

Android OpenGL Part 2 - Polygon

[Pemrograman OpenGL Android 02][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:

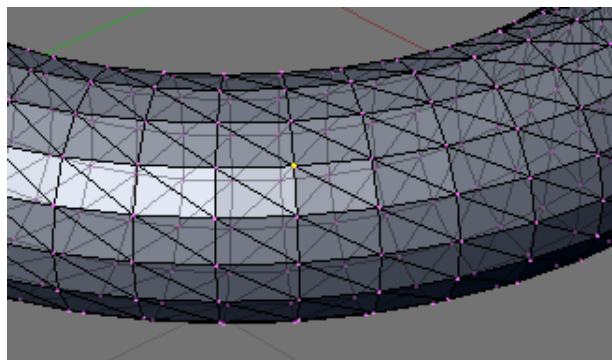
- Pemrograman Dasar Android 01 - Instalasi di <http://www.javaclopedia.com/>
- Pemrograman Android OpenGL 01 – Hello Android OpenGL di <http://javaclopedia.com/40/android-opengl-part-1.php>

Pendahuluan

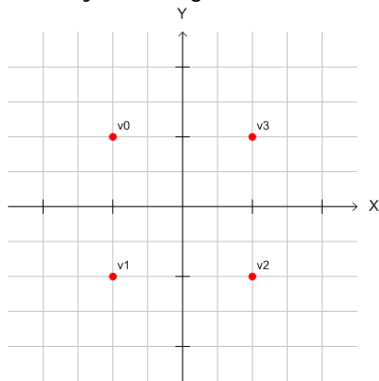
Pada tutorial sebelumnya penulis membahas tentang Hello Android, yaitu dasar pemrograman OpenGL di Android. Apa yang ditampilkan pada tutorial tersebut masih dalam bentuk yang sangat dasar yaitu hanya menampilkan warna background. Pada tutorial ini, penulis akan membahas tentang Vertex, Edge, Face dan Polygon.

Vertex

Apa itu vertex? Vertex adalah titik-titik di dalam dunia 3D. Vertex merupakan elemen terkecil dalam dunia 3D. sebuah vertex dapat mewakili posisi kamera atau cahaya (Gambar, titik berwarna kuning).



Misalnya kita ingin membuat vertex-vertex seperti Gambar berikut:



Maka kita perlu melakukan beberapa hal sebagai berikut:

1. Kita definisikan vertex-vertex tersebut dalam array Float

```
private float vertices[] = {  
    -1.0f, 1.0f, 0.0f, // 0, Top Left  
    -1.0f, -1.0f, 0.0f, // 1, Bottom Left  
    1.0f, -1.0f, 0.0f, // 2, Bottom Right  
    1.0f, 1.0f, 0.0f, // 3, Top Right  
};
```

2. Kemudian kita perlu mengkonversikan float array menjadi FloatBuffer (karena di dalam OpenGL hanya mengenal Buffer)

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

3. Setelah itu kita dapat mengaktifkan status VERTEX_ARRAY dengan cara

```
gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
```

4. Kita menggambar vertex-vertex tersebut dalam OpenGL

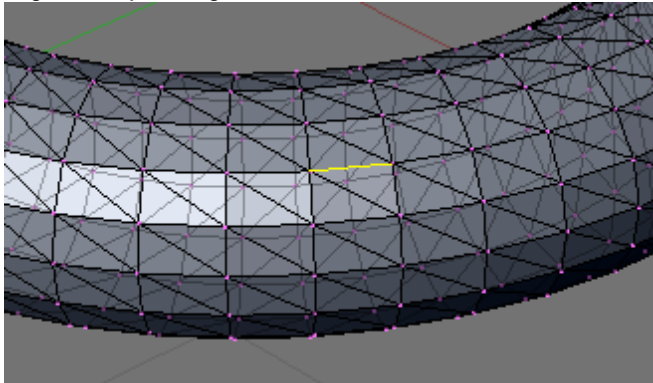
```
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, vertexBuffer);
```

5. Terakhir kita disable kembali status VERTEX_ARRAY

```
gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);
```

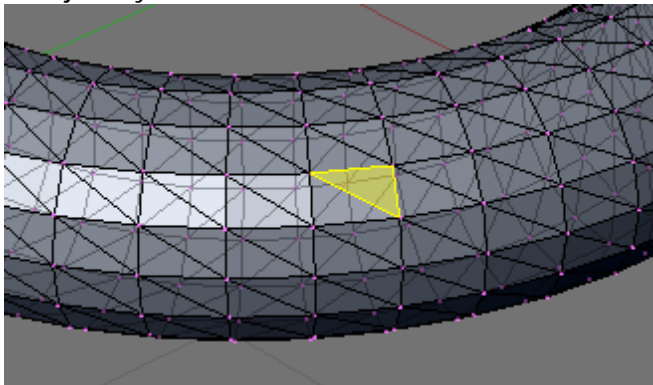
Edge

Edge merupakan garis diantara dua vertex. Lihat Gambar di bawah bagian warna kuning:



Face

Face adalah segitiga, yang terdiri dari tiga vertex dan tiga edge yang menghubungkannya. Untuk lebih jelasnya lihat Gambar di bawah:

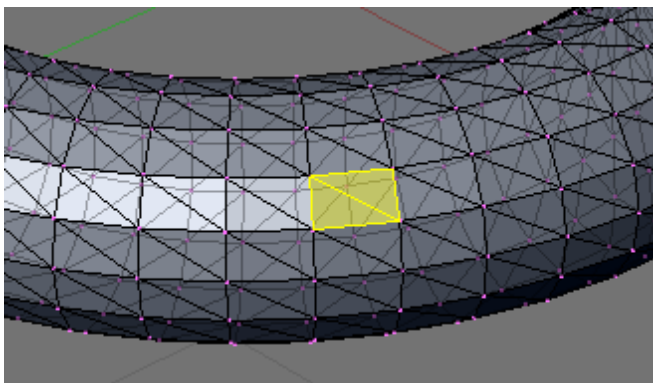


Kita bisa menggunakan Face dengan cara:

```
gl.glFrontFace(GL10.GL_CCW) ;  
gl.glEnable(GL10.GL_CULL_FACE) ;  
gl.glCullFace(GL10.GL_BACK) ;
```

Polygon

Sekarang apa itu Polygon? Polygon adalah gabungan dari beberapa Face. Untuk lebih jelasnya lihat Gambar di bawah:





Catatan

Polygon merupakan bentuk dasar atau primitives. Berbagai object 3D seperti kubus, silinder dan lain-lain merupakan kombinasi dari banyak sekali Polygon.

Hello OpenGL Android (ANDROID IN ACTION!)

Tidak perlu berlama-lama, mari kita belajar membuat Hello OpenGL di Android:

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:

New Android Project

Creates a new Android Project resource.

Project name:

Contents

- Create new project in workspace
- Create project from existing source
- Use default location

Location:

- Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API ..
<input checked="" type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Google APIs	Google Inc.	1.6	4
<input type="checkbox"/> Android 2.1-update1	Android Open Source Project	2.1-upd...	7
<input type="checkbox"/> Google APIs	Google Inc.	2.1-upd...	7
<input type="checkbox"/> Android 2.2	Android Open Source Project	2.2	8
<input type="checkbox"/> Android 2.3.3	Android Open Source Project	2.3.3	10
<input type="checkbox"/> Android Honeycom...	Android Open Source Project	Honeyc...	Ho...

Standard Android platform 1.6

Properties

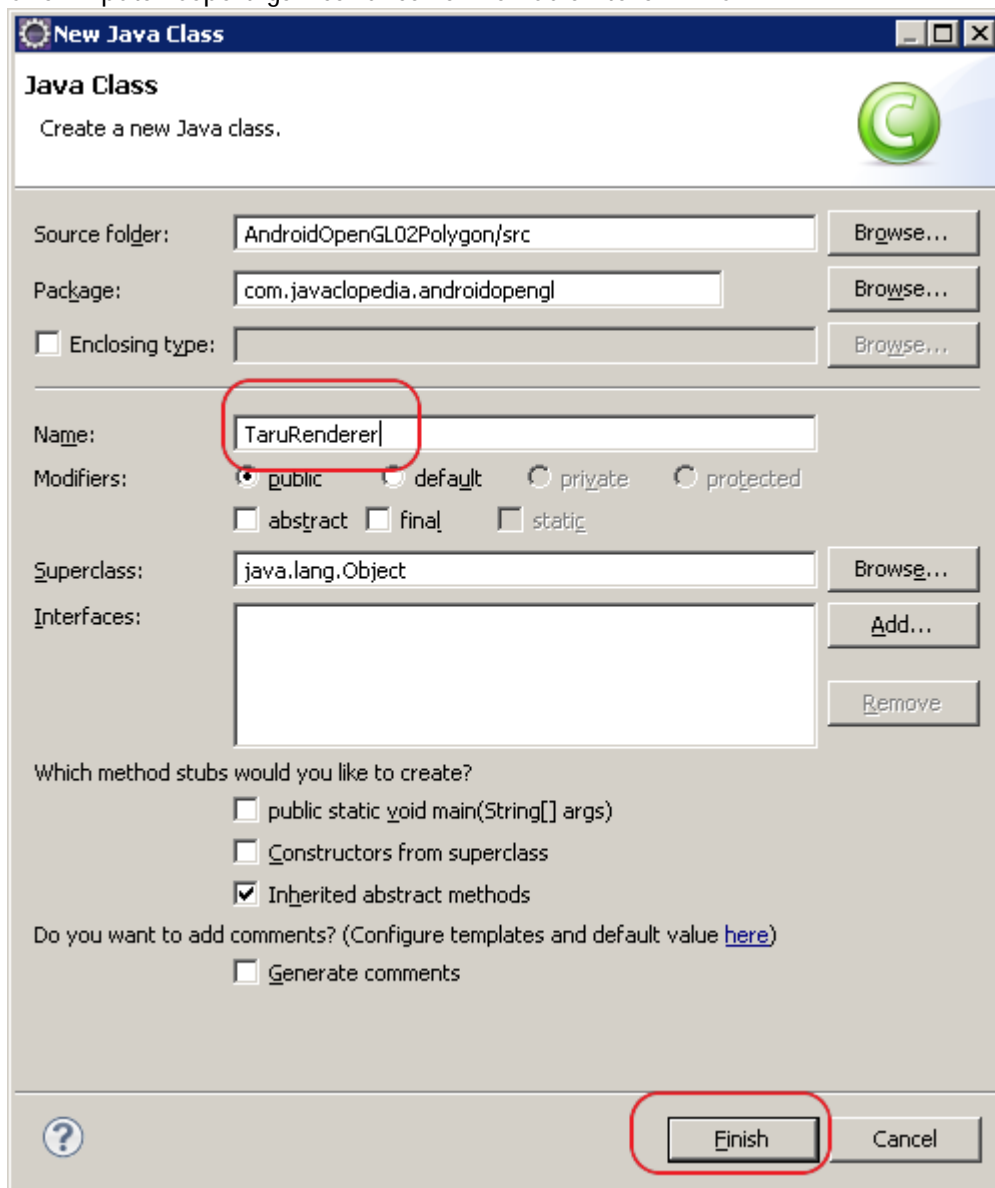
Application name:

Package name:

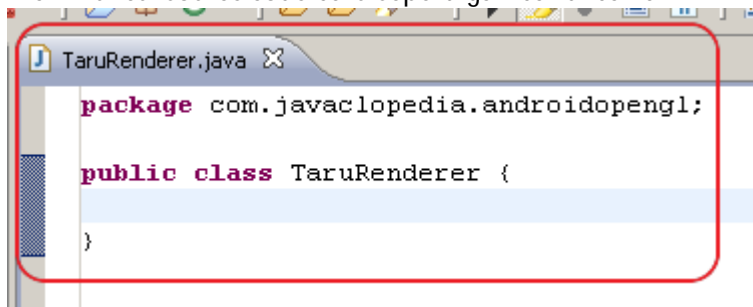
Create Activity:

Min SDK Version:

4. Tekan Finish
5. Expand Project AndroidOpenGL02Polygon masuk ke bagian src kemudian klik kanan package *com.javaclopedia.androidopengl* > 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.androidopengl;

import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.FloatBuffer;
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.glDrawElements(GL10.GL_TRIANGLES, 3,
            GL10.GL_UNSIGNED_SHORT, _indexBuffer);
    }

    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();

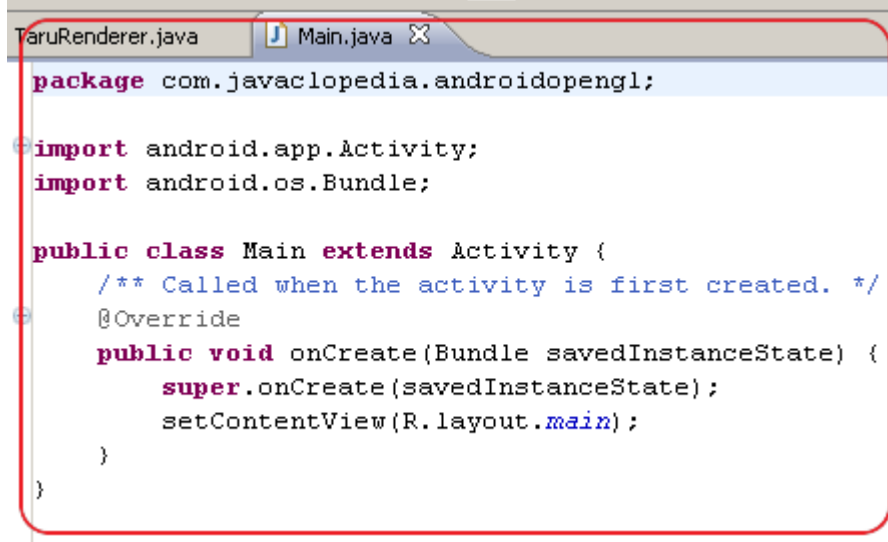
        float[] coords = {
            -0.5f, -0.5f, 0f, // (x1, y1, z1)
            0.5f, -0.5f, 0f, // (x2, y2, z2)
            0f, 0.5f, 0f // (x3, y3, z3)
        };

        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:

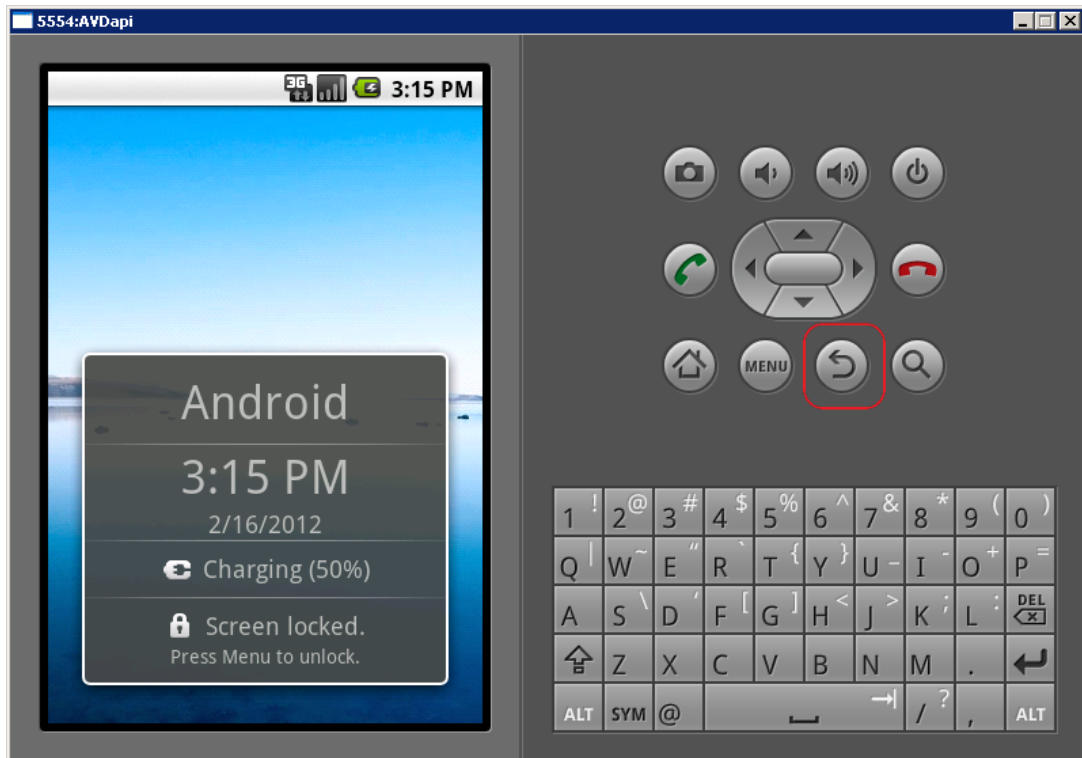


```
package com.javaclopedia.androidopengl;  
  
import android.app.Activity;  
import android.os.Bundle;  
  
public class Main extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);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

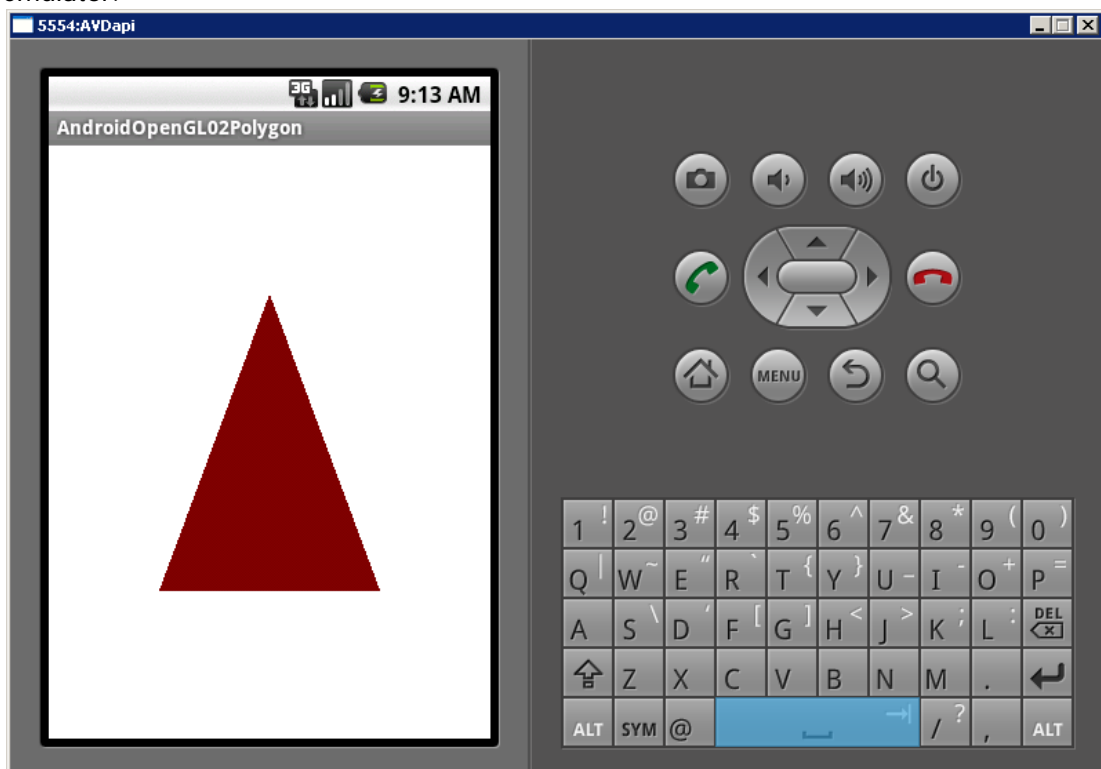
10. Lengkapi berkas Main.java menjadi seperti beriku ini:

```
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 AndroidOpenGL02Polygon > 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 Polygon OpenGL di Android. **SELAMAT!!!**

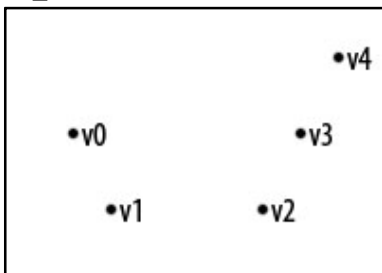
Object Primitives

Selain polygon yang telah kita buat di atas, kita bisa membuat objek-objek primitive lainnya dengan menggunakan konstanta `glBegin()` dan `glEnd()`. Berikut objek yang dapat kita gunakan di dalam OpenGL.

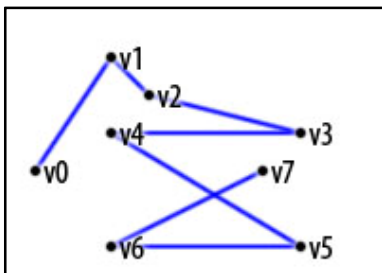
Contoh:

```
gl.glBegin(GL10.GL_LINE_STRIP);  
    gl.glPoints(0.1f,0.2f,0.0f);  
    gl.glPoints(0.1f,0.6f,0.0f);  
gl.glEnd();
```

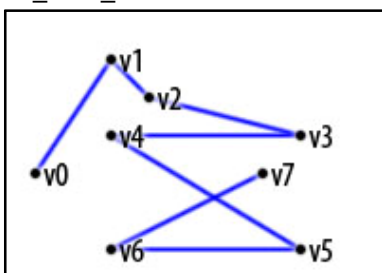
1. GL_POINTS



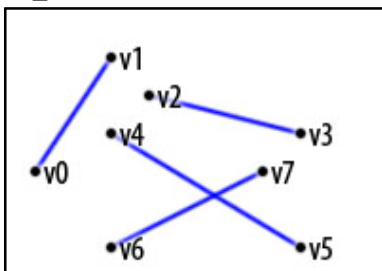
2. GL_LINE_STRIP



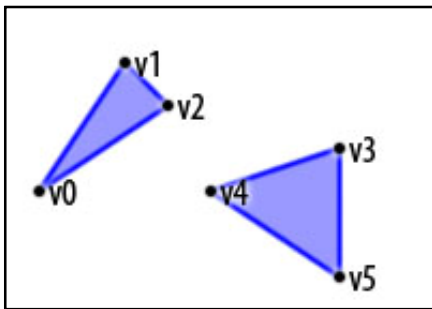
3. GL_LINE_LOOP



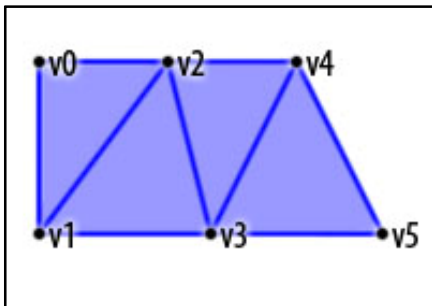
4. GL_LINES



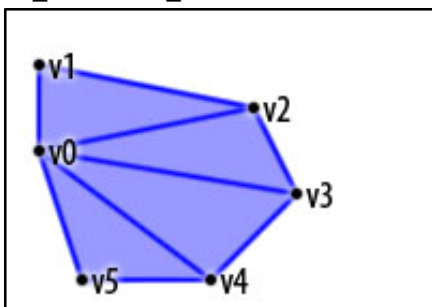
5. GL_TRIANGLES



6. GL_TRIANGLE_STRIP



7. GL_TRIANGLE_FAN



Kesimpulan

1. Elemen dasar dari objek 3D di dalam OpenGL terdiri dari Vertex, Edge, Face dan Polygon
2. Setiap vertex didefinisikan dengan array float (x,y,z) kemudian harus dikonversikan kedalam bentuk FloatBuffer
3. Terdapat berbagai macam objek primitive yang dapat dipakai di dalam OpenGL
4. Objek3D terdiri dari banyak vertex dan face.

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