# Android Open GLES

1. Pokok Bahasan Workshop
2. Pengenalan Open GL
3. Pengenalan OpenGL ES environment
4. Membangun OpenGL ES environment dan menggambar object 3D sederhana
5. Tujuan Instruksional Workshop
6. Peserta memahami dasar-dasar Open GL dan proses rendering
7. Peserta memahami, mengetahui dan dapat menggunakan OpenGL pada aplikasi Android
8. Durasi

120 menit

# MATERI

## Open GL dan OpenGL ES pada Android

OpenGL (*Open Graphics Library*) adalah spesifikasi standar yang mendefinisikan sebuah lintas-bahasa, lintas platform API untuk mengembangkan aplikasi yang menghasilkan grafis komputer 2D maupun 3D. OpenGL dikembangkan oleh Silicon Graphics Inc (SGI) pada tahun 1992 dan secara luas digunakan dalam CAD, *augmented reality*, *virtual reality*, visualisasi ilmiah, visualisasi informasi, dan simulasi penerbangan. Ini juga digunakan dalam video game, yang bersaing dengan Direct3D di Microsoft Windows platform. OpenGL dikelola oleh sebuah teknologi konsorsium nirlaba yaitu Khronos Group.



OpenGL for Embedded Systems (OpenGL ES) adalah bagian dari OpenGL untuk merender grafik komputer 3D dan 2D menggunakan unit pemroses grafis (GPU). Application programming interface (API) ini dapat digunakan pada berbagai perangkat dan platform sehingga tersedia pula hampir semua perangkat berbasis Androi. OpenGLES API yang disediakan oleh Android menawarkan seperangkat *tools* untuk menampilkan grafis animasi kelas atas yang memanfaatkan *graphic prosesing unit* (GPU) yang disediakan di banyak perangkat Android.

## Android Open GLES environment

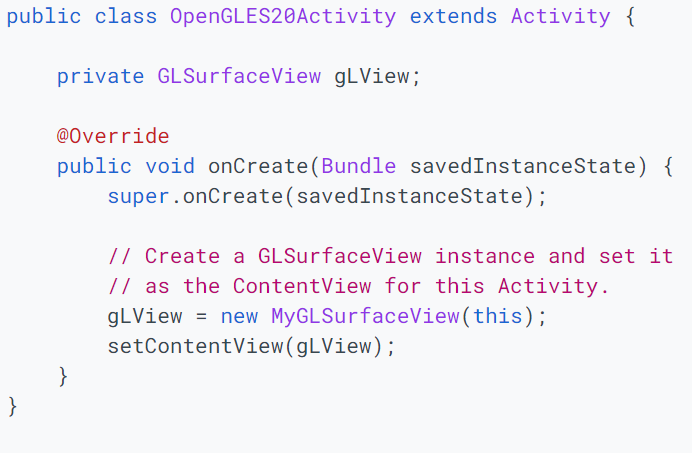
Untuk menggambar grafik dengan OpenGL ES di aplikasi Android, Anda harus membuat *container* untuk grafik tersebut. Salah satu cara mudah untuk melakukannya adalah dengan mengimplementasikan GLSurfaceView dan GLSurfaceView.Renderer. GLSurfaceView adalah *view container* yang diperuntukan unutk menggambar grapfik dengan OpenGL. Sedangkan GLSurfaceView. Renderer mengontrol apa yang digambar dalam *view* tersebut. GLSurfaceView hanyalah salah satu cara untuk menggabungkan grafik OpenGL ES ke dalam aplikasi Android.

### Mendeklarasikan OpenGLES pada manifest

Tambahkan deklarasi berikut pada file Manifest.xml untuk dapat menggunakan OpenGL ES.



Untuk dapat menggunakan OpenGLES maka GLSurfaceView harus disertakan pada activity aplikasi Android. Contoh berikut melakukan penambahan GlSurfaceView pada activity.



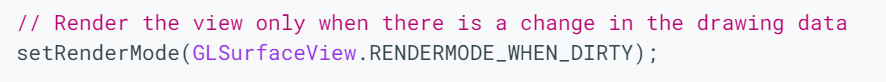
### Membuat GLSurfaceView object

GLSurfaceVIew adalah kompone *view* khusus ang digunakan untuk menggambar grafik OpenGL ES. Kemudian proses penggambaran grafik 3D dilakukan oleh class GlSurfaceView.Renderer. Contoh berikut adalah kode implementasi GlSurfaceView yang akan digunakan activity yang sebelumnya telah dibuat.

Graphical user interface, text, application

Description automatically generated

Konfigurasi tambahan dapat diberikan untuk memastikan proses penggambaran grafis dilakukan saat terjadi perubahan pada data grafis.



Konfigurasi tambahan ini mencegah frame GLSurfaceView Digambar ulang hingga method requestRender() dipanggil.

### Membuat Renderer class

GLSurfaceView.Renderer dibutuhkan untuk melakukan proses penggambaran grafik openGL ES. Class ini mengendalikan apa saja grafik yang akan Digambar oleh GLSurfaceVIew. Terdapat 3 method yang akan dipanggil oleh sistem Android :

* + 1. onSurfaceCreated() : dipanggil saat setup environment *view* OpenGL ES
    2. onDrawFrame() : dipanggil saat proses penggambaran grafis
    3. onSurfaceChanged() : dipanggil jika geometri view berubah, misalnya saat oreintasi smartphone berubah dari portrait ke landscape.

Berikut ini adalah contoh implementasi dari OpenGL ES renderer yang melakukan penggambaran background berwarna hitam pada GLSurfaceView.

Graphical user interface, text, application, email

Description automatically generated

## Mendefinisikan geometri sederhana

OpenGL ES digunakan untuk menggambarkan obyek menggunkan kordinat tiga dimensi. Oleh karena itu, dibutuhak definisi kordinat sebelum menggambarkan geometri sederhana seperti ebuah segitiga. Cara yang paling umum digunakan dalam OpenGL adalah dengan mendefinisikan array vertex bertipe float untuk menyimpan titik kordinat segitiga. ByteBuffer digunakan, untuk meningkatkan efisiensi pengiriman data dari CPU ke GPU melalui OpenGL ES graphics pipeline. Berikut ini adalah contoh class segitiga yang memiliki definisi vertex dan menggunakan buffer.

Text

Description automatically generated

Text

Description automatically generated with low confidence

OpenGL ES menggunakan sistem kordinat dimana [0, 0, 0] (X, Y, Z) menandakan center GLSurfaceView frame, [1, 1, 0] menandakan sudut pojok kanan atas frame, dan [-1, -1, 0 ] menandakan sudut pojok kiri bawah dari frame.

Shape

Description automatically generated

## Menggambar geometri sederhana

Setelah geometri telah dipersiapkan maka proses penggambaran dapat dilakukan. Proses ini diawali dengan inisialisasi dan memuat geometri yang akan digambar ke dalam program saat program dibuat, yang dilakukan pada method onSurfaceCreated(). Berikut adalah contoh kode bagaimana proses inisialisasi geometri dilakukan pada class Renderer.

Graphical user interface, text, application

Description automatically generated

Untuk menggambar geomtri ini menggunakan OpenGL ES, maka beberapa tahap perlu dilakukan :

1. Mendefinisikan vertex shader : adalah kode rendering OpenGL ES yang khusus digunakan untuk me-render posisi vertex.
2. Mendefiniskan fragment shader : adalah kode rendering yang khusus digunakan untuk me-render warna vertext.
3. Mendefinisikan program : obyek OpenGL ES yang berisi shaders untuk menggambar.

Vertex dan fragment shader harus di-compile dan ditambahkan ke program OpenGL ES agar dapat digunakan untuk menggambar geometri. Berikut ini adalah contoh shader sederhana untuk mengambar geometri segitiga.

Text

Description automatically generated

Contoh kode berikut adalah proses compile shader yang dilakukan pada class Renderer:

Text

Description automatically generated

Untuk dapat menggambar geometri, shader harus dikompile dan ditambahkan ke program OpenGL ES. Proses ini dilakukan satu kali saat proses inisialisasi geometri. Berikut ini contoh kode proses kompilasi dan penambahan shader ke program OpenGL ES.

Text

Description automatically generated

Proses penggambaran geometri selanjutnya membutuhkan beberapa parameter untuk menkonfigurasi rendering pipeline. Method draw() dapat dibuat untuk meletakan logika proses penggambaran geometri yang kita inginkan. Method ini berisi pengesetan nilai posisi dan warna vertex ke vertex dan fragment shader, serta mengeksekusi proses penggambaran. Contoh berikut ini adalah implementasi method draw.

Text

Description automatically generated

Method draw() ini akan dipanggil didalam method onDrawFrame() pada class Rendering yang sebelumnya sudah dipersiapkan.

A picture containing text

Description automatically generated

Berikut ini adalah hasil akhir bila aplikasi dijalankan.

Shape

Description automatically generated

Bila diperhatikan, gambar segitiga yang ditampilkan nampak pendek sehingga tidak berbentuk segitiga sama sisi. Ini disebabkan oleh proses penggambaran segitiga tidak disesuaikan dengan proporsi dan ukuran layar dimana GLSurfaceView ditampilkan. Untuk memperbaikinya, dibuthkan proyeksi dan camera view.

## Refrensi

1. <https://developer.android.com/develop/ui/views/graphics/opengl/draw>
2. <https://developer.android.com/develop/ui/views/graphics/opengl/about-opengl>