

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

LOADING SCREEN PLAYSTATION 2 DALAM OPENGL

Dosen Pengampu : Sri Rahayu, M.Kom.

Instruktur Pratikum : Arul Budi Kalimat, S.Kom.



Disusun oleh

Kelompok 6

Tabina Athifa Rahmaniya

2306110

Ali Noval Saputra

2306103

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN ILMU KOMPUTER

INSTITUT TEKNOLOGI GARUT

2024

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Grafik Komputer ini. Laporan ini dibuat sebagai salah satu tugas dari mata kuliah Grafik Komputer, dengan tujuan untuk memberikan pemahaman yang lebih baik tentang pembuatan loading screen PlayStation 2 di dalam OpenGL.

Kami mengucapkan terima kasih kepada dosen pengampu ibu Sri Rahayu, instruktur praktikum bapak Arul Budi Kalimat, serta semua pihak yang telah memberikan dukungan dalam penyusunan laporan ini.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, untuk itu kami mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Garut, 12 Januari 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	1
BAB II TINJAUAN PUSTAKA	2
2.1 OpenGL.....	2
2.2 Transformasi di Dalam OpenGL.....	2
2.3 Proyeksi 3D di Dalam OpenGL	4
2.4 Pencahayaan di Dalam OpenGL	4
2.5 Scene di Dalam OpenGL	5
2.6 Loading Screen PlayStation 2 di Dalam OpenGL	5
BAB III HASIL.....	6
3.1 Source Code.....	6
3.2 Output.....	6
3.3 Penjelasan.....	7
BAB IV PENUTUP	8
DAFTAR PUSTAKA	9

DAFTAR GAMBAR

Gambar 1 Scene 1	6
Gambar 2 Scene 2	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

OpenGL (Open Graphics Library) adalah sebuah API (Application Programming Interface) lintas platform yang banyak digunakan untuk rendering grafis 2D dan 3D. Penggunaan OpenGL sangat relevan dalam pengembangan aplikasi grafis interaktif, seperti game, simulasi, dan animasi. Dalam dunia teknologi saat ini, penguasaan OpenGL menjadi salah satu keahlian penting, terutama bagi para pengembang perangkat lunak yang ingin menghasilkan grafik berkualitas tinggi.

Pembuatan loading screen adalah salah satu elemen penting dalam desain aplikasi dan game. Loading screen tidak hanya memberikan waktu bagi sistem untuk memproses data, tetapi juga berfungsi sebagai elemen visual yang meningkatkan pengalaman pengguna. Salah satu contoh loading screen yang ikonik adalah milik PlayStation 2, yang dikenal dengan desain sederhana namun memiliki daya tarik visual yang khas.

Maka dari itu, kelompok kami memutuskan untuk merancang sebuah proyek yang mengangkat tema loading screen PlayStation 2 yang pembuatannya dikemas menggunakan OpenGL.

1.2 Rumusan Masalah

1. Apa yang dimaksud dengan OpenGL?
2. Bagaimana cara penerapan konsep yang tersedia di dalam OpenGL?
3. Bagaimana membuat *Loading Screen PlayStation 2* dalam OpenGL

1.3 Tujuan

1. Mengetahui apa itu OpenGL
2. Mengetahui penerapan konsep yang dituangkan dengan sebuah tema di dalam OpenGL
3. Mengetahui cara pembuatan *Loading Screen PlayStation 2* dalam OpenGL

BAB II

TINJAUAN PUSTAKA

2.1 OpenGL

OpenGL (Open Graphics Library) adalah spesifikasi antarmuka perangkat lunak lintas platform yang digunakan untuk menghasilkan grafika 2D dan 3D. OpenGL menyediakan seperangkat perintah yang digunakan untuk menggambar primitif grafis seperti titik, garis, dan poligon. Didesain agar fleksibel dan berorientasi pada kinerja, OpenGL banyak digunakan dalam pengembangan game, aplikasi simulasi, dan visualisasi data.

OpenGL dirancang untuk berjalan di berbagai sistem operasi, seperti Windows, macOS, dan Linux. Hal ini memungkinkan pengembang untuk membuat aplikasi yang dapat dijalankan di berbagai perangkat keras tanpa perlu mengkhawatirkan kompatibilitas platform tertentu [1].

OpenGL menggunakan grafik pipeline yang mencakup proses seperti vertex processing, rasterization, fragment processing, dan tahap akhir rendering. Versi modern dari OpenGL (sejak versi 3.0) mendukung konsep programmable pipeline, memungkinkan pengembang untuk menggunakan shader yang ditulis dalam GLSL (OpenGL Shading Language) [2].

OpenGL dikembangkan dan dikelola oleh Khronos Group, sebuah konsorsium industri yang terdiri dari perusahaan-perusahaan besar di bidang teknologi. Dukungan luas dari industri ini menjadikan OpenGL sebagai standar de facto dalam grafik komputer [3].

OpenGL memungkinkan pengembang untuk menggunakan ekstensi tertentu yang dirilis oleh vendor perangkat keras, seperti NVIDIA dan AMD, untuk meningkatkan kemampuan rendering sesuai kebutuhan aplikasi [4].

OpenGL digunakan dalam berbagai aplikasi, termasuk pengembangan game (e.g., Minecraft), visualisasi ilmiah (e.g., simulasi molekul), dan perangkat lunak desain berbantuan komputer (e.g., Blender) [5].

2.2 Transformasi di Dalam OpenGL

Transformasi dalam konteks grafika komputer adalah metode untuk mengubah posisi, ukuran, atau orientasi objek dalam ruang. Terdapat beberapa jenis transformasi dasar yang sering digunakan, yaitu translasi, rotasi, dan skalasi. Pada proyek ini, kelompok kami menggunakan penerapan transformasi translasi dan rotasi.

Translasi adalah proses memindahkan objek dari satu lokasi ke lokasi lain tanpa

mengubah bentuk atau ukuran objek tersebut. Dalam translasi dua dimensi, perubahan posisi dilakukan dengan menambah atau mengurangi koordinat x dan y objek dengan nilai tertentu, yang disebut sebagai vektor translasi. Rumus yang digunakan adalah $x' = x + tx$ $y' = y + ty$, di mana tx dan ty adalah nilai pergeseran pada sumbu x dan y, masing masing.

Rotasi adalah transformasi yang memutar posisi objek di sekitar titik pusat atau sumbu tertentu. Sudut rotasi menentukan seberapa jauh objek akan diputar. Fungsi yang digunakan untuk rotasi dalam grafika komputer adalah ***glRotatef(θ , Rx,Ry, Rz)***. Parameter θ adalah sudut rotasi, sedangkan Rx,Ry, Rz Menujukan sumbu rotasi (x, y, z). Jika θ positif, objek akan berputar berlawanan arah jarum jam, jika negative, searah jarum jam. Rumus untuk menghitung koordinat baru setelah rotasi terhadap titik pusat (xp, yp) adalah $x' = xp + (x - xp) \cos(\theta) - (y - yp) \sin(\theta)$ $y' = yp + (x - xp) \sin(\theta) + (y - yp) \cos(\theta)$

2.3 Proyeksi 3D di Dalam OpenGL

3D viewing adalah proses visualisasi objek tiga dimensi pada layar dua dimensi. Dalam konteks grafik komputer, ini melibatkan penggunaan sistem koordinat untuk menggambarkan posisi objek dalam ruang 3D dan cara objek tersebut diproyeksikan ke dalam ruang 2D. Proses ini mencakup berbagai teknik, termasuk proyeksi paralel dan perspektif, yang memungkinkan pengguna untuk melihat objek dari berbagai sudut pandang. Pada proyek ini, objek 3D yang tersedia berupa sebuah kubus.

Proyeksi paralel adalah metode untuk memproyeksikan objek 3D ke dalam bidang 2D dengan menggunakan garis-garis sejajar. Dalam proyeksi ini, jarak antara objek dan bidang proyeksi tidak mempengaruhi ukuran objek yang ditampilkan, sehingga objek terlihat dalam ukuran sebenarnya tanpa distorsi perspektif.

Proyeksi perspektif adalah metode di mana objek diproyeksikan ke dalam bidang 2D dengan mempertimbangkan jarak dan sudut pandang, sehingga memberikan efek kedalaman yang lebih realistis. Dalam proyeksi ini, objek yang lebih jauh akan tampak lebih kecil dibandingkan dengan objek yang dekat, menciptakan ilusi ruang tiga dimensi.

2.4 Pencahayaan di Dalam OpenGL

Pencahayaan di OpenGL terdiri dari beberapa komponen utama, yaitu **ambient**, **diffuse**, dan **specular**, yang masing-masing berkontribusi pada cara cahaya berinteraksi dengan objek. **Ambient light** adalah cahaya yang tersebar merata ke segala arah, memberikan iluminasi dasar pada seluruh objek tanpa memandang posisi atau arah sumber cahaya. Cahaya ini mensimulasikan pantulan dari permukaan lingkungan sekitar, sehingga objek tetap terlihat meskipun tidak terkena cahaya langsung.

Diffuse light adalah cahaya yang berasal dari sumber cahaya tertentu dan menyebar saat mengenai permukaan. Komponen ini menciptakan efek pencahayaan yang bergantung pada sudut antara sumber cahaya dan permukaan objek, menghasilkan bayangan lembut yang realistis pada objek dengan permukaan yang kasar atau matte.

Specular light, di sisi lain, adalah cahaya yang dipantulkan langsung dari permukaan yang mengkilap atau reflektif. Komponen ini menghasilkan highlight atau kilau terang di area tertentu pada objek, memberikan ilusi permukaan yang halus seperti logam atau kaca. Kombinasi dari ketiga jenis pencahayaan ini memungkinkan OpenGL untuk menciptakan efek pencahayaan yang realistis pada model 3D.

2.5 Scene di Dalam OpenGL

Scene adalah lingkungan yang dapat ditampilkan pada satu layar penuh. Dalam dunia game, scene dapat merujuk ke layar menu utama, gameplay, atau game over screen. Konsep scene di OpenGL mirip dengan halaman atau view di aplikasi modern.

2.6 Loading Screen PlayStation 2 di Dalam OpenGL

Loading screen PlayStation 2 merupakan tema yang diangkat untuk proyek tugas besar kelompok 6 ini, dengan menerapkan konsep-konsep OpenGL yang sudah dikenali sebelumnya. Di antaranya adalah penerapan dari transformasi yang meliputi translasi dan rotasi, proyeksi 3 dimensi, pencahayaan, dan scene.

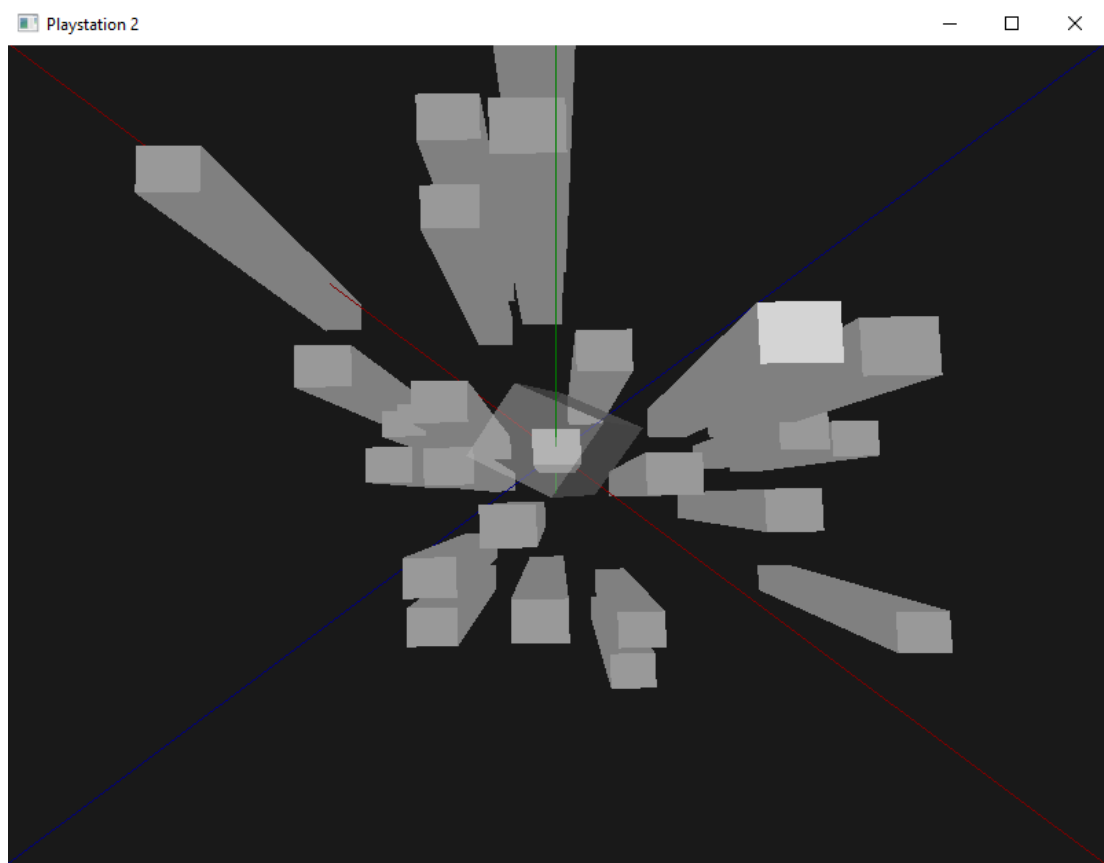
BAB III

HASIL

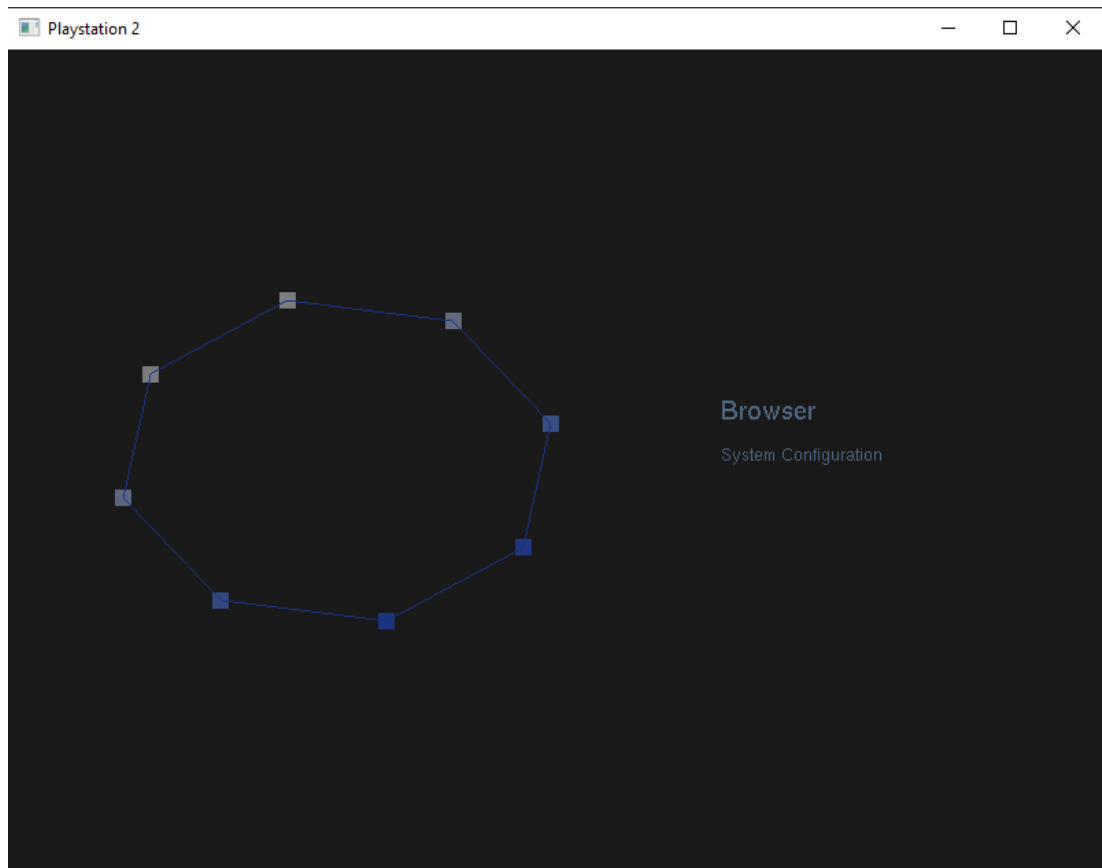
3.1 Source Code

https://github.com/binachuu/kelompok_6

3.2 Output



Gambar 1 Scene 1



Gambar 2 Scene 2

3.3 Penjelasan

Output dari source code yang dihasilkan adalah loading screen PS2. Terdapat transisi menggunakan scene di mana scene pertama yang berupa 3D beralih ke scene kedua yang berupa 2D. Terdapat juga pergerakan kamera pada scene pertama, yang mendasari peralihan ke scene kedua, serta fungsi keyboard untuk menyembunyikan atau menampilkan garis kartesius yang telah dibuat.

BAB IV

PENUTUP

4.1 Kesimpulan

Projek ini berhasil merealisasikan pembuatan loading screen PlayStation 2 menggunakan OpenGL, sebuah teknologi yang mendukung pembuatan grafis 2D dan 3D dengan performa tinggi dan fleksibilitas platform. Dalam prosesnya, berbagai konsep dasar grafika komputer diterapkan, seperti transformasi translasi dan rotasi, proyeksi tiga dimensi, serta pencahayaan yang realistis. Selain itu, konsep scene dimanfaatkan untuk menciptakan transisi dari mode 3D ke mode 2D yang memperkaya pengalaman visual.

Penggunaan scene pertama menampilkan objek kubus 3D dengan berbagai pergerakan kamera, memberikan efek perspektif yang dinamis sebelum transisi ke scene kedua yang berupa tampilan 2D. Interaksi pengguna juga diperhatikan melalui implementasi fungsi keyboard, yang memungkinkan pengguna untuk mengontrol elemen visual, seperti menampilkan atau menyembunyikan garis kartesius.

Melalui implementasi ini, proyek tidak hanya memenuhi tujuan pembelajaran dalam memahami dasar-dasar OpenGL, tetapi juga memberikan gambaran bagaimana elemen-elemen grafis dapat digunakan untuk menciptakan pengalaman yang menarik. Dengan demikian, hasil ini membuktikan potensi besar OpenGL sebagai alat untuk mengembangkan aplikasi grafis yang interaktif dan estetik.

DAFTAR PUSTAKA

- [1] Silicon Graphics, "The OpenGL Graphics System: A Specification (Version 4.6)," 2017.
- [2] M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, 9th ed., Addison-Wesley, 2018.
- [3] Khronos Group, "Khronos OpenGL Registry," 2021. [Online]. Available: <https://www.khronos.org/registry/OpenGL/>
- [4] T. Kessenich, D. Baldwin, and J. Spitzer, *The OpenGL Shading Language*, 3rd ed., Addison-Wesley, 2016.
- [5] Blender Foundation, "Blender Documentation," [Online]. Available: <https://docs.blender.org/>.