

Book Series
of
jago{coding}
Learn & Share {Your Code}

www.jagocoding.com

Membuat Interface Dinamis Menggunakan FrameLayout

Oleh: Hasyemi Rafsanjani Asyari

Pada tutorial kali ini kita akan belajar membuat aplikasi android dengan fragment. Seperti yang kita ketahui aplikasi android terdiri dari sekumpulan activity. Nah pada activity tersebut kita bisa menyisipkan semacam sub-activity yang kita sebut sebagai fragment. Fragment sangat berguna karena dapat memb...

Pada tutorial kali ini kita akan belajar membuat aplikasi android dengan fragment. Seperti yang kita ketahui aplikasi android terdiri dari sekumpulan activity. Nah pada activity tersebut kita bisa menyisipkan semacam sub-activity yang kita sebut sebagai fragment. Fragment sangat berguna karena dapat membuat user-interface dari aplikasi yang kita buat menjadi dinamis. Untuk mempersingkat waktu, berikut hal yang dibutuhkan untuk membuat aplikasi ini.

- IDE (Android Studio)
- FragmentActivity
- FrameLayout
- FragmentManager
- FragmentTransaction

Buat project baru di Android studio dengan minSDKversion version 11



Pilih BlankActivity dan beri nama MainActivity, nama boleh diganti namun saat mengikuti tutorial harus mampu menyesuaikan.

Pada saat awal project dibuat kita hanya mempunyai satu activity dan layout. Sekarang buka layout tersebut yang bernama activity_main.xml (nama layout bisa berbeda tergantung nama activity) lalu ubah codenya menjadi seperti berikut.

activity_main.xml

```
<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:padding="20dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Fragment A"
        android:onClick="fragmentA"/>

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Fragment B"
    android:onClick="fragmentB"/>

<FrameLayout
    android:id="@+id/frame"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>

</LinearLayout>

```

Pada layout ini kita menaruh FrameLayout yang mana akan menjadi wadah atau container dari fragment yang akan kita buat. Juga terdapat 2 button untuk merubah fragment, button Fragment A dan button Fragment B.

Setelah itu buka **MainActivity.java** lalu ubah class tersebut menjadi FragmentActivity dengan cara mengubah extends menjadi **FragmentActivity**. Jangan lupa untuk mengimport class nya.

*Tips : gunakan alt + enter.

Sekarang kita mulai membuat fragment. Kita akan dua buah fragment dengan tampilan layout yang berbeda. Dengan Android Studio buat fragment melalui File -> New -> Fragment

Buat dua fragment dengan nama FragmentA dan FragmentB. Pada layout fragment A kita akan berikan widget jam analog dan pada fragment B widget jam Digital sehingga terlihat perbedaannya.

fragment_a.xml

```

<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:padding="20dp"
    android:orientation="vertical"
    tools:context="android.fragmentexample.FragmentA">

    <!-- TODO: Update blank fragment layout -->
    <Button
        android:text="Fragment A"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

```

```
<AnalogClock
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fragment A dengan jam analog"/>
</LinearLayout>
```

fragment_b.xml

```
<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:padding="20dp"
    android:orientation="vertical"
    tools:context="android.fragmentexample.FragmentA">

    <!-- TODO: Update blank fragment layout -->
    <Button
        android:text="Fragment B"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <DigitalClock
        android:layout_gravity="center"
        android:textSize="80dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment B dengan jam digital"/>

</LinearLayout>
```

Lalu untuk class Fragment kita ubah code nya menjadi seperti berikut

FragmentA.java

```
package android.fragmentexample;

import android.app.Activity;
```

```

import android.net.Uri;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class FragmentA extends Fragment
{
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState)
    {
        View view = inflater.inflate(R.layout.fragment_a, container,
false);
        return view;
    }
}

```

FragmentB.java

```

package android.fragmentexample;

import android.app.Activity;
import android.net.Uri;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class FragmentB extends Fragment {
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState)
    {
        View view = inflater.inflate(R.layout.fragment_b, container,
false);
        return view;
    }
}

```

Terakhir kita bisa mengendalikan kedua fragment tersebut melalui MainActivity. MainActivity disini bertindak sebagai container dari Fragment yang kita buat sehingga MainActivity harus extends FragmentActivity.

Untuk bisa mengendalikan fragment maka dibutuhkan FragmentManager. FragmentTransaction digunakan untuk melakukan transaksi fragment seperti tambah fragment, ganti fragment, hapus fragment. Untuk sekarang kita hanya akan menggunakan add dan replace fragment saja.

MainActivity.java

```
package android.fragmentexample;

import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.support.v4.app.FragmentActivity;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

public class MainActivity extends FragmentActivity {

    FragmentManager fragmentManager;
    FragmentTransaction transaction;

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

        fragmentManager = getFragmentManager();
        transaction = fragmentManager.beginTransaction();

        FragmentA a = new FragmentA();
        transaction.add(R.id.frame, a);
        transaction.commit();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    public void fragmentA(View view)
    {
```

```
        fragmentManager = getFragmentManager();
        transaction = fragmentManager.beginTransaction();

        FragmentA a = new FragmentA();
        transaction.replace(R.id.frame, a);
        transaction.commit();
    }

    public void fragmentB(View view)
    {
        fragmentManager = getFragmentManager();
        transaction = fragmentManager.beginTransaction();

        FragmentB b = new FragmentB();
        transaction.replace(R.id.frame, b);
        transaction.commit();
    }
}
```

Jika user menekan button yang ada maka akan ditampilkan fragment yang sudah kita buat.



Sekian tutorial kali ini, semoga bermanfaat dan terima kasih :)

Tentang Penulis



Hasyemi Rafsanjani Asyari

null. I'm proud to be Stalker. Currently Mahasiswa Politeknik Negeri Jakarta. Currently Microsoft Student Partners Indonesia Regional Jabodetabek. Founder of a Company.