

MOBILE PROGRAMMING



Oleh :

CHALIFA CHAZAR



MATERI 9

THREAD

Tujuan:

Mahasiswa memahami dan mampu menerapkan penggunaan thread dalam pembuatan suatu aplikasi Android.

Pustaka:

Abdul Kadir, 2013: Pemograman Aplikasi Android. Andi Yogyakarta.

Ivan Michael Siregar, 2011: Membongkar Source Code Berbagai Aplikasi Android. Gava Media.

[Http://www.gits.co.id](http://www.gits.co.id)

1. Thread

Salah satu hal yang penting dalam pembuatan aplikasi berbasis mobile adalah penggunaan thread. Thread adalah suatu struktur pengendali program yang memungkinkan program berjalan secara background atau multitasking (dapat menjalankan beberapa tugas secara bersamaan), dan cara kerjanya seperti listener.

Sebagai contoh, pada sebuah aplikasi Messenger. Paling tidak terdapat dua buah thread, yaitu thread untuk menerima pesan dan thread untuk menerima pesan. Penggunaan thread sangat baik untuk menghindari adanya deadlock atau freeze pada suatu layanan.

2. Membuat Thread

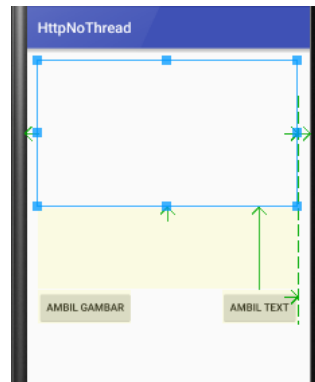
Pada proyek sebelumnya Anda membuat sebuah program yang dapat mengambil gambar dan text dari internet. Akan tetapi pada sebuah tombol hanya dapat diklik ketika fungsinya sudah selesai dilakukan. Apabila kedua tombol langsung diklik, maka program akan mengalami "freeze" atau "not responding".

Pada proyek berikut ini Anda akan memperbaiki program tersebut dengan menggunakan **thread** sehingga kedua tombol dapat diklik bersamaan, dan dapat menampilkan fungsi



menampilkan gambar dan text, dan program tidak "freeze" atau "not responding". Untuk membuat program tersebut, berikut ini adalah langkah-langkah yang perlu dilakukan.

1. Buka project pada modul sebelumnya. Atau Anda dapat membuat sebuah project Android baru.
2. Berikut ini adalah tampilan project yang telah dibuat sebelumnya. Terdiri dari beberapa komponen, yaitu 1 buah TextView, 2 buah Button, dan 1 buah ImageView.



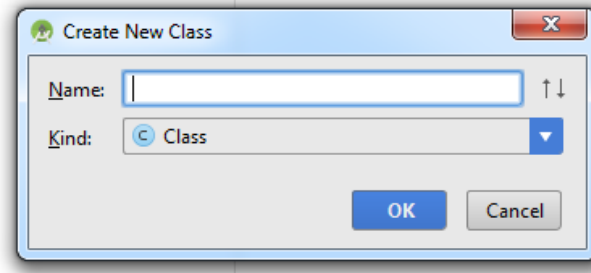
3. Pastikan nilai **ID** dan **Text** komponen-komponen yang digunakan telah sesuai seperti berikut ini.

Komponen	Properti	Nilai Baru
imageView	ID	imageView
Button	ID	ButtonGambar
	Text	Ambil Gambar
Button2	ID	ButtonText
	Text	Ambil Text
textView	ID	textView1

4. Untuk memastikan tidak ada perubahan simpan kembali project atau tekan Ctrl+S untuk menyimpan project.

Selanjutnya Anda akan membuat sebuah **thread** yang berfungsi untuk men-download gambar dan menampilkan gambar kedalam `ImageView`. Lakukan langkah-langkah berikut ini.

1. Buatlah sebuah file java baru
2. Beri nama file tersebut (biasanya diberi nama sesuai dengan fungsi yang akan dilakukan).



3. Buka file tersebut (dalam modul ini file diberi nama ThreadGambar.java) dan lakukan perubahan sehingga tampilan source menjadi seperti berikut ini.

```
package com.example.fae.latihan10thread;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.util.Log;
import android.widget.ImageView;

import java.io.InputStream;

/**
 * Created by FAE on 10/13/2015.
 */
public class ThreadGambar extends AsyncTask <String, Void, Bitmap>{
    ImageView bmImage;

    public ThreadGambar (ImageView bmImage){
        this.bmImage = bmImage;
    }

    protected Bitmap doInBackground(String... urls) {
        String urldisplay = urls[0];
        Bitmap mIcon11 = null;
        try {
            InputStream in = new
java.net.URL(urldisplay).openStream();
            mIcon11 = BitmapFactory.decodeStream(in);
        } catch (Exception e){
            Log.e("Error", e.getMessage());
            e.printStackTrace();
        }
        return mIcon11;
    }
    protected void onPostExecute(Bitmap result) {
        bmImage.setImageBitmap(result);
    }
}
```

4. Buka file MainActivity.java, lakukan perubahan untuk membuat fungsi men-download gambar dan menampilkan gambar ke dalam ImageView. Berikut ini bagian source yang ditambahkan.



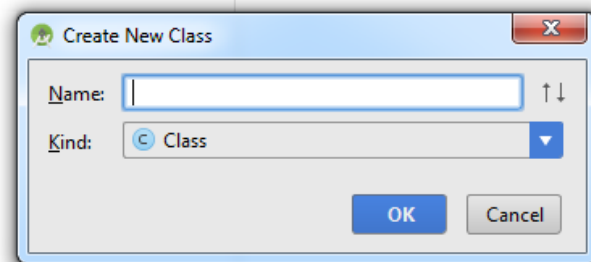
```
Button btnimage = (Button) findViewById(R.id.btnimage);

btnimage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ImageView Img = (ImageView)
        findViewById(R.id.imageView);
        new
        ThreadGambar(Img).execute("http://edu.script.id/images/LOGO-EDU-
        Header.png");

        KlikGambar++;
        TextView txtKlikGambar = (TextView)
        findViewById(R.id.textView2);
        txtKlikGambar.setText("Button Gambar diklik sebanyak "
        + String.valueOf(KlikGambar) + "");
    }
});
```

Selanjutnya Anda akan membuat sebuah **thread** yang berfungsi untuk men-download text dan menampilkan text kedalam TextView. Lakukan langkah-langkah berikut ini.

1. Buatlah sebuah file java baru
2. Beri nama file tersebut (biasanya diberi nama sesuai dengan fungsi yang akan dilakukan).



3. Buka file tersebut (dalam modul ini file diberi nama ThreadText.java) dan lakukan perubahan sehingga tampilan source menjadi seperti berikut ini.

```
package com.example.fae.latihan10thread;

import android.graphics.Bitmap;
import android.os.AsyncTask;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
```



```
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

/**
 * Created by FAE on 10/13/2015.
 */
public class ThreadText extends AsyncTask<String, Void, String> {
    TextView bmText;

    public ThreadText (TextView bmText){
        this.bmText = bmText;
    }

    protected String doInBackground(String... urls) {
        String urldisplay = urls[0];
        String Hasil = "";
        URL url;
        HttpURLConnection connection = null;
        try {
            url = new URL(urldisplay);
            connection = (HttpURLConnection)url.openConnection();
            connection.setDoOutput(true);
            DataOutputStream wr = new DataOutputStream
(connection.getOutputStream());
            InputStream is = connection.getInputStream();
            BufferedReader rd = new BufferedReader(new
InputStreamReader(is));
            String line;
            StringBuffer response = new StringBuffer();
            while ((line = rd.readLine()) !=null) {
                response.append(line);
                response.append("\r");
            }
            rd.close();
            Hasil = response.toString();
        }
        catch (FileNotFoundException e){
            e.printStackTrace();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (connection !=null){
                connection.disconnect();
            }
        }
        return Hasil;
    }

    protected void onPostExecute (String result) {
        bmText.setText(result);
    }
}
```

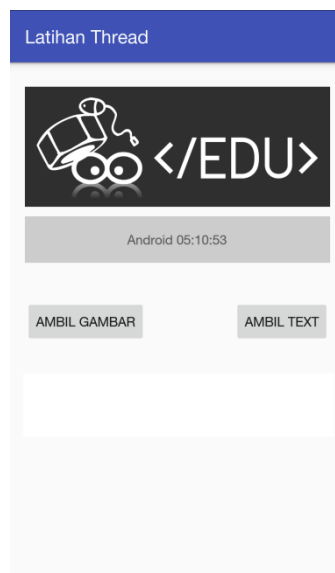


4. Buka file `MainActivity.java`, lakukan perubahan seperti yang telah dilakukan sebelumnya.

Berikut ini adalah tampilan awal program ketika dijalankan. Selanjutnya coba klik tombol "Ambil Gambar" dan "Ambil Text" secara bersamaan.



Pada program sebelumnya, program akan menjadi "freeze" atau "not responding" ketika kedua tombol diklik pada waktu yang hampir bersamaan. Akan tetapi setelah dilakukan perubahan hal tersebut tidak lagi terjadi. Berikut ini tampilan ketika kedua tombol diklik hampir bersamaan.





Untuk dapat melihat perbedaan pada latihan program sebelumnya, buat 2 buah tombol lagi untuk menampilkan gambar dan text dengan alamat yang berbeda.