Android OpenGL Part 7 - Lighting

[Pemrograman OpenGL Android 07][Level: Mahir]

Andi Taru Nugroho Nur Wismono S.Kom.,M.Cs.
andi.taru@gmail.com

Persiapan

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

- Pemrograman Dasar Android 01 - Instalasi di http://www.javaclopedia.com/
- Pemrograman Android OpenGL 01 – Hello Android OpenGL di http://javaclopedia.com/40/android-opengl-part-1.php
- Pemrograman Android OpenGL 02 – Polygon di OpenGL http://javaclopedia.com/40/android-opengl-part-2.php
- Pemrograman Android OpenGL 03 –Transformasi http://javaclopedia.com/69/tutorial-dasar-android-opengl-part-3-transformasi.php
- Pemrograman Android OpenGL 04-Color http://javaclopedia.com/75/tutorial-dasar-android-opengl-part-4-color.php
- Pemrograman Android OpenGL 05-Cube http://javaclopedia.com/79/tutorial-dasar-android-opengl-part-5-cube.php
- Pemrograman Android OpenGL 06-Texture http://javaclopedia.com/83/tutorial-dasar-android-opengl-part-6-texture.php

Pendahuluan

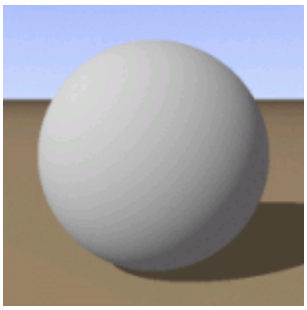
Lighting di dalam OpenGL dibagi menjadi Jenis Cahaya dan Sumber Cahaya. Jenis Cahaya terdiri dari Ambient, Diffuse, dan Specular, sedangkan Sumber Cahaya terdiri dari Directional, Point, Spot.

JENIS CAHAYA

Ambient

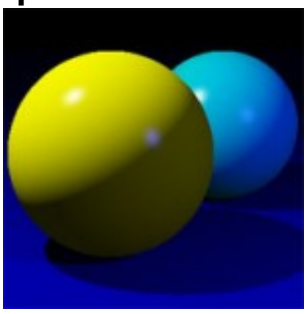
Adalah cahaya yang dipancarkan oleh setiap benda itu sendiri kepada benda yang lain. Ambient biasanya menentukan warna objek.

Diffuse



Adalah cahaya yang dipancarkan oleh setiap benda karena pantulan dari cahaya yang lain. Contohnya adalah ketika kita melihat tembok yang memantulkan cahaya dari lampu.

Specular



Adalah cahaya yang dipantulkan berbalik arah sesuai dengan kondisi obyek. Cahaya spekular akan memberikan efek mengkilap pada suatu benda.

SUMBER CAHAYA

Directional

Sumber cahaya yang datang dari segala arah, bersumber dari tempat yang jauh, merata memiliki kecerahan yang sama. Sumber cahaya yang paling sederhana, kekuatan yang sama, dan arah yang sama pada berbagai tempat.

Point

Sumber cahaya yang berasal dari titik tertentu. Misalnya matahari.

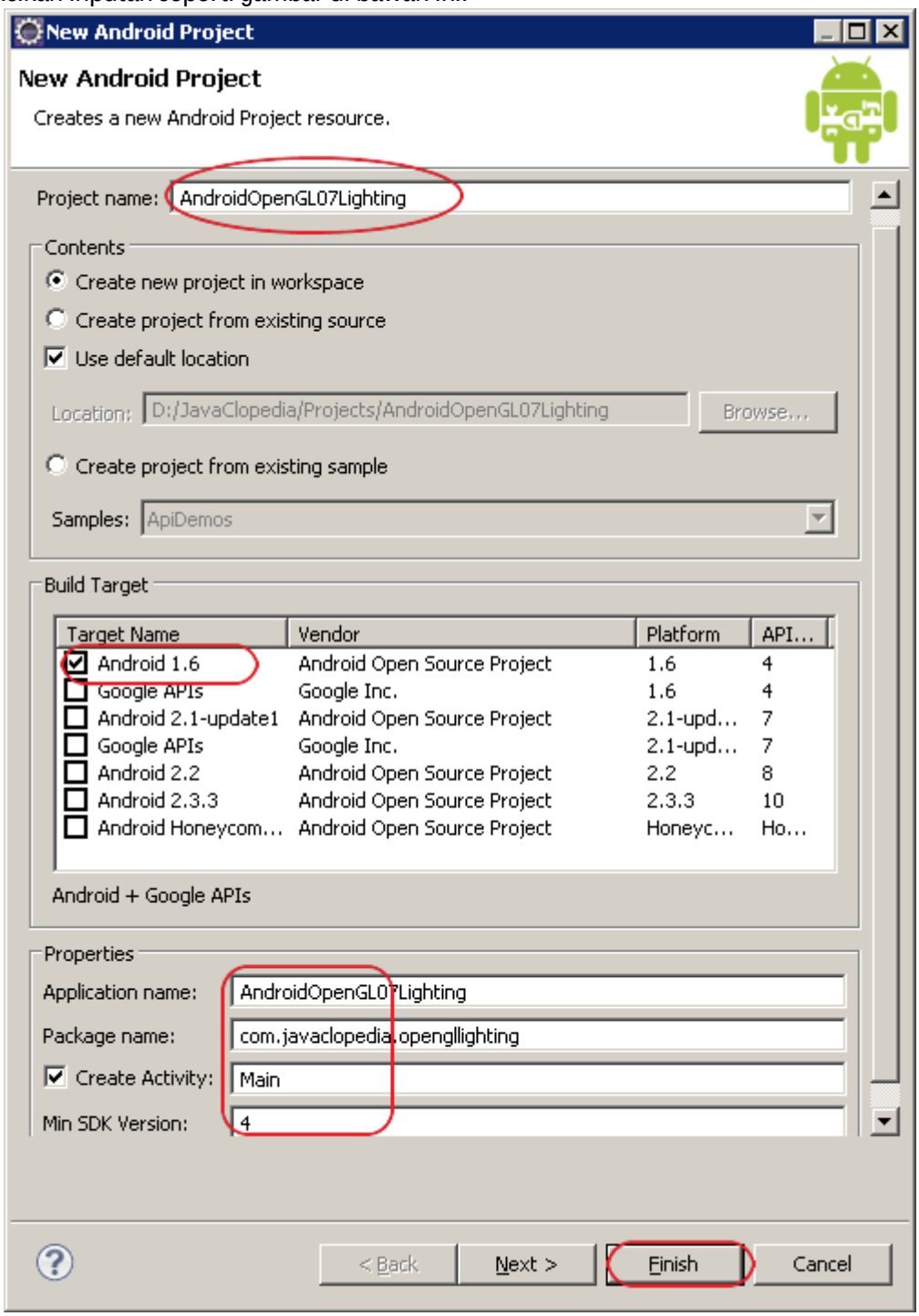
Spot

Sumber cahaya menyoroti bagian tertentu. Misalnya lampu senter.

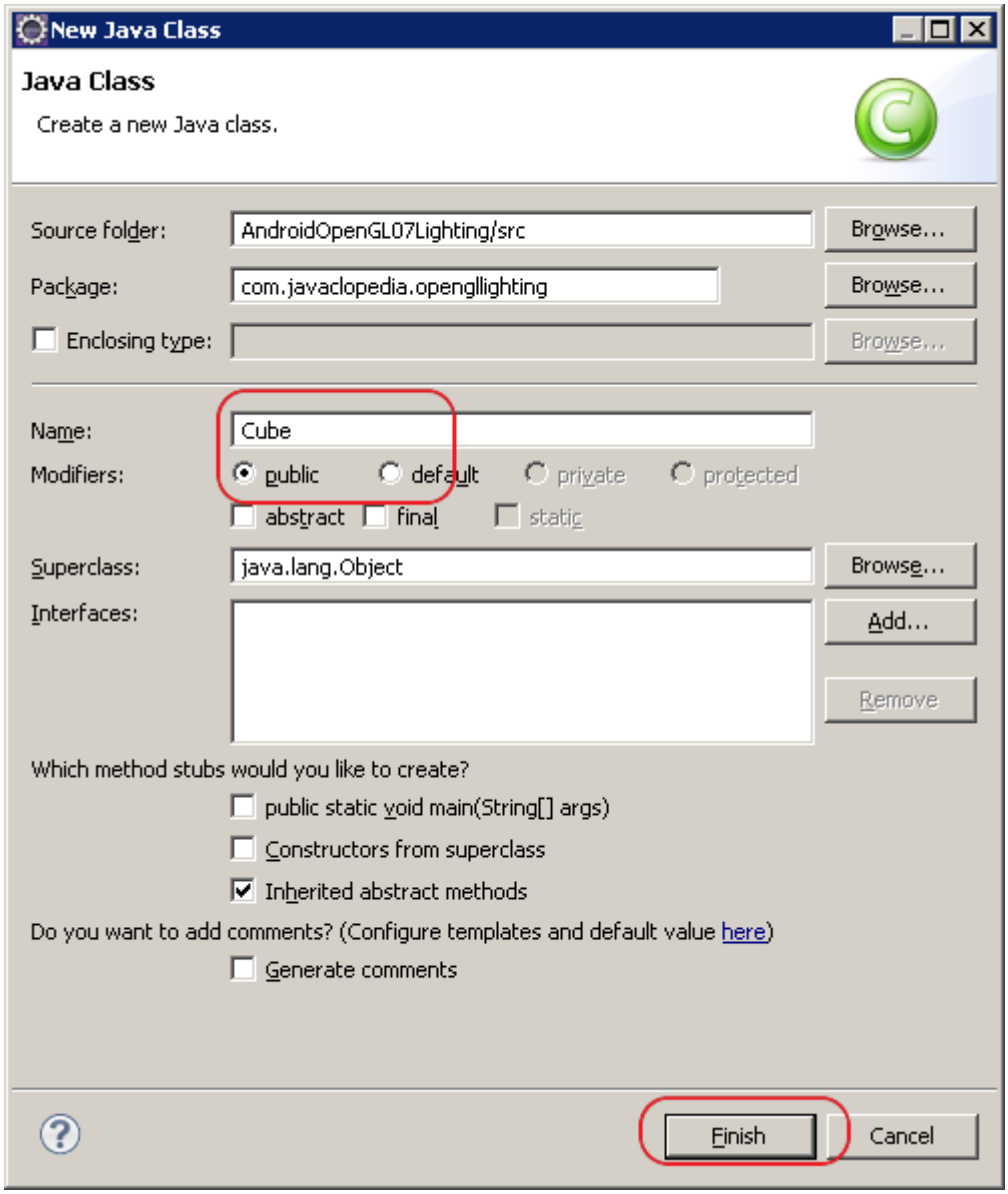
ANDROID IN ACTION!

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

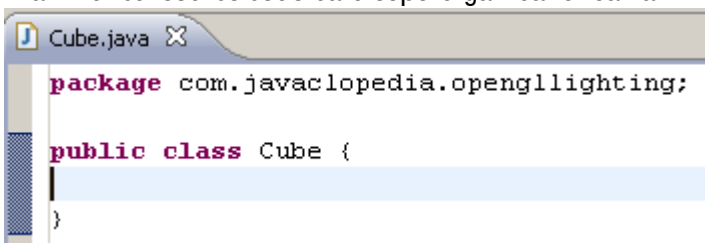
1. Buka Editor Eclipse
2. Buat project baru dengan cara *File* > *New* > *Other* > *Android* > *Android Project* > *Next*.
3. Isikan inputan seperti gambar di bawah ini:



4. Tekan Finish
5. Expand Project AndroidOpenGL07Lighting masuk ke bagian src kemudian klik kanan package *com.javaclopedia.opengllightinge* > *New* > *Class*.
6. Isikan inputan seperti gambar di bawah kemudian tekan Finish.



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



8. Lengkapi kode program menjadi seperti berikut ini:

```
package com.javaclopedia.opengllighting;

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

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

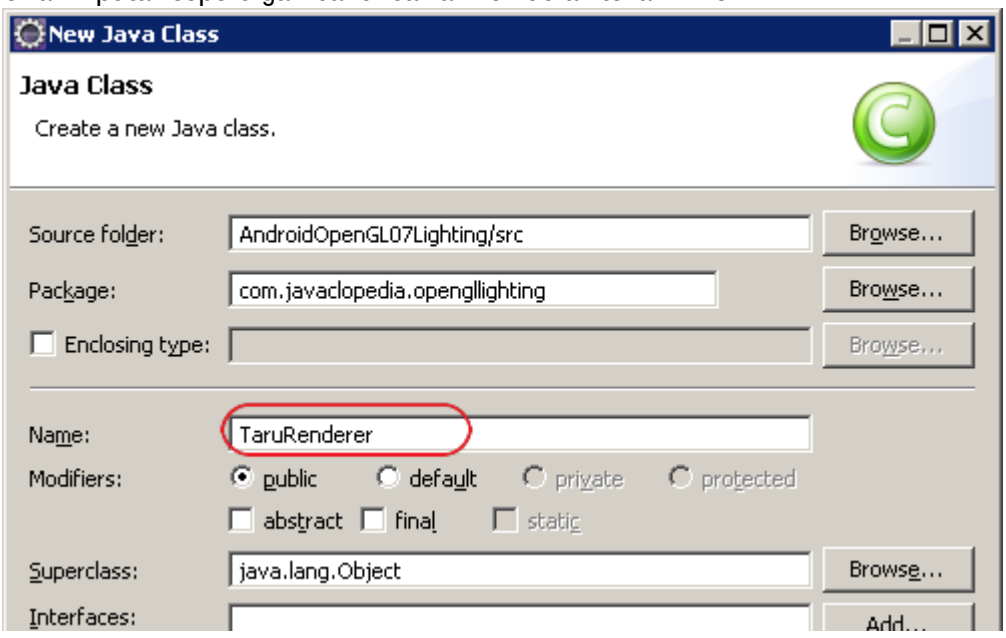
```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.opengl.GLU;

public class Cube {
   Context context;
   float box[] = new float[] {
                // FRONT
                 0.5f, -0.5f,  0.5f,
                 0.5f,  0.5f,  0.5f,
                -0.5f, -0.5f,  0.5f,
                -0.5f,  0.5f,  0.5f,
                // BACK
                -0.5f, -0.5f, -0.5f,
                -0.5f,  0.5f, -0.5f,
                 0.5f, -0.5f, -0.5f,
                 0.5f,  0.5f, -0.5f,
                // LEFT
                -0.5f, -0.5f,  0.5f,
                -0.5f,  0.5f,  0.5f,
                -0.5f, -0.5f, -0.5f,
                -0.5f,  0.5f, -0.5f,
                // RIGHT
                 0.5f, -0.5f, -0.5f,
                 0.5f,  0.5f, -0.5f,
                 0.5f, -0.5f,  0.5f,
                 0.5f,  0.5f,  0.5f,
                // TOP
                -0.5f,  0.5f,  0.5f,
                 0.5f,  0.5f,  0.5f,
                 -0.5f,  0.5f, -0.5f,
                 0.5f,  0.5f, -0.5f,
                // BOTTOM
                -0.5f, -0.5f,  0.5f,
                -0.5f, -0.5f, -0.5f,
                 0.5f, -0.5f,  0.5f,
                 0.5f, -0.5f, -0.5f,
          };

   float texCoords[] = new float[] {
                // FRONT
                 0.0f, 1.0f,
                 0.0f, 0.0f,
                 1.0f, 1.0f,
                 1.0f, 0.0f,
                // BACK
                 1.0f, 0.0f,
                 1.0f, 1.0f,
                 0.0f, 0.0f,
                 0.0f, 1.0f,
                // LEFT
                 1.0f, 0.0f,
                 1.0f, 1.0f,
                 0.0f, 0.0f,
                 0.0f, 1.0f,
                // RIGHT
                 1.0f, 0.0f,
                 1.0f, 1.0f,
                 0.0f, 0.0f,
```

```
                0.0f, 1.0f,
               // TOP
                0.0f, 0.0f,
                1.0f, 0.0f,
                0.0f, 1.0f,
                1.0f, 1.0f,
               // BOTTOM
                1.0f, 0.0f,
                1.0f, 1.0f,
                0.0f, 0.0f,
                0.0f, 1.0f
        };

    FloatBuffer cubeBuff;
    FloatBuffer texBuff;

    float xrot = 0.0f;
    float yrot = 0.0f;

    int tex;

    public Cube(Context context) {
          this.context = context;
          cubeBuff = makeFloatBuffer(box);
          texBuff = makeFloatBuffer(texCoords);
    }

    float lightAmbient[] = new float[] { 0.3f, 0.3f, 0.3f, 1.0f };
    float lightDiffuse[] = new float[] { 0.6f, 0.6f, 0.6f, 1.0f };
    float[] lightPos = new float[] {0,0,3,1};

    float matAmbient[] = new float[] { 1f, 1f, 1f, 1.0f };
    float matDiffuse[] = new float[] { 1f, 1f, 1f, 1.0f };

    protected void init(GL10 gl) {
          gl.glEnable(GL10.GL_LIGHTING);
          gl.glEnable(GL10.GL_LIGHT0);
          gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_AMBIENT,
matAmbient, 0);
          gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_DIFFUSE,
matDiffuse, 0);

          gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_AMBIENT, lightAmbient, 0);
          gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_DIFFUSE, lightDiffuse, 0);
          gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_POSITION, lightPos, 0);

          gl.glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

          gl.glEnable(GL10.GL_DEPTH_TEST);
          gl.glEnable(GL10.GL_CULL_FACE);
          gl.glDepthFunc(GL10.GL_LEQUAL);
          gl.glClearDepthf(1.0f);
          gl.glShadeModel(GL10.GL_SMOOTH);

          gl.glEnable(GL10.GL_TEXTURE_2D);
          tex = loadTexture(gl, R.drawable.icon);
    }

    public void onDrawFrame(GL10 gl) {
```

```
            gl.glClear(GL10.GL_COLOR_BUFFER_BIT |
   GL10.GL_DEPTH_BUFFER_BIT);

            gl.glMatrixMode(GL10.GL_MODELVIEW);
            gl.glLoadIdentity();
            GLU.gluLookAt(gl, 0, 0, 3, 0, 0, 0, 0, 1, 0);

            gl.glVertexPointer(3, GL10.GL_FLOAT, 0, cubeBuff);
            gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
            gl.glTexCoordPointer(2, GL10.GL_FLOAT, 0, texBuff);
            gl.glEnableClientState(GL10.GL_TEXTURE_COORD_ARRAY);

            gl.glRotatef(xrot, 1, 0, 0);
            gl.glRotatef(yrot, 0, 1, 0);

            gl.glColor4f(1.0f, 0, 0, 1.0f);
            gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 0, 4);
            gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 4, 4);

            gl.glColor4f(0, 1.0f, 0, 1.0f);
            gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 8, 4);
            gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 12, 4);

            gl.glColor4f(0, 0, 1.0f, 1.0f);
            gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 16, 4);
            gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 20, 4);

            xrot += 1.0f;
            yrot += 0.5f;
      }

      protected static FloatBuffer makeFloatBuffer(float[] arr) {
            ByteBuffer bb = ByteBuffer.allocateDirect(arr.length*4);
            bb.order(ByteOrder.nativeOrder());
            FloatBuffer fb = bb.asFloatBuffer();
            fb.put(arr);
            fb.position(0);
            return fb;
      }

      protected static ByteBuffer makeByteBuffer(Bitmap bmp) {
            ByteBuffer bb =
   ByteBuffer.allocateDirect(bmp.getHeight()*bmp.getWidth()*4);
            bb.order(ByteOrder.BIG_ENDIAN);
            IntBuffer ib = bb.asIntBuffer();

            for (int y = 0; y < bmp.getHeight(); y++)
                  for (int x=0;x<bmp.getWidth();x++) {
                        int pix = bmp.getPixel(x, bmp.getHeight()-y-1);
                        // Convert ARGB -> RGBA
                        byte alpha = (byte)((pix >> 24)&0xFF);
                        byte red = (byte)((pix >> 16)&0xFF);
                        byte green = (byte)((pix >> 8)&0xFF);
                        byte blue = (byte)((pix)&0xFF);

                        ib.put(((red&0xFF) << 24) |
                                    ((green&0xFF) << 16) |
                                    ((blue&0xFF) << 8) |
                                    ((alpha&0xFF)));
                  }
```

```
            ib.position(0);
            bb.position(0);
            return bb;
    }

    protected int loadTexture(GL10 gl, int resource) {
            int[] tmp_tex = new int[1];

            gl.glGenTextures(1, tmp_tex, 0);
            int tex = tmp_tex[0];
            Bitmap bmp =
BitmapFactory.decodeResource(context.getResources(), resource);

            gl.glBindTexture(GL10.GL_TEXTURE_2D, tex);
            gl.glTexImage2D(GL10.GL_TEXTURE_2D, 0, GL10.GL_RGBA,
bmp.getWidth(), bmp.getHeight(), 0, GL10.GL_RGBA, GL10.GL_UNSIGNED_BYTE,
null);
            gl.glTexSubImage2D(GL10.GL_TEXTURE_2D, 0, 0, 0, bmp.getWidth(),
bmp.getHeight(), GL10.GL_RGBA, GL10.GL_UNSIGNED_BYTE,
makeByteBuffer(bmp));
            gl.glTexParameterf(GL10.GL_TEXTURE_2D,
GL10.GL_TEXTURE_MIN_FILTER, GL10.GL_LINEAR);
            gl.glTexParameterf(GL10.GL_TEXTURE_2D,
GL10.GL_TEXTURE_MAG_FILTER, GL10.GL_LINEAR);

            return tex;
    }
}
```

9. Expand Project AndroidOpenGL07Lighting masuk ke bagian src kemudian klik kanan package *com.javaclopedia.opengllighting* > *New* > *Class*.
10. Isikan inputan seperti gambar di bawah kemudian tekan Finish.



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

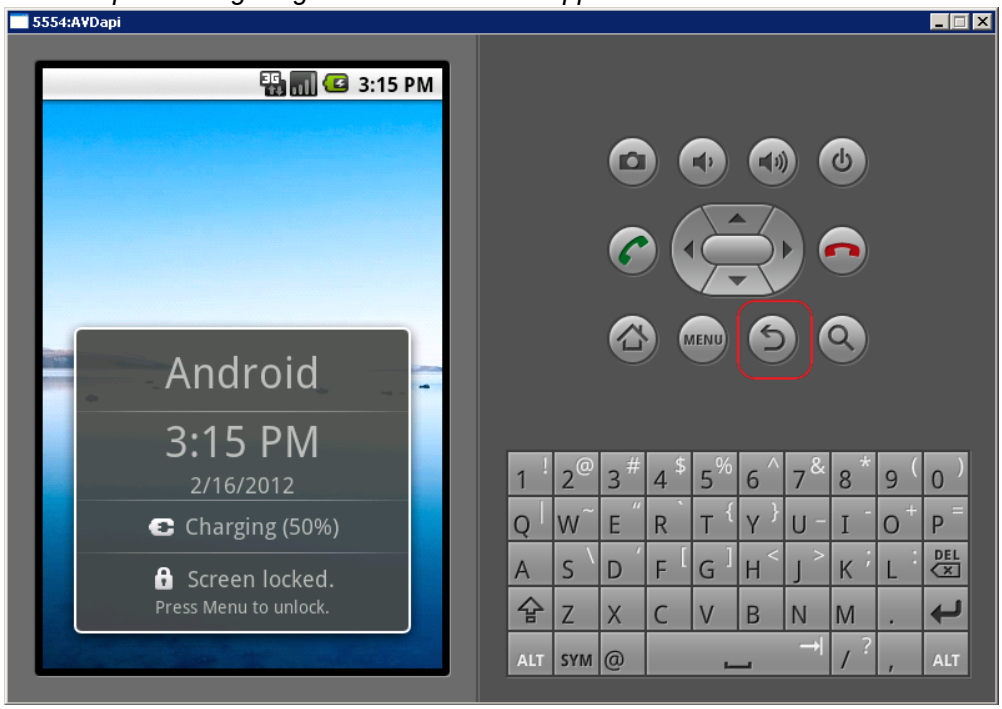
12. Lengkapi kode program menjadi seperti berikut ini:

```
package com.javaclopedia.opengllighting;

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

import android.content.Context;
import android.opengl.GLU;
import android.opengl.GLSurfaceView.Renderer;

public class TaruRenderer implements Renderer {
   private Cube cube;

   public TaruRenderer(Context context) {
         cube = new Cube(context);
   }

   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);

         cube.init(gl);
   }

   public void onDrawFrame(GL10 gl) {
         gl.glClear(GL10.GL_COLOR_BUFFER_BIT |
GL10.GL_DEPTH_BUFFER_BIT);
         gl.glLoadIdentity();
         gl.glTranslatef(0, 0, -4);

         cube.onDrawFrame(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();
   }
}
```

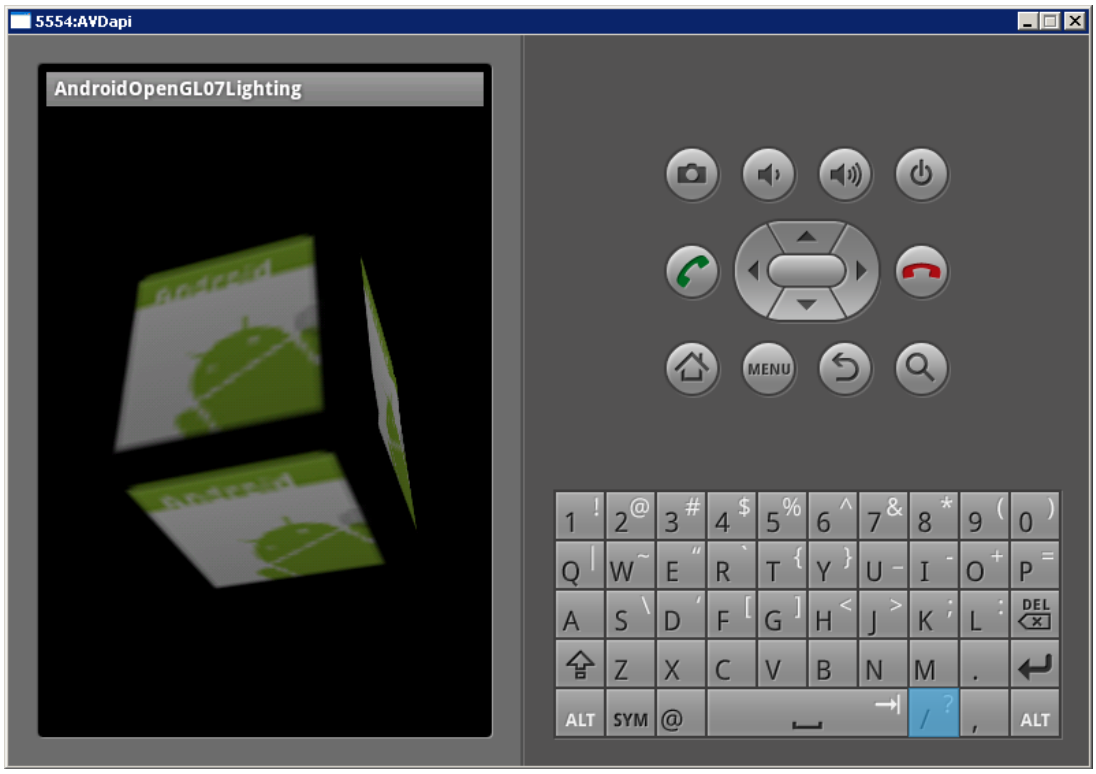
13. Buka class Main, dan ubah menjadi seperti berikut ini:

```
package com.javaclopedia.opengllighting;

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

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


getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN
                    ,WindowManager.LayoutParams.FLAG_FULLSCREEN);

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

    }
}
```

14. Setelah itu, kita jalankan aplikasi kita dengan cara, *Klik Kanan Project AndroidOpenGL07lighting > Run As > Android Application*.



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



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

Kesimpulan

1. Cahaya terdiri dari dua macam yaitu jenis cahaya dan sumber cahaya
2. Jenis cahaya terdiri dari Ambient, Diffuse, Specular
3. Sumber cahaya terdiri dari Directional, Point, Spot

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