuka dan bahkan pengguna tidak akan akan tahu bagaimana dia bekerja. Pengalaman yang diberikan oleh service hanya berupa proses yang tidak terlihat. Ia digunakan untuk menjalankan beragam macam proses yang memakan waktu lama.

Walaupun berjalan secara *background*, pada dasarnya service dan komponen Android lainnya berjalan pada satu proses dan *thread* yang sama yaitu *main thread* atau *ui thread*. Bekerja di *background* bukan berarti ia bekerja secara *asynchronous.* Service tetap membutuhkan *thread* terpisah jika kita ingin melakukan proses yang membutuhkan komputasi intensif atau yang memakan waktu.

Contoh pemanfaatan service sudah banyak sekali, antara lain:

- Aplikasi sosial media atau aplikasi yang memiliki kemampuan untuk menerima *push notification*. Aplikasi semacam ini pasti memiliki sebuah service yang berjalan dalam posisi *stand by* untuk selalu menerima pesan yang masuk.
- Aplikasi *chat* juga membutuhkan service untuk melakukan pengiriman dan menerima pesan yang dikirimkan oleh pengguna.
- Aplikasi pemutar musik juga melakukan hal yang sama. Untuk memberikan pengalaman yang lebih baik kepada pengguna, aplikasi pemutar musik biasanya meletakkan proses *streaming* atau memainkan musik di komponen service dengan tetap mempertahankan integrasi dengan komponen lain, misalnya notifikasi. Seperti gambar berikut :



Secara umum, terdapat dua bentuk dari service :

1. **Started**

   Service berjenis ini adalah tipe yang dijalankan oleh komponen lain, misal activity. Sekali dijalankan, service ini akan berjalan selama belum dimatikan atau proses yang dijalankan selesai. Sevice akan tetap berjalan walaupun komponen yang lain dimatikan oleh sistem Android. Umumnya penggunaan service ini adalah untuk melakukan proses yang tidak memberikan nilai balik ke komponen yang memanggilnya. Contohnya adalah, mengunduh atau mengunggah berkas.

2. **Bound**

   Service jenis ini merupakan tipe service yang dijalankan oleh komponen lain, namun saling mengikat. Hubungan yang terjadi antar kedua komponen tersebut seperti *client-server*. Bisa saling menerima hasil dan menerima *request*yang ada. Pada service ini dimungkinkan terjadi proses IPC (*Interprocess Communication*).

   Service ini akan tetap berjalan di background selama masih ada komponen lain yang mengikatnya. Jika tidak, maka Service akan dimatikan oleh sistem. Aplikasi pemutar musik merupakan salah satu jenis aplikasi yang mengimplementasikan service jenis ini.

Pada bagian ini kita akan sepenuhnya membahas service berjenis started. Tipe service tersebut akan dibagi menjadi dua bagian dalam implementasinya:

- **Kelas service yang inherit langsung kepada kelas Service**

  Ketika sebuah kelas java *inherit* ke service ingin menjalankan proses yang memakan waktu lama, maka kelas tersebut diharuskan membuat thread terpisah agar tidak memblok ui thread yang ada. Service ini akan selalu hidup di *background* selama tidak ada komponen yang memanggil stopService() atau dimatikan oleh sistem.

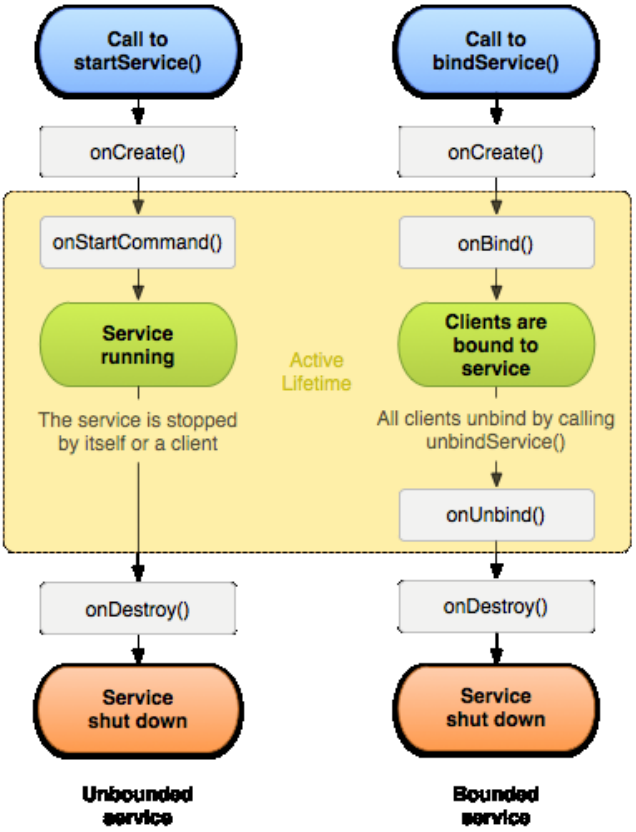
- **Kelas service yang inherit ke IntentService**

  Ini adalah kelas yang sangat memudahkan hidup kita. Dia bersifat *fire and forget*, ketika ia telah menyelesaikan tugasnya, ia akan mematikan dirinya.

Poin-poin penting lain tentang service diantaranya:

- Setiap kelas Java dinyatakan sebuah service bila kelas tersebut inherit/extends ke kelas Service atau IntentService.

- Service memiliki *life cycle*-nya sendiri dan bergantung pada tipe service apa yang digunakan, started atau bound service.



- Untuk menjalankan service dari komponen lain seperti activity, cukup menggunakan `startService(Intent)`. Sebaliknya untuk mematikan/stop service terdapat dua cara. Pertama `stopService(Intent)` dijalankan dari komponen yang memanggil dan `stopSelf()` dari kelas Service itu sendiri.

Untuk mendalami topik Service lebih lanjut, Anda dapat membaca tautan berikut:

- https://developer.android.com/reference/android/app/Service
- https://developer.android.com/guide/components/services
- https://developer.android.com/guide/topics/manifest/service-element#exported

3. Alat dan Bahan

a. BKPM
b. Komputer
c. LCD
d. Alat Tulis Kantor (ATK)

4. Pelaksanaan Praktikum

Anda sudah paham service secara garis besar berikut pemanfaatannya. Sekarang saatnya kita menerapkannya.

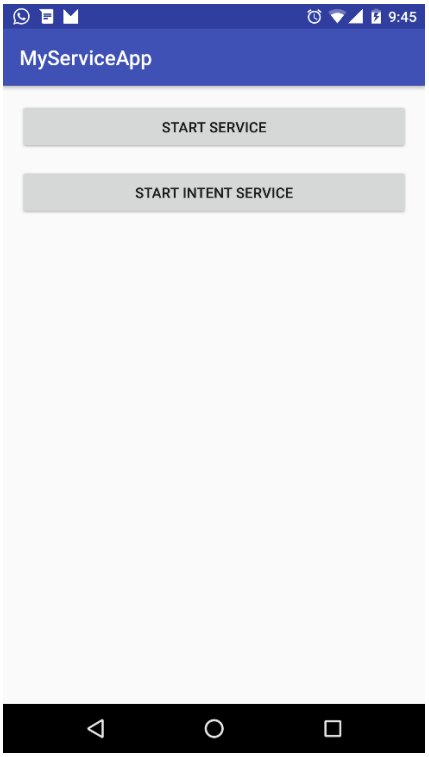
1. Baik, buat proyek baru dengan nama MyService. **Pilih Empty Activity** dengan pilihan *default* pada *set up* proyek. Setelah proyek tercipta, lengkapi activity_main.xml dengan contoh seperti ini:

```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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context="com.dicoding.myserviceapp.MainActivity">
    <Button
        android:id="@+id/btn_start_service"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start Service"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <Button
        android:id="@+id/btn_start_intent_service"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start Intent Service"/>
</LinearLayout>
```

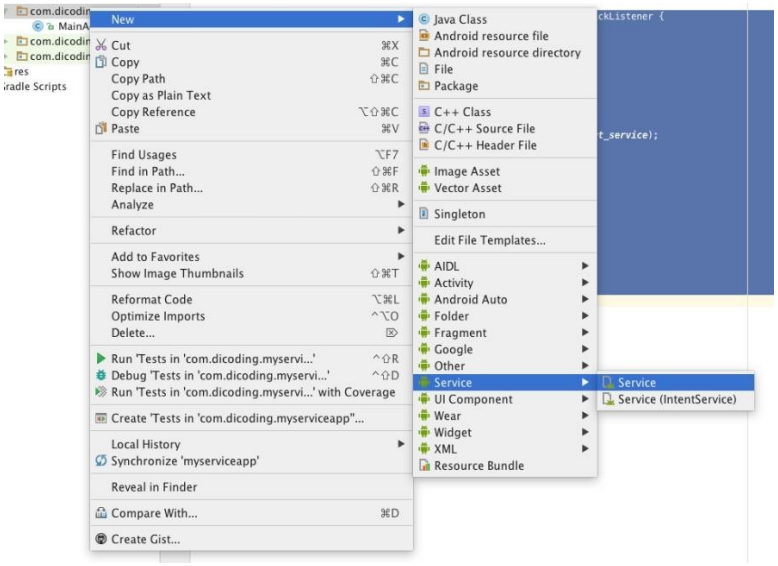
2. Pada MainActivity.java silakan lengkapi kode-nya menjadi sebagai berikut:

```java
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnStartService;
    private Button btnStartIntentService;

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

        btnStartService = (Button)findViewById(R.id.btn_start_service);
        btnStartService.setOnClickListener(this);
        btnStartIntentService = (Button)findViewById(R.id.btn_start_intent_service);
        btnStartIntentService.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.btn_start_service:
                break;
            case R.id.btn_start_intent_service:
                break;
        }
    }
}
```

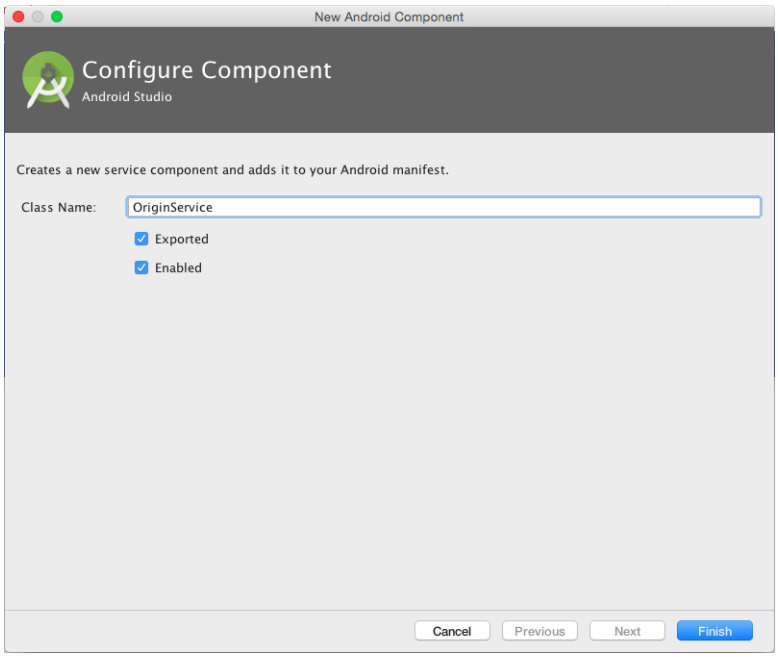
3. Tampilan yang seharusnya ada pada MainActivity, adalah seperti ini.



4. Lanjut, buat kelas service bernama OriginService dengan cara **klik kanan** pada **package** > **project** > **New** > **Service** > **Service.** OriginService akan *inherit* (*extends*) kepada kelas Service.



5. Selanjutnya pada dialog yang tampil, isikan nama kelas service yang diinginkan. Di sini kita menamainya sebagai OriginService dan biarkan exported dan enabled tercentang. Klik Finish untuk menyelesaikan proses.



6. Selanjutnya, buka berkas AndroidManifest.xml pada package manifest dan perhatikan isi berkas tersebut. Service yang baru saja kita buat sudah ada didalam tag <application> :

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.dicoding.myserviceapp">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service
            android:name=".OriginService"
            android:enabled="true"
            android:exported="true">
        </service>
    </application>
</manifest>
```

7. Berkas AndroidManifest sudah dibuat secara otomatis. Dengan demikian kita sudah bisa menjalankan kelas service tersebut. Namun, sebelum menjalankan aplikasi, lengkapi kode pada OriginService menjadi seperti berikut :

```java
public class OriginService extends Service {
    public static final String ORIGIN_SERVICE = "OriginService";

    public OriginService() {

    }
    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Log.d(ORIGIN_SERVICE, "OriginService dijalankan");
        return START_STICKY;
    }
}
```

8. Selanjutnya pada MainActivity.java di metode onClick() pada case R.id.btn_start_service tambahkan baris berikut :

```java
Intent mStartServiceIntent = new Intent(MainActivity.this, OriginService.class);
startService(mStartServiceIntent);
```

9. Sehingga kode pada metode onClick() menjadi seperti ini :

```java
@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.btn_start_service:
            Intent mStartServiceIntent = new Intent(MainActivity.this, OriginService.class);
            startService(mStartServiceIntent);
            break;
        case R.id.btn_start_intent_service:
            break;
    }
}
```

10. Sekarang jalankan aplikasi. Klik tombol ‘start service’ dan perhatikan pada log-nya. OriginService telah dijalankan dan tidak akan pernah mati sampai dimatikan oleh sistem atau metode stopSelf() atau stopService() dijalankan.

11. Baik, sekarang kita akan menambahkan sebuah inner class AsyncTask. Ia seakan-akan menjalankan sebuah proses secara *asynchronous* dan mematikan/menghentikan dirinya sendiri dengan memanggil metode stopSelf(). Lengkapi kodenya menjadi sebagai berikut:

```
1. public class OriginService extends Service {
2.
3.     public static final String ORIGIN_SERVICE = "OriginService";
4.
5.     public OriginService() {
6.
7.     }
8.
9.     @Override
10.    public IBinder onBind(Intent intent) {
11.        // TODO: Return the communication channel to the service.
12.        throw new UnsupportedOperationException("Not yet implemented");
13.    }
14.
15.    @Override
16.    public int onStartCommand(Intent intent, int flags, int startId) {
17.        Log.d(ORIGIN_SERVICE, "OriginService dijalankan");
18.        ProcessAsync mProcessAsync = new ProcessAsync();
19.        mProcessAsync.execute();
20.        return START_STICKY;
21.    }
22.
23.    private class ProcessAsync extends AsyncTask<Void, Void, Void>{
24.        @Override
25.        protected Void doInBackground(Void... params) {
26.            try {
27.                Thread.sleep(3000);
28.            } catch (InterruptedException e) {
29.                e.printStackTrace();
30.            }
31.            return null;
32.        }
33.
34.        @Override
35.        protected void onPostExecute(Void aVoid) {
36.            super.onPostExecute(aVoid);
37.            Log.d(ORIGIN_SERVICE, "StopService");
38.            stopSelf();
39.        }
40.    }
41.
42.    @Override
43.    public void onDestroy() {
44.        super.onDestroy();
45.        Log.d(ORIGIN_SERVICE, "onDestroy()");
46.    }
47. }
```

12. Jalankan aplikasinya. Klik tombol ‘start service’ dan perhatikan *log*-nya. Service dijalankan secara *asynchronous* dan mematikan dirinya sendiri setelah proses selesai.

13. Jika berhasil dijalankan, pada log androiod monitor akan seperti ini :

    09-22 09:52:25.028 10209-10209/com.polije.myserviceapp D/OriginService: OriginService dijalankan

    09-22 09:52:28.074 10209-10209/com.polije.myserviceapp D/OriginService: StopService

    09-22 09:52:28.078 10209-10209/com.polije.myserviceapp D/OriginService: onDestroy()