

Peran Penting OpenGL dalam Perkembangan Teknologi Grafis

Ahmad Thariq Syauqi
Nanda Brilianto
Wiliam Fajar Dicka

Department of Electrical Engineering and Information Technology
Universitas Gadjah Mada
Yogyakarta, Indonesia

Abstract—Seperti yang telah diketahui bersama bahwa OpenGL memiliki peran yang sangat penting dalam dunia teknologi grafis dan perkembangannya. Oleh karena itu, dibutuhkan pemahaman yang mendalam tentang penerapannya terkhusus bagi kalangan penggiat teknologi.

I. PENGERTIAN

Teknologi grafis merupakan salah satu industri yang terbesar di dunia. Di Amerika Serikat, industri teknologi grafis mencakup lebih dari 65.000 perusahaan individu yang mempekerjakan lebih dari satu juta orang. Industri grafis adalah industri yang berkembang sangat pesat sehingga membutuhkan banyak pekerja baru yang memiliki ketrampilan khusus sehingga industri grafis dapat terus berinovasi

Teknologi grafis meliputi dunia percetakan, yakni segala sesuatu yang berhubungan dengan desain dalam berbagai bentuk. Industri ini meliputi percetakan elektronik dan tradisional, penerbitan, pengemasan produk, digital imaging, komputer grafis, pengembangan website, fotografi digital, elektronik cetak dan bidang terkait lainnya.

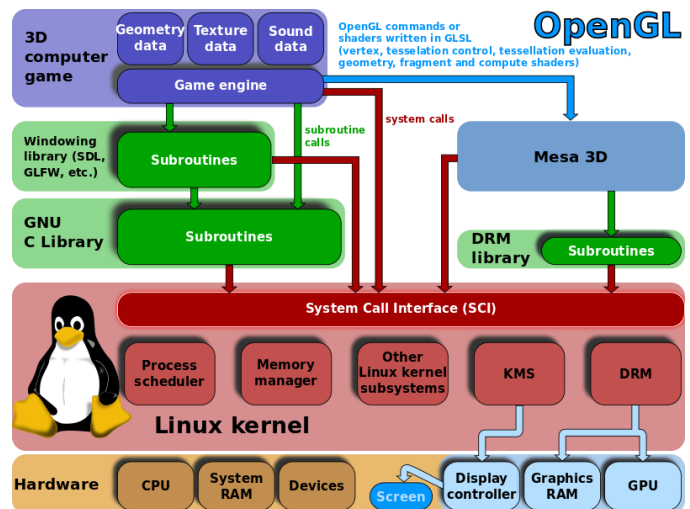
Semua bidang ini saling terhubung satu sama lain sehingga memungkinkan untuk terciptanya peluang di berbagai bidang, seperti komunikasi yang melibatkan penciptaan, produksi, manajemen dan distribusi iklan, pemasaran, website, buku, majalah, surat kabar, katalog, dan paket dalam bentuk cetak dan digital.

Dalam teknologi grafis dikenal juga istilah OpenGL. *Open Graphics Library* (OpenGL) adalah Application Programming Interface (API) multi-bahasa, multi-platform untuk menerjemahkan grafik vektor 2D dan 3D. API umumnya digunakan untuk berkomunikasi dengan *Graphics Processing Unit* (GPU) agar hardware dapat melakukan pekerjaan yang berhubungan dengan grafis.

OpenGL dikembangkan oleh Silicon Graphics Inc., (SGI) pada tahun 1991 dan dirilis setahun kemudian. Setelah perilisannya, OpenGL banyak digunakan pada bidang *Computer-aided Design* (CAD), *virtual reality*, *scientific visualization*, *information visualization*, simulasi penerbangan, dan video games.

II. PEMBAHASAN

OpenGL bertugas mendefinisikan berbagai instruksi untuk menggambar objek, image (umumnya 3D) dan melakukan berbagai operasi terhadap objek-objek tersebut. OpenGL tidak mengandung source code, hanya spesifikasi saja. Pembuat *Graphical Processing Unit* (GPU) seperti NVIDIA, Intel, Samsung, dan lain-lain yang akan membuat implementasi. Dengan cara ini walaupun GPU diproduksi oleh berbagai produsen dengan berbagai variasi tipe dan implementasi, semuanya dapat diperintah dengan spesifikasi yang sama. Dapat dibayangkan apabila setiap produsen memiliki spesifikasi yang berbeda, suatu game ini hanya dapat dijalankan oleh salahsatu produsen kartu grafis saja.



Gambar 1. Proses render dalam OpenGL

OpenGL dirancang independen terhadap sistem operasi, hardware, maupun bahasa pemrograman yang digunakan. Bahkan jika GPU tidak tersedia, OpenGL dapat dijalankan diatas software yang mengemulasi hardware, tentu dengan kinerja yang lebih rendah.

Saat ini, OpenGL telah menjadi standard industri. OpenGL ada hampir disemua platform: Windows, Linux, Mac, Smartphone, game console, avionic dan berbagai embedded system. Dari sisi software, OpenGL digunakan untuk berbagai macam hal mulai dari game, visualisasi, simulasi, CAD (*Computer-Aided Design*) sampai editing video dan gambar.

Standard yang ada di OpenGL dikelola oleh konsorsium yang berisi berbagai pihak yang berkepentingan dengan komputer grafis. Konsorsium itu disebut Khronos yang anggotanya antara lain: AMD, Intel, NVIDIA, Apple, ARM, Nokia, Qualcomm, Samsung, Sony, Epic Games. Khronos juga mengelola standard lain seperti OpenCL, OpenVG dan WebGL.

OpenGL adalah low level API, artinya, saat suatu objek digambar, *texture*, *shaders*, dan lainnya harus dikirimkan terlebih dahulu sebelum objek dapat di render. Hal ini membuat programming dengan OpenGL terlihat rumit, namun disisi lain menjadi lebih *powerful* dan fleksibel. Untuk programmer pemula hal ini terlihat rumit karena terdapat banyak variasi teknik yang dapat dilakukan untuk mencapai hasil yang sama. Umumnya developer game tidak menggunakan OpenGL secara langsung, tetapi melalui game engine seperti Unity.

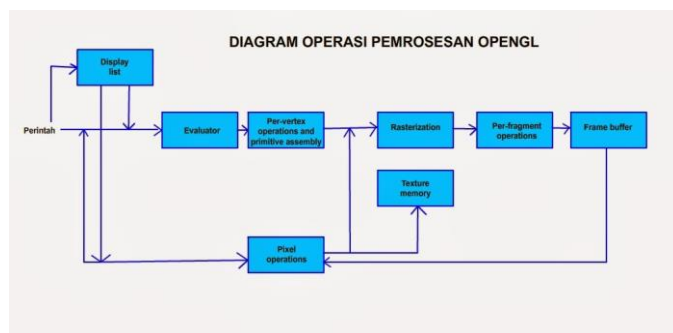
OpenGL terus berkembang baik dalam segi kualitas maupun kompleksitas sehingga pada versi 3, fungsi-fungsi yang dianggap sudah tidak dibutuhkan lagi dibuang. Tetapi karena penolakan beberapa vendor seperti NVIDIA, maka di versi 3.2 diperkenalkan dua profile: yang pertama adalah core profile yang seperti versi 3 membuang bagian yang tidak diperlukan dan versi kedua adalah compatibility profile yang tetap compatible sampai dengan OpenGL versi 1.

III. FUNGSI DAN CARA KERJA

OpenGL bertujuan melakukan rendering objek-objek dua dimensi dan tiga dimensi berupa susunan vertex atau pixel ke dalam frame buffer. Beberapa tahap pemrosesan dilakukan oleh OpenGL pada data ini sehingga terbentuklah image akhir pada frame buffer. OpenGL tidak menyediakan fungsi-fungsi seperti windows dan sebagainya karena bergantung pada sistem. Tetapi, OpenGL memiliki fungsi-fungsi *helper* seperti *GL Utility Toolkit* (GLUT), yang merupakan API karya Mark Kilgard yang memungkinkan pengguna untuk membuat windows dan menangani pesan-pesan. OpenGL berbasiskan pada variabel variabel state dan tidak memiliki class, berbeda dengan DirectX, namun sudah terstruktur secara logis.

Sekumpulan elemen seperti titik, segmen, atau poligon yang disebut dengan primitives digambarkan oleh OpenGL pada beberapa mode yang dapat dipilih. Perintah-perintah dibuat dalam pemanggilan fungsi untuk melakukan operasi OpenGL selain spesifikasi primitives dan pengaturan mode. Sekumpulan vertex melakukan definisi terhadap primitives dan diasosiasikan dengan data berupa koordinat, warna, normal, koordinat tekstur, dan tanda edge. Sekumpulan perintah dibuat oleh suatu aplikasi lalu diterjemahkan dan diproses oleh OpenGL. Sistem window yang mengalokasikan

frame-buffer resources mengendalikan efek akhir dari perintah-perintah OpenGL pada frame buffer.



Gambar 2. Diagram alur pemrosesan OpenGL

Secara garis besar, OpenGL memiliki beberapa tahapan operasi pemrosesan. Tahap pertama adalah *display list*, yaitu pengguna bisa mengumpulkan beberapa perintah-perintah yang digunakan pada display list untuk diproses pada nantinya. Tahap kedua adalah *evaluator*, yang melakukan evaluasi perintah polinomial nilai-nilai input untuk melakukan perhitungan pada kurva dan geometri permukaan. Tahap ketiga adalah *per-vertex operations and primitive assembly*, yaitu tahapan pemrosesan pada OpenGL di mana sekumpulan vertex ditransformasikan dan primitives dipotong dan dimasukkan ke dalam *viewport* untuk dilakukan proses rasterization. Tahap keempat adalah *rasterization*, di mana sekumpulan alamat frame-buffer dan nilai-nilai yang berkaitan dihasilkan dengan deskripsi dua dimensi baik mengenai titik, segmen garis, maupun poligon. Tahap terakhir adalah *per-fragment operations*, yang melibatkan perbaharuan tertentu pada frame-buffer berdasarkan nilai z yang telah disimpan dan yang akan dimasukkan, di mana nilai z ini berguna untuk *z-buffering*. Tahap ini juga melakukan perbaharuan tertentu pada frame-buffer berdasarkan pencampuran warna-warna pixel yang hendak diinput dengan warna yang disimpan, serta operasi-operasi logika lainnya pada nilai-nilai pixel. Data input bisa dalam bentuk kumpulan pixel maupun vertex, yang melewati tahap pertama dan diproses sebagai pixel dalam tahap pixel operations. Hasil tahapan ini bisa disimpan sebagai texture memory, untuk digunakan dalam tahap rasterization, atau dilakukan rasterization dan hasil fragmentasinya digabungkan ke dalam frame-buffer.

OpenGL ES

OpenGL ES adalah versi OpenGL untuk embedded system dan mobile device khususnya untuk iPhone dan Android. Untuk 'merampingkan' OpenGL ES, API OpenGL yang jarang digunakan atau terlalu kompleks pada sistem mobile device dan embedded system dibuang.

IV. KESIMPULAN

Jadi, OpenGL merupakan teknologi grafis yang dirancang independen terhadap sistem operasi, hardware, maupun bahasa pemrograman yang digunakan. Bahkan jika GPU tidak tersedia, OpenGL dapat dijalankan diatas software yang mengemulasi hardware, tentu dengan kinerja yang lebih rendah. Standard yang ada di OpenGL dikelola oleh konsorsium yang berisi berbagai pihak yang berkepentingan dengan komputer grafis, seperti AMD, Intel, dan NVIDIA.

V. DAFTAR PUSTAKA

- [1] <https://sites.google.com/site/graphictechnologiesnewsletter/home/what-is-graphic-technologies> diakses pada 12 Februari 2017
- [2] <https://en.wikipedia.org/wiki/OpenGL> diakses pada 12 Februari 2017
- [3] <https://www.opengl.org/> diakses pada 12 Februari 2017
- [4] <https://www.khronos.org/opengl/> diakses pada 12 Februari 2017
- [5] <http://stackoverflow.com/questions/11695756/what-is-the-purpose-of-a-graphics-library-such-as-opengl> diakses pada 12 Februari 2017