

Android OpenGL Part 3 - Transformasi

[Pemrograman OpenGL Android 03][Level: Mahir]

Andi Taru Nugroho Nur Wismono S.Kom.,M.Cs.

andi.taru@gmail.com

Lisensi Dokumen:

Copyright ©2012 JavaClopedia.com

Seluruh dokumen di JavaClopedia.com dapat digunakan dan disebarakan secara bebas untuk tujuan non-komersial dan harus menyertakan penulis serta sumber asli dokumen yaitu JavaClopedia.com.

Penulisan ulang tidak diperkenankan tanpa seijin JavaClopedia.com

Persiapan

Sebelum mengikuti tutorial ini, ada baiknya pembaca telah membaca beberapa tutorial sebagai berikut:

- Pemrograman Dasar Android 01 - Instalasi di <http://www.javaclipedia.com/>
- Pemrograman Android OpenGL 01 – Hello Android OpenGL di <http://javaclipedia.com/40/android-opengl-part-1.php>
- Pemrograman Android OpenGL 02 – Polygon di OpenGL <http://javaclipedia.com/40/android-opengl-part-2.php>

Pendahuluan

Pada tutorial sebelumnya, penulis membahas tentang cara pembuatan polygon. Adapun pada tutorial kali ini, penulis akan membahas tentang cara melakukan transformasi yang terdiri dari Translasi, Skala dan Rotasi. Pada tutorial ini dibahas juga tentang LoadIdentity, PopMatrix dan PushMatrix.

Translasi

Translasi artinya kita menggeser posisi dari posisi x,y,z ke posisi x2,y2 dan z2. Translasi dapat dilakukan dengan cara memanggil fungsi:

```
glTranslatef(float x, float y, float z);
```

Misalnya titik awal adalah [0.2, 0.4, 0.6] kemudian dilakukan translasi dengan matriks [0.2, 0.0, 0.0] maka hasilnya titik akhir adalah [0.4, 0.4, 0.6].

Skala

Skala artinya kita mengalikan ukuran titik dari posisi x,y,z dengan nilai x_2,y_2 dan z_2 . Skala dapat digunakan untuk memperbesar atau memperkecil polygon. Dapat dilakukan dengan cara memanggil fungsi:

```
glScalef(float x, float y, float z);
```

Misalnya titik awal adalah $[0.2, 0.4, 0.6]$ kemudian dilakukan skala dengan matriks $[2.0, 2.0, 2.0]$ maka hasilnya titik akhir adalah $[0.4, 0.8, 1.2]$.

Rotasi

Translasi artinya kita menggeser posisi dari posisi x,y,z ke posisi x_2,y_2 dan z_2 . Translasi dapat dilakukan dengan cara memanggil fungsi:

```
glTranslatef(float x, float y, float z);
```

Misalnya titik awal adalah $[0.2, 0.4, 0.6]$ kemudian dilakukan rotasi dengan matriks $[60.0, 0.0, 0.0]$ maka titik-titik akan dilakukan rotasi sebesar 60 derajat berdasarkan sumbu x.

Load Identity

Load identity digunakan untuk memanggil matriks identitas. Gunanya adalah untuk mereset transformasi yang sudah dilakukan. Load identity dapat dipanggil dengan cara:

```
glLoadIdentity();
```

Push Matrix

PushMatrix digunakan untuk menyimpan kondisi transformasi saat itu ke dalam sebuah stack. PushMatrix dapat dipanggil dengan cara:

```
glPushMatrix();
```

Pop Matrix

PopMatrix digunakan untuk mengambil kondisi transformasi dari stack yang paling terakhir di Push (paling atas). PopMatrix dapat dipanggil dengan cara:

```
glPopMatrix();
```



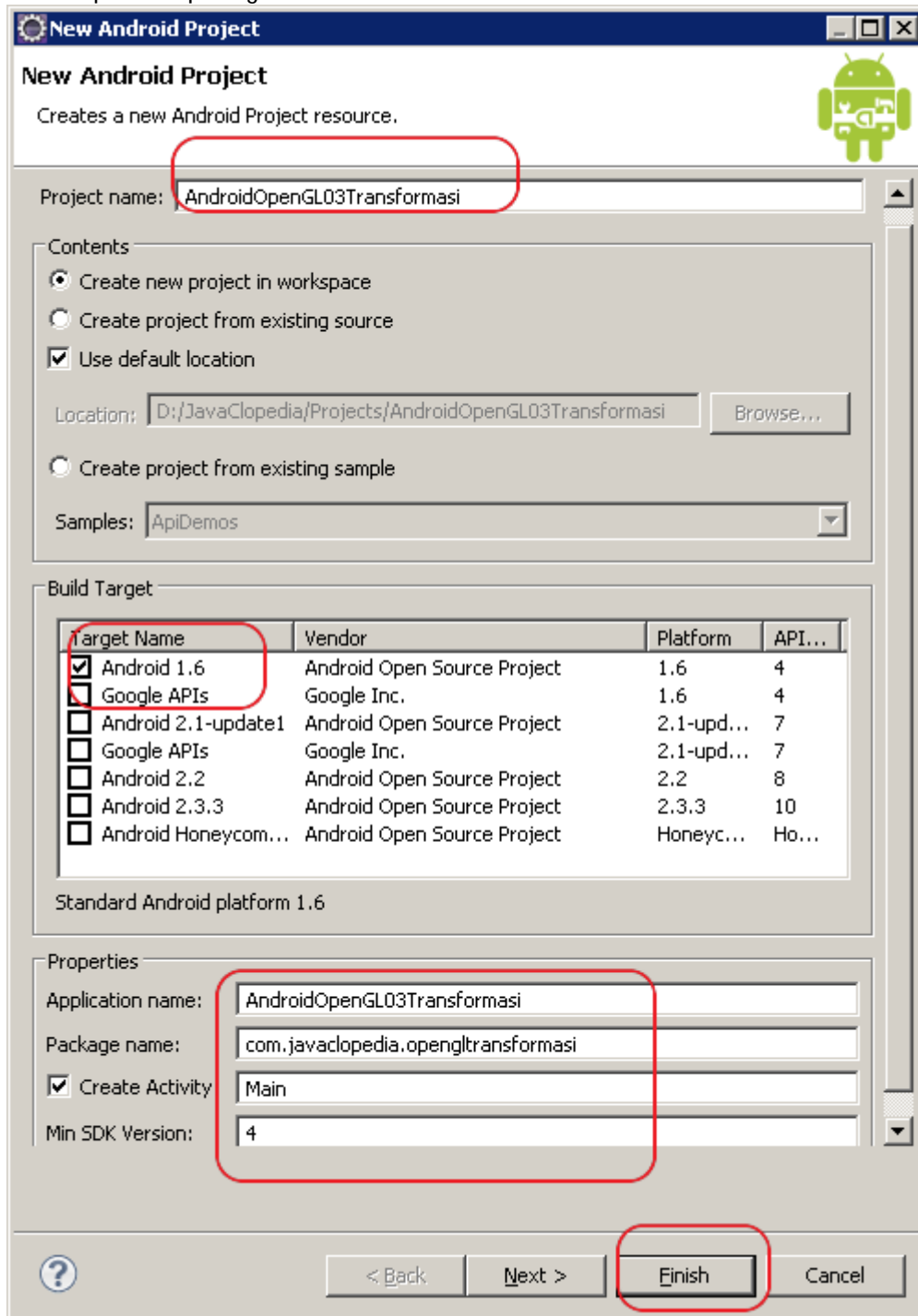
Catatan

Titik-titik di dalam OpenGL dikalkulasikan menggunakan metode matriks. Pengetahuan penjumlahan, perkalian matriks akan sangat membantu memahami konsep transformasi.

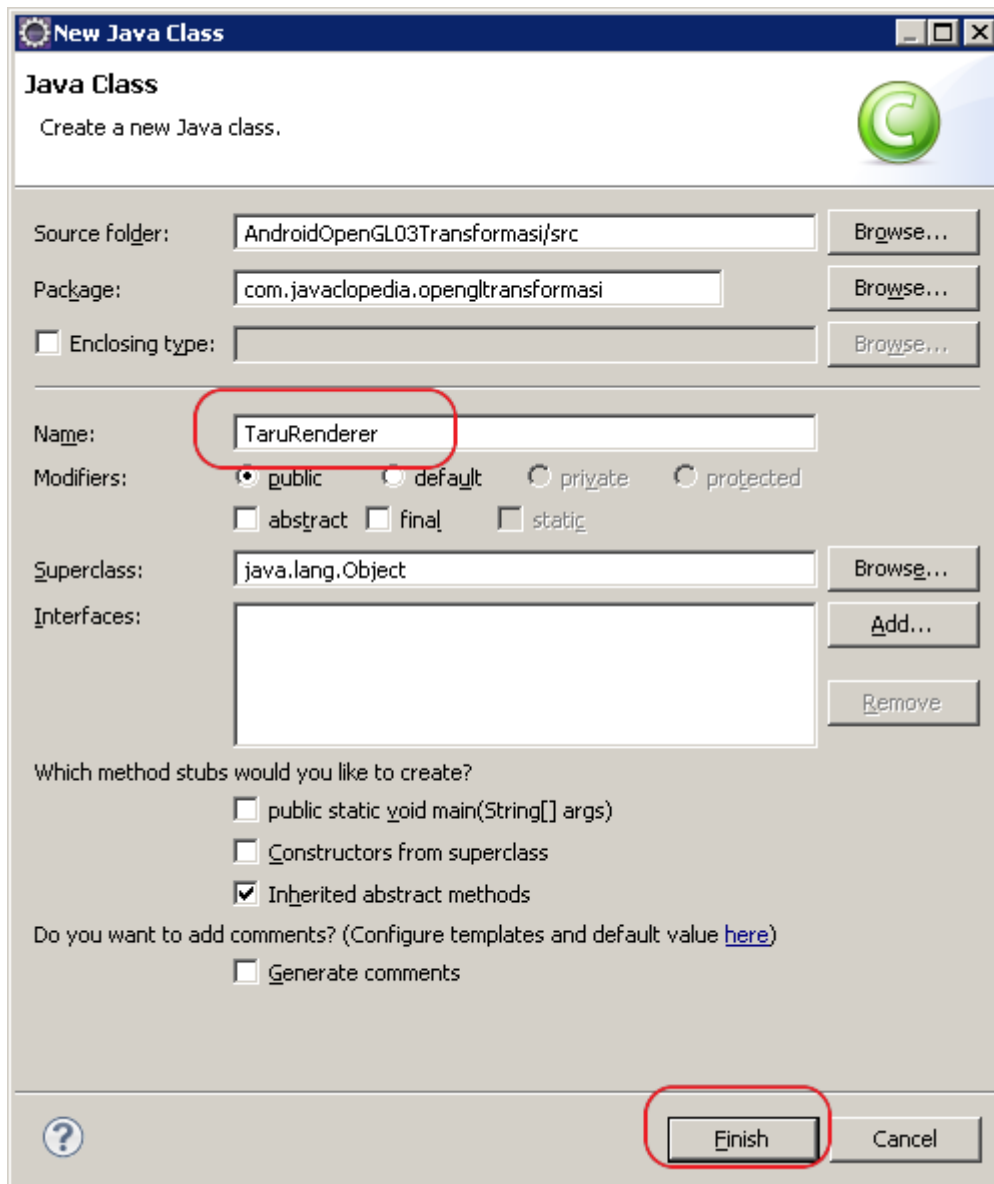
ANDROID IN ACTION!

Tidak perlu berlama-lama, mari kita belajar langsung praktik:

1. Buka Editor Eclipse
2. Buat project baru dengan cara *File > New > Other > Android > Android Project > Next*.
3. Isikan inputan seperti gambar di bawah ini:



4. Tekan Finish
5. Expand Project AndroidOpenGL03Transformasi masuk ke bagian src kemudian klik kanan package *com.javaclopedia.opengltransformasi > New > Class*.
6. Isikan inputan seperti gambar di bawah kemudian tekan Finish.



7. Akan muncul source code baru seperti gambar di bawah:

```
TaruRenderer.java x
package com.javaclopedia.opengltransformasi;

public class TaruRenderer {
    |
}
```

8. Lengkapi kode program menjadi seperti berikut ini:

```
package com.javaclopedia.opengltransformasi;

import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.FloatBuffer;
import import java.nio.ShortBuffer;
```

```
import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;

import android.opengl.GLSurfaceView.Renderer;

public class TaruRenderer implements Renderer {
    private ShortBuffer _indexBuffer;
    private FloatBuffer _vertexBuffer;

    public void onDrawFrame(GL10 gl) {
        gl.glClearColor(1f, 1f, 1f, 1f);
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT);

        gl.glColor4f(0.5f, 0f, 0f, 0.5f);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, _vertexBuffer);

        // gl.glLoadIdentity();
        // gl.glTranslatef(0.01f, 0.0f, 0.0f);
        gl.glRotatef(10, 0.0f, 0.0f, 1.0f);
        // gl.glScalef(1.5f, 1.5f, 1.5f);
        gl.glDrawElements(GL10.GL_TRIANGLES, 3,
            GL10.GL_UNSIGNED_SHORT, _indexBuffer);
        // gl.glRotatef(10, 0.0f, 0.0f, -1.0f);
    }

    public void onSurfaceChanged(GL10 gl, int width, int height) {
        gl.glViewport(0, 0, width, height);
    }

    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
        initTriangle();
    }

    private void initTriangle() {
        int _nrOfVertices = 3;

        ByteBuffer vbb = ByteBuffer.allocateDirect(_nrOfVertices * 3 * 4);
        vbb.order(ByteOrder.nativeOrder());
        _vertexBuffer = vbb.asFloatBuffer();

        ByteBuffer ibb = ByteBuffer.allocateDirect(_nrOfVertices * 2);
        ibb.order(ByteOrder.nativeOrder());
        _indexBuffer = ibb.asShortBuffer();

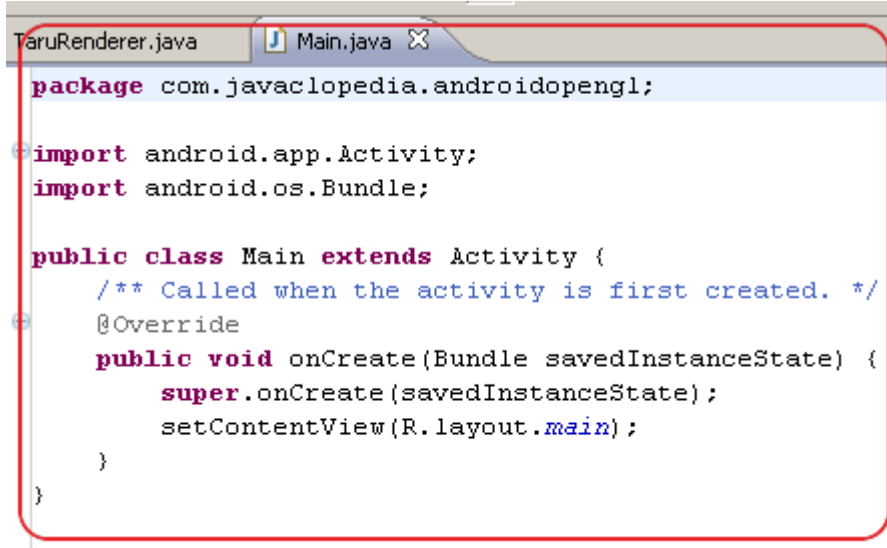
        float[] coords = {
            -0.5f, -0.5f, 0f, // (x1, y1, z1)
            0.5f, -0.5f, 0f, // (x2, y2, z2)
            0f, 0.5f, 0f // (x3, y3, z3)
        };

        short[] _indicesArray = { 0, 1, 2 };

        _vertexBuffer.put(coords);
        _indexBuffer.put(_indicesArray);

        _vertexBuffer.position(0);
        _indexBuffer.position(0);
    }
}
```

9. Kemudian buka berkas Main.java yang sebelumnya tampilannya seperti ini:



```
package com.javaclopedia.androidopengl;

import android.app.Activity;
import android.os.Bundle;

public class Main extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

10. Lengkapi berkas Main.java menjadi seperti berikut ini:

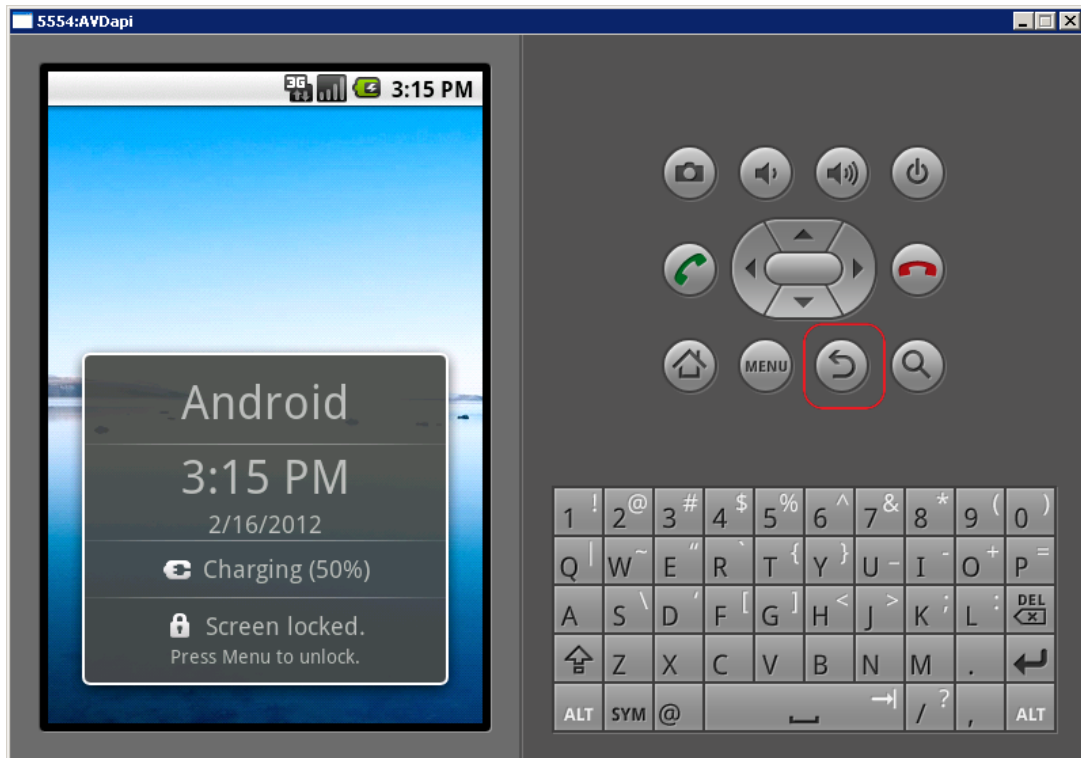
```
package com.javaclopedia.androidopengl;

import android.app.Activity;
import android.opengl.GLSurfaceView;
import android.os.Bundle;

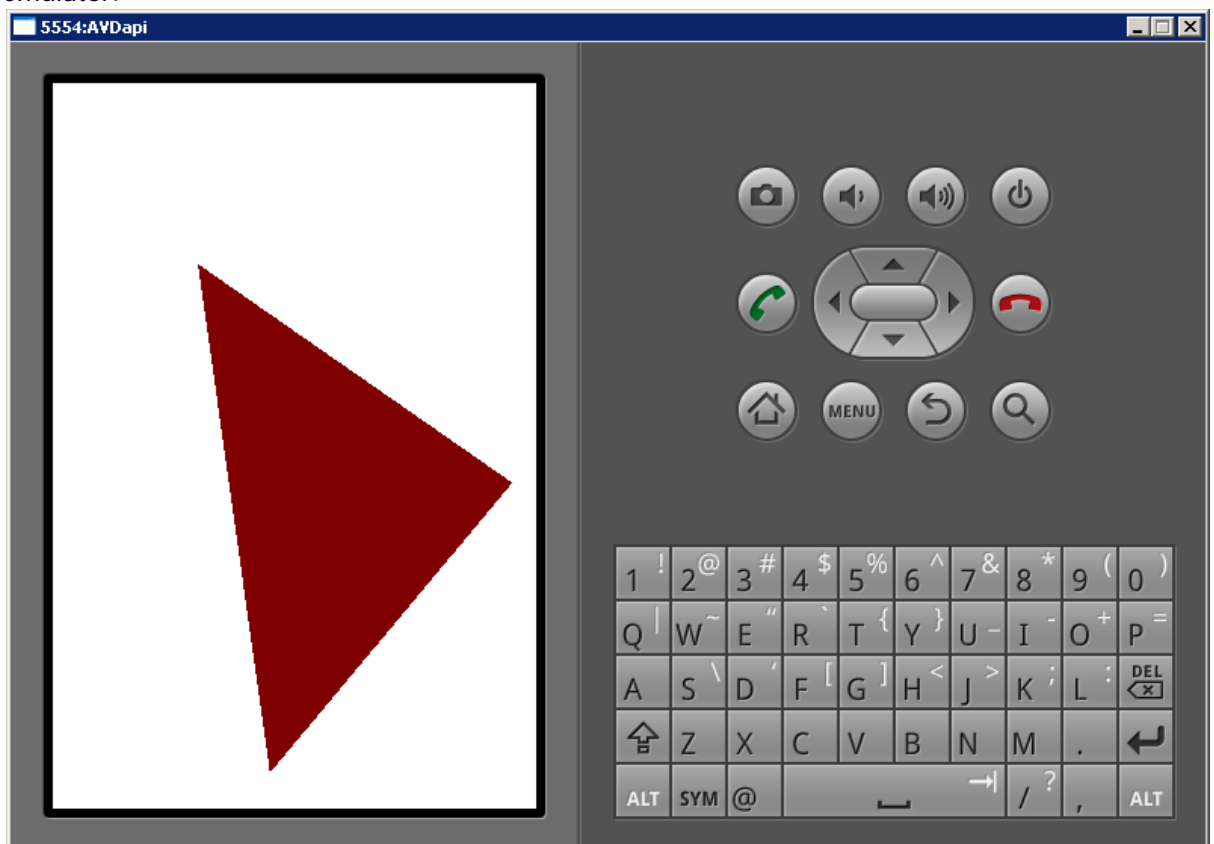
public class Main extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        GLSurfaceView view = new GLSurfaceView(this);
        view.setRenderer(new TaruRenderrer());
        setContentView(view);
    }
}
```

11. Setelah itu, kita jalankan aplikasi kita dengan cara, *Klik Kanan Project AndroidOpenGL03Transformasi > Run As > Android Application.*



12. Jika Emulator sudah menunjukkan seperti Gambar di atas, maka tekan tombol MENU pada emulator.



13. Jika sudah muncul seperti tampilan di atas, berarti Pembaca telah berhasil membuat aplikasi Transformasi OpenGL di Android. **SELAMAT!!!**



Kesimpulan

1. Transformasi di dalam OpenGL terdiri dari translasi, skala dan rotasi.
2. Operasi Transformasi dilakukan dengan menggunakan matriks.
3. LoadIdentity dapat digunakan untuk melakukan reset matriks.
4. PushMatrix digunakan untuk menyimpan kondisi transformasi, sedangkan PopMatrix digunakan untuk mengaktifkan kondisi transformasi yang telah di push sebelumnya.

Biografi Penulis



Andi Taru Nugroho Nur Wismono, Lahir di Tuntang, 01 April 1987. Menyelesaikan S1 Fakultas TI-TI pada tahun 2009 dan menyelesaikan S2 Fakultas TI-SI pada tahun 2011. Penulis merupakan founder dari **JavaClopedia.com** juga Founder dan CEO perusahaan IT **EducaStudio** (educastudio.com). Fokus penulis ada pada pemrograman Java baik itu pemrograman **game**, pemrograman **desktop**, pemrograman **mobile** dan pemrograman **enterprise**. Pengalaman belajar Java penulis, dimulai sejak tahun 2005. Ingin konsultasi pemrograman Java dan Android? request tutorial? Kritik dan Saran? Kirimkan email ke andi.taru@gmail.com