

Android OpenGL Part 4 - Color

[Pemrograman OpenGL Android 04][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>
- Pemrograman Android OpenGL 03 –Transformasi <http://javaclipedia.com/69/tutorial-dasar-android-opengl-part-3-transformasi.php>

Pendahuluan

Pada tutorial sebelumnya, penulis membahas tentang transformasi objek di OpenGL. Adapun pada tutorial kali ini, penulis akan membahas tentang konsep pemberian warna pada OpenGL. Warna di dalam OpenGL hanya terdapat dua macam yaitu Flat Color dan Smooth Color.

Color

Di dalam OpenGL, color menggunakan ukuran float antara 0 sampai 1. Jika pada umumnya menggunakan nilai RGBA sebesar 0-255, maka harus di petakan di dalam OpenGL antara 0 sampai 1. Contoh warna color hitam adalah sebagai berikut: [0.0f, 0.0f, 0.0f, 1.0f]

Flat Color

Adalah pemberian warna secara solid, pada setiap vertex yang ada. Semua warna diberikan warna yang merata dan sama. Flat Color dapat diberikan dengan cara:

```
glColor4f(float red, float green, float blue, float alpha);
```

Smooth color

Adalah pemberian warna secara smooth, yaitu setiap vertex diberikan warna yang berbeda-beda. pemberian warna yang berbeda-beda tersebut akan memberikan efek warna halus yang menarik. Smooth Color dapat diberikan dengan cara:

1) Buat Warna sesuai jumlah vertex di dalam matriks Float.

```
float[] colors = {  
    1f, 0f, 0f, 1f, // point 0 red  
    0f, 1f, 0f, 1f, // point 1 green  
    0f, 0f, 1f, 1f, // point 2 blue  
    1f, 0f, 1f, 1f,  
};
```

2) Buat FloatBuffer untuk menampung color.

```
FloatBuffer vertexBuffer;  
  
ByteBuffer cbb = ByteBuffer.allocateDirect(colors.length * 4);  
cbb.order(ByteOrder.nativeOrder());  
colorBuffer = cbb.asFloatBuffer();  
colorBuffer.put(colors);  
colorBuffer.position(0);
```

3) Implementasikan Color di dalam Draw.

```
gl.glEnableClientState(GL10.GL_COLOR_ARRAY);  
gl.glColorPointer(4, GL10.GL_FLOAT, 0, colorBuffer);  
gl.glDisableClientState(GL10.GL_COLOR_ARRAY);
```



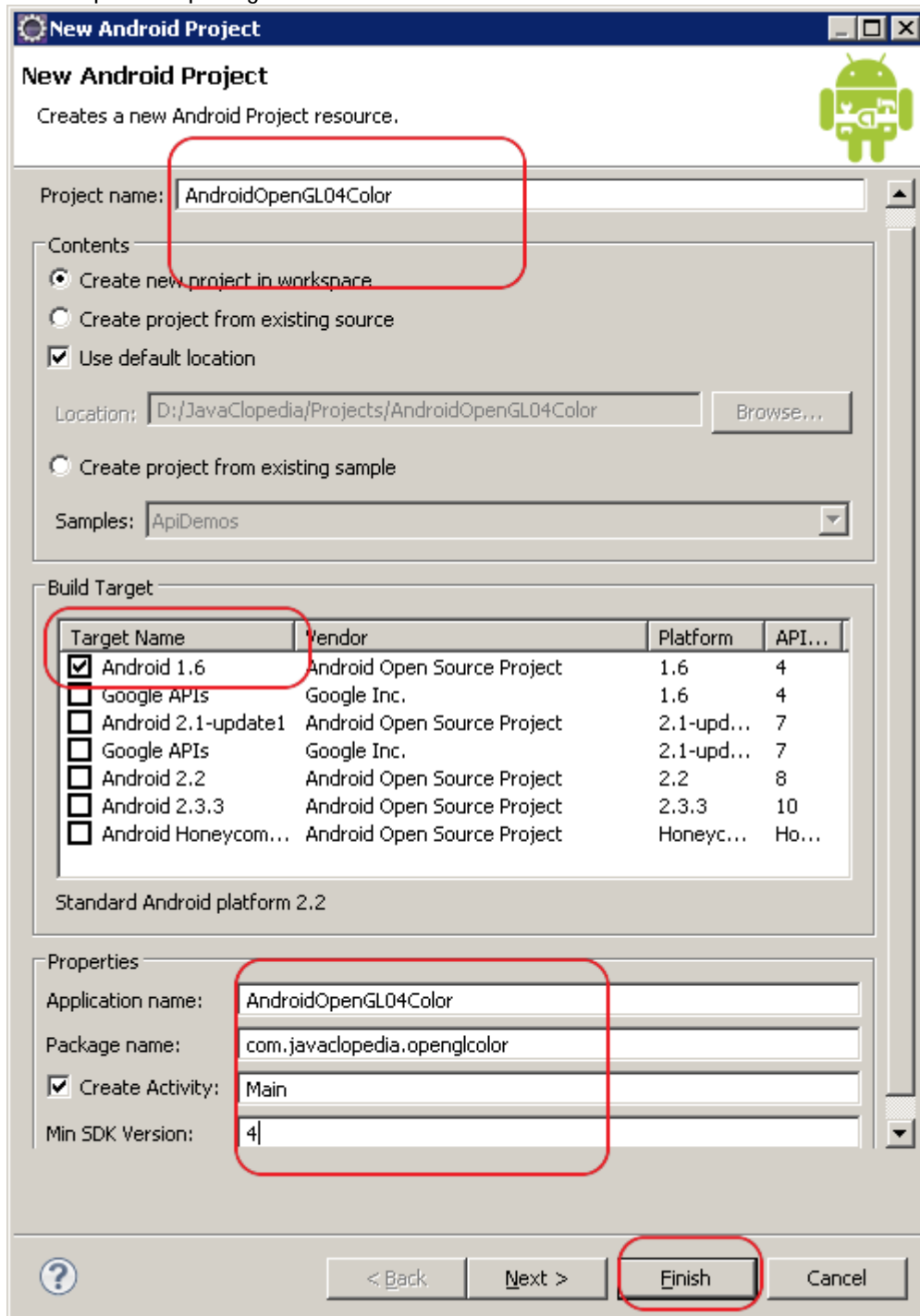
Catatan

Perhatikan bahwa untuk menggunakan Smooth Color kita harus mengaktifkan GL_COLOR_ARRAY. Kemudian men-disabled kembali setelah tidak dipakai.

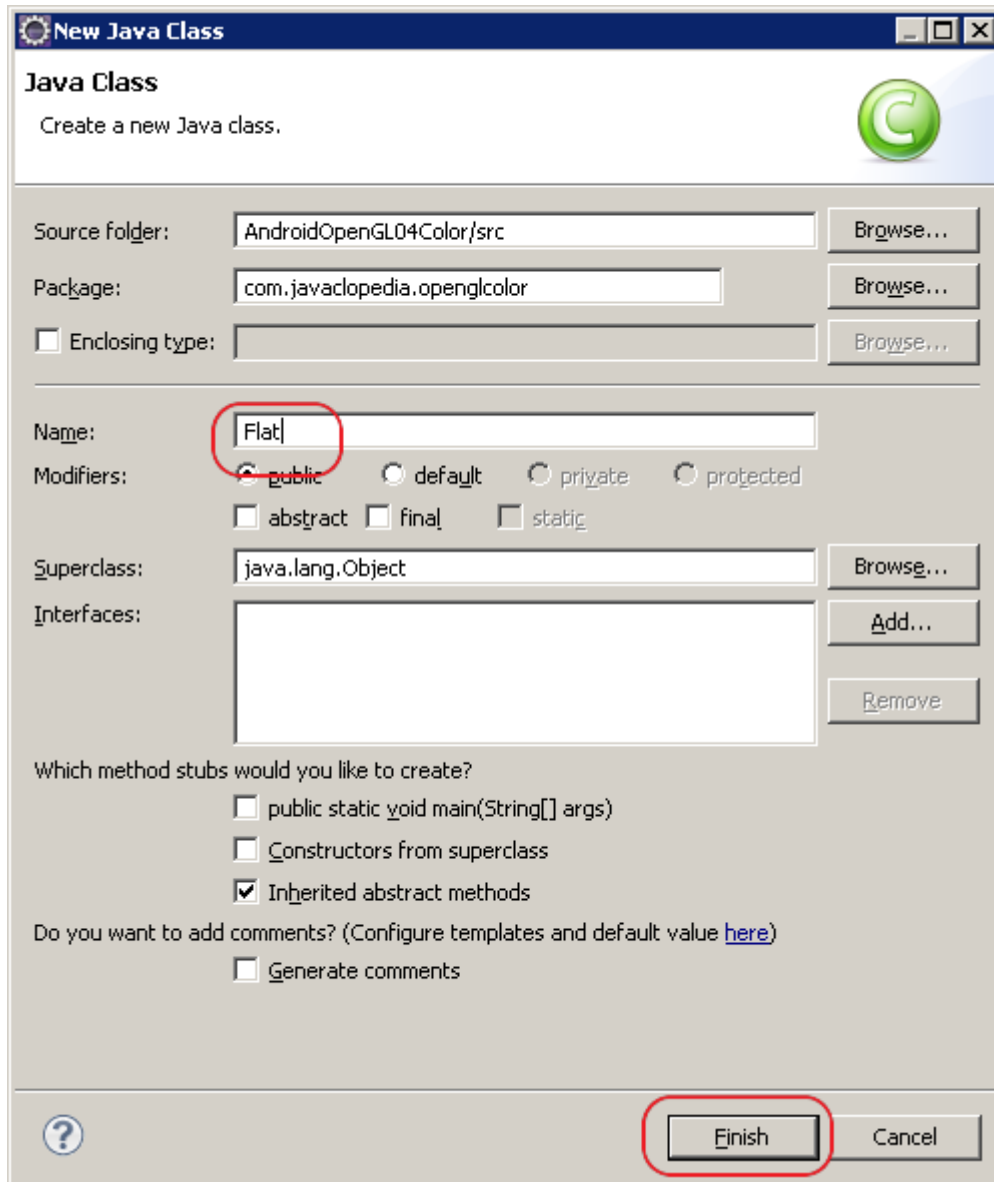
ANDROID IN ACTION!

Tidak perlu berlama-lama, mari kita praktikan di dalam pemrograman:

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 AndroidOpenGL04Color masuk ke bagian src kemudian klik kanan package *com.javaclopedia.openglcolor > New > Class*.
6. Isikan inputan seperti gambar di bawah kemudian tekan Finish.



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

```
TaruRenderer.java Flat.java X
package com.javaclopedia.openglcolor;

public class Flat {
}
```

8. Lengkapi kode program menjadi seperti berikut ini:

```
package com.javaclopedia.openglcolor;

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

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

```
public class Flat {
    private float[] vertices = {
        -1.0f, 1.0f, 0.0f,
        -1.0f, -1.0f, 0.0f,
        1.0f, -1.0f, 0.0f,
        1.0f, 1.0f, 0.0f,
    };

    private short[] indices = { 0, 1, 2, 0, 2, 3 };
    private FloatBuffer vertexBuffer;
    private ShortBuffer indexBuffer;

    public Flat() {
        ByteBuffer vbb =
        ByteBuffer.allocateDirect(vertices.length * 4);
        vbb.order(ByteOrder.nativeOrder());
        vertexBuffer = vbb.asFloatBuffer();
        vertexBuffer.put(vertices);
        vertexBuffer.position(0);

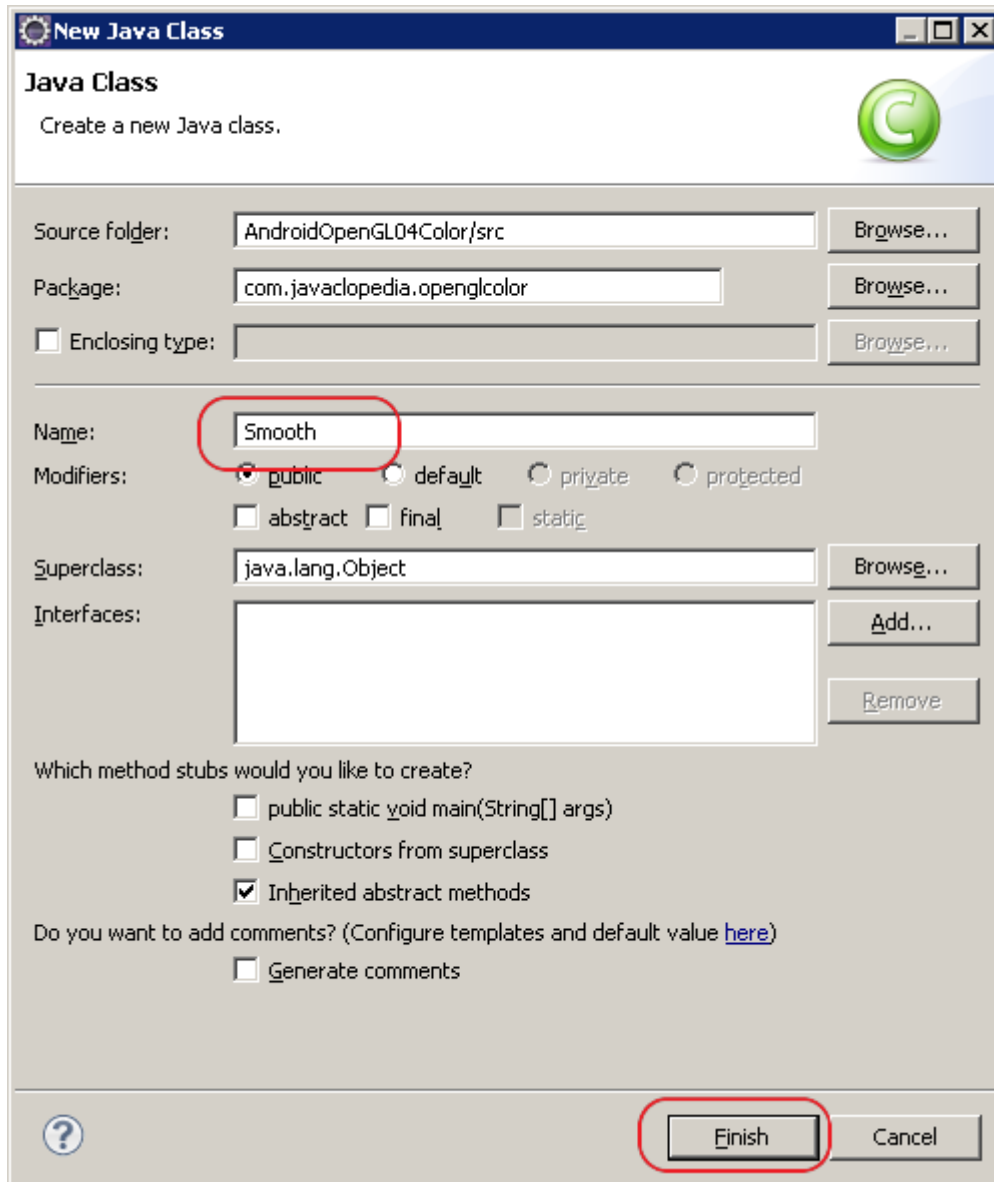
        ByteBuffer ibb = ByteBuffer.allocateDirect(indices.length * 2);
        ibb.order(ByteOrder.nativeOrder());
        indexBuffer = ibb.asShortBuffer();
        indexBuffer.put(indices);
        indexBuffer.position(0);
    }

    public void draw(GL10 gl) {
        gl.glColor4f(0.5f, 0.5f, 1.0f, 1.0f);
        gl.glFrontFace(GL10.GL_CCW);
        gl.glEnable(GL10.GL_CULL_FACE);
        gl.glCullFace(GL10.GL_BACK);

        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, vertexBuffer);
        gl.glDrawElements(GL10.GL_TRIANGLES, indices.length,
            GL10.GL_UNSIGNED_SHORT, indexBuffer);

        gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);
        gl.glDisable(GL10.GL_CULL_FACE);
    }
}
```

- Expand Project AndroidOpenGL04Color masuk ke bagian src kemudian klik kanan package *com.javaclopedia.openglcolor* > New > Class.
- Isikan inputan seperti gambar di bawah kemudian tekan Finish.



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

```
TaruRenderer.java Flat.java Smooth.java X
package com.javaclopedia.openglcolor;

public class Smooth {
}
```

12. Lengkapi kode program menjadi seperti berikut ini:

```
package com.javaclopedia.openglcolor;

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

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

```
public class Smooth {
    private float[] vertices = {
        -1.0f, 1.0f, 0.0f,
        -1.0f, -1.0f, 0.0f,
        1.0f, -1.0f, 0.0f,
        1.0f, 1.0f, 0.0f,
    };

    private short[] indices = { 0, 1, 2, 0, 2, 3 };

    float[] colors = {
        1f, 0f, 0f, 1f,
        0f, 1f, 0f, 1f,
        0f, 0f, 1f, 1f,
        1f, 0f, 1f, 1f,
    };

    private FloatBuffer vertexBuffer;
    private ShortBuffer indexBuffer;
    private FloatBuffer colorBuffer;

    public Smooth() {
        ByteBuffer vbb =
        ByteBuffer.allocateDirect(vertices.length * 4);
        vbb.order(ByteOrder.nativeOrder());
        vertexBuffer = vbb.asFloatBuffer();
        vertexBuffer.put(vertices);
        vertexBuffer.position(0);

        ByteBuffer ibb = ByteBuffer.allocateDirect(indices.length * 2);
        ibb.order(ByteOrder.nativeOrder());
        indexBuffer = ibb.asShortBuffer();
        indexBuffer.put(indices);
        indexBuffer.position(0);

        ByteBuffer cbb = ByteBuffer.allocateDirect(colors.length * 4);
        cbb.order(ByteOrder.nativeOrder());
        colorBuffer = cbb.asFloatBuffer();
        colorBuffer.put(colors);
        colorBuffer.position(0);
    }

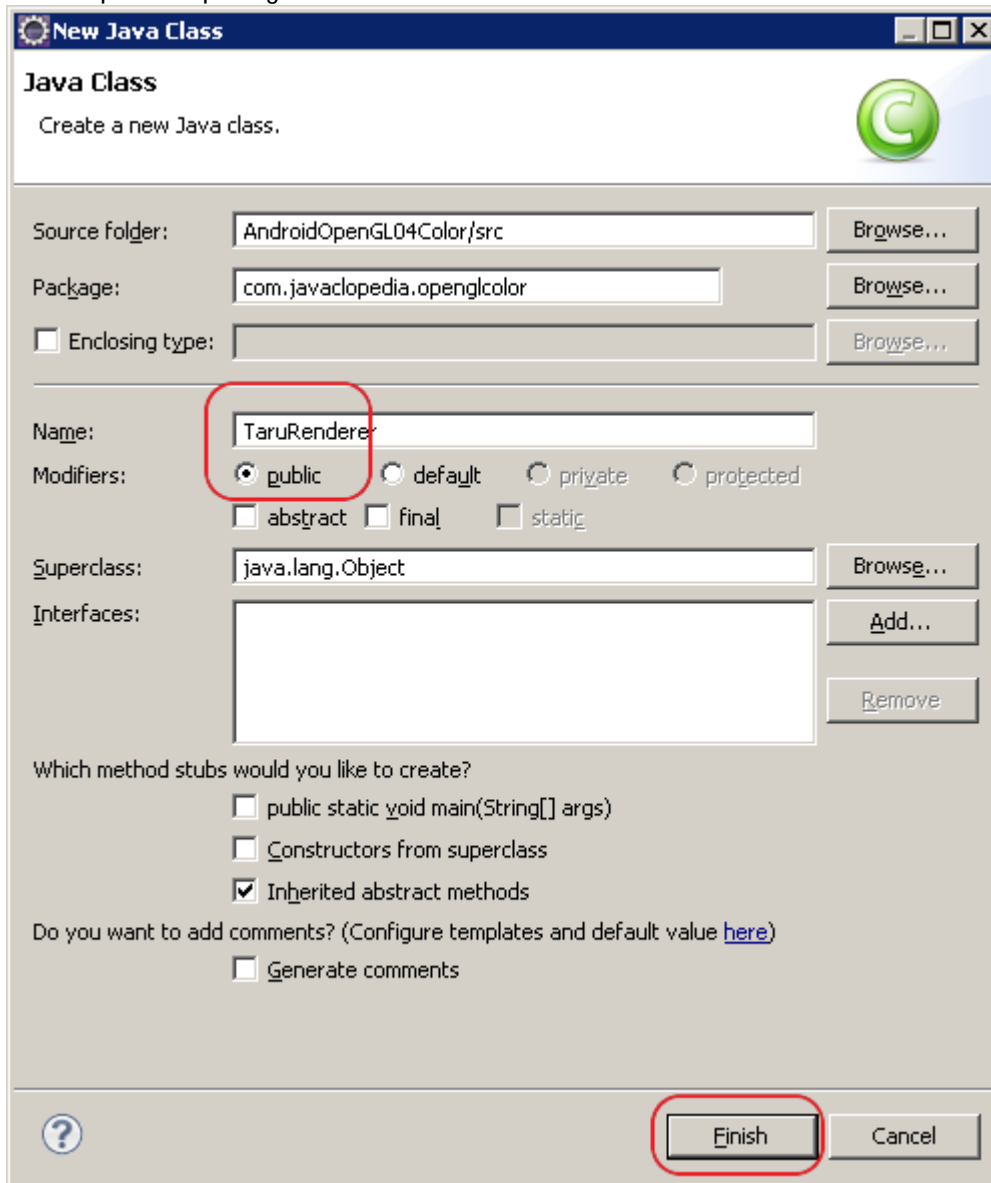
    public void draw(GL10 gl) {
        gl.glFrontFace(GL10.GL_CCW);
        gl.glEnable(GL10.GL_CULL_FACE);
        gl.glCullFace(GL10.GL_BACK);

        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, vertexBuffer);

        gl.glEnableClientState(GL10.GL_COLOR_ARRAY);
        gl.glColorPointer(4, GL10.GL_FLOAT, 0, colorBuffer);

        gl.glDrawElements(GL10.GL_TRIANGLES, indices.length,
            GL10.GL_UNSIGNED_SHORT, indexBuffer);
        gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);
        gl.glDisableClientState(GL10.GL_COLOR_ARRAY);
        gl.glDisable(GL10.GL_CULL_FACE);
    }
}
```

- Expand Project AndroidOpenGL04Color masuk ke bagian src kemudian klik kanan package *com.javaclopedia.openglcolor* > New > Class.
- Isikan inputan seperti gambar di bawah kemudian tekan Finish.



- Akan muncul source code baru seperti gambar di bawah:

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

public class TaruRenderer {
}
```

- Lengkapi kode program menjadi seperti berikut ini:

```
package com.javaclopedia.openglcolor;

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



```
import android.opengl.GLU;
import android.opengl.GLSurfaceView.Renderer;

public class TaruRenderer implements Renderer {
    private Flat flatSquare;
    private Smooth smoothSquare;

    public TaruRenderer() {
        flatSquare = new Flat();
        smoothSquare = new Smooth();
    }

    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        gl.glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
        gl.glShadeModel(GL10.GL_SMOOTH);
        gl.glClearDepthf(1.0f);
        gl.glEnable(GL10.GL_DEPTH_TEST);
        gl.glDepthFunc(GL10.GL_LEQUAL);
        gl.glHint(GL10.GL_PERSPECTIVE_CORRECTION_HINT,
GL10.GL_NICEST);
    }

    public void onDrawFrame(GL10 gl) {
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT |
GL10.GL_DEPTH_BUFFER_BIT);
        gl.glLoadIdentity();
        gl.glTranslatef(0, 1.5f, -7);
        flatSquare.draw(gl);
        gl.glTranslatef(0, -3f, 0);
        smoothSquare.draw(gl);
    }

    public void onSurfaceChanged(GL10 gl, int width, int height) {
        gl.glViewport(0, 0, width, height);
        gl.glMatrixMode(GL10.GL_PROJECTION);
        gl.glLoadIdentity();
        GLU.gluPerspective(gl, 45.0f,
            (float) width / (float) height, 0.1f,100.0f);
        gl.glMatrixMode(GL10.GL_MODELVIEW);
        gl.glLoadIdentity();
    }
}
```

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

```
package com.javaclopedia.openglcolor;

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);
    }
}
```

18. Lengkapi berkas Main.java menjadi seperti beriku ini:

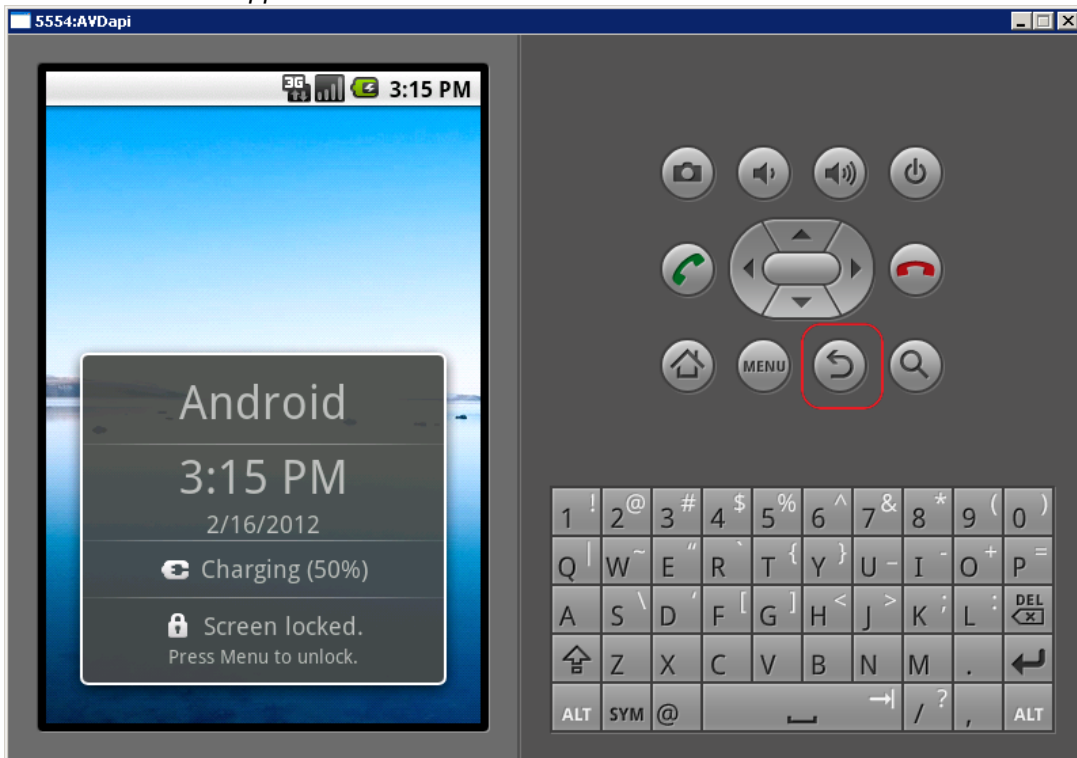
```
package com.javaclopedia.openglcolor;

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

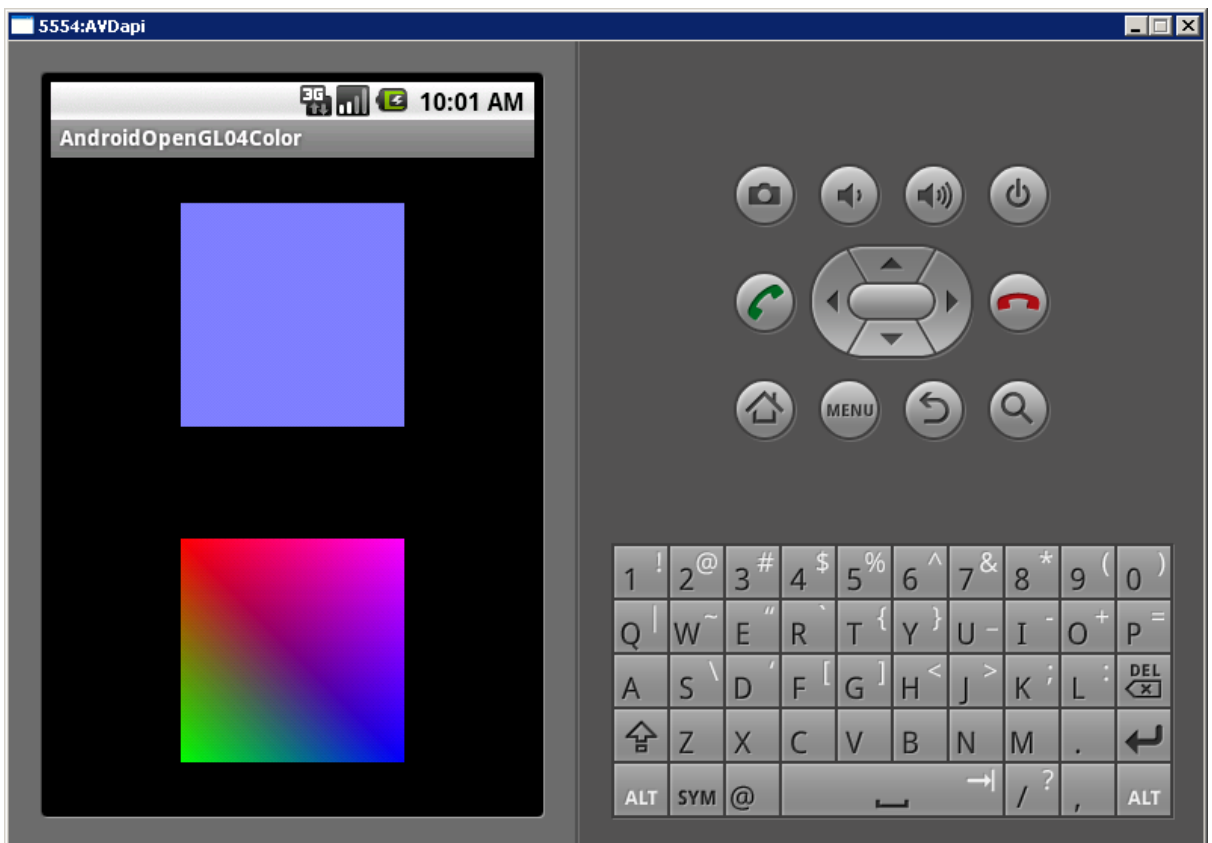
public class Main extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

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

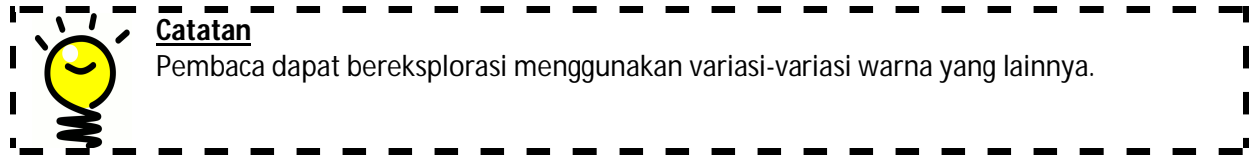
19. Setelah itu, kita jalankan aplikasi kita dengan cara, *Klik Kanan Project AndroidOpenGL04Color > Run As > Android Application.*



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



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



Kesimpulan

1. Color di dalam OpenGL menggunakan ukuran Float dari 0-1
2. Color di dalam OpenGL ada dua macam yaitu Flat dan Smooth
3. Untuk menggunakan Smooth Color harus mengaktifkan `GL_COLOR_ARRAY`

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