

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Komputer Grafis 3D (tiga dimensi) merupakan teknologi yang dipakai untuk pengembangan aplikasi berbasis 3D yang meliputi bidang yang cukup luas mulai dari industri film, pengembangan game, berbagai simulasi pelatihan bahkan sampai penggunaan *Computer-Aided Design* (CAD) untuk konstruksi bangunan dan tata kota. Banyaknya developer yang berminat mengembangkan aplikasi perangkat lunak berbasis 3D tersebut maka dikembangkanlah dukungan perangkat lunak dalam bentuk API (*Application Programming Interface*) Grafis 3D seperti Direct3D dan OpenGL. Selain itu juga adanya dukungan pengembangan perangkat keras berupa kartu grafis yang ikut membantu dalam perkembangan teknologi tersebut.

Walaupun telah didukung oleh API Grafis 3D, pengembangan aplikasi berbasis 3D oleh pihak developer ternyata memiliki banyak permasalahan. API Grafis 3D sejak awal memang dikembangkan sebagai antarmuka yang mewakili semua fitur perangkat keras sehingga menjadi terlalu kompleks bagi pihak developer dalam menggunakannya terutama dalam pengembangan program aplikasi berskala kecil maupun juga yang besar. Permasalahan lain yang timbul adalah kurangnya dukungan algoritma-algoritma pendukung dalam pemrograman 3D dan fitur matematika 3D pada API sehingga pihak developer terpaksa harus mengembangkan sendiri fitur tersebut. Dan tidak adanya manajemen untuk mengelola sumber daya yang saling terintegrasi dalam API Grafis 3D.

*Framework* adalah kerangka kerja. Juga dapat diartikan sebagai sekumpulan komponen pemrograman (terutama *class* dan *function*) yang siap *re-use* yang membantu developer dalam menangani masalah-masalah pemrograman, sehingga developer lebih fokus dan lebih cepat dalam membangun aplikasi. Sedangkan *Engine* merupakan sistem yang didesain untuk membantu pengembangan aplikasi dengan melakukan fungsionalitas tertentu, menjalankan fungsi-fungsi mulai dari tingkat rendah seperti komunikasi dengan antarmuka, dan mengelola sumber daya secara *realtime*.

Dengan adanya framework tersebut diharapkan memberikan kerangka kerja dalam pengembangan aplikasi berbasis 3D yaitu dalam bentuk komponen-komponen pemrograman untuk persiapan dan menjalankan proses rendering, modul untuk algoritma-algoritma yang dibutuhkan dalam pemrograman 3D, fungsi-fungsi matematika 3D, dan serta modul-modul lainnya yang membantu proses rendering grafis 3D. Berlaku juga untuk engine diharapkan untuk mengelola sumber daya secara efektif dan efisien mulai dari manajemen memori dan *buffer*, *material*, *lighting*, *texturing*, dan file-file yang dibutuhkan agar proses rendering berjalan dengan lancar.

Agar hal tersebut dapat terwujud, maka perlu dibuat suatu sistem aplikasi framework rendering engine untuk pengembangan aplikasi berbasis 3D. Dengan adanya aplikasi tersebut pihak developer dapat menggunakan komponen-komponen yang tersedia yang memfokuskan pengembangan rendering grafis 3D pada aplikasinya sendiri dan bukan pada detail-detail yang ada di dalam antarmuka sistem rendering yang kompleks. Aplikasi framework rendering engine ini juga memberikan fitur-fitur algoritma dan matematika 3D yang kompatibel dan

terintegrasi sehingga pihak developer dengan leluasa menggunakan komponen tersebut agar aplikasinya menjadi semakin optimal. Dan aplikasi framework rendering engine juga menyediakan manajemen pengelolaan sumber daya sehingga pihak developer dapat dengan mudah mengatur sumber daya rendering tersebut secara realtime dan terintegrasi.

Aplikasi framework rendering engine ini memiliki kelebihan dengan adanya fasilitas fitur teknik operasi perhitungan matematika cepat yaitu SIMD (*Single Instruction Multiple Data*) sehingga dapat meningkatkan kinerja dari proses rendering engine. Kelebihan lainnya yaitu adanya algoritma seperti BSP tree, Octree, Quaternion, dan Euler Angle sehingga pengembangan rendering grafis 3D diharapkan bisa menjadi semaksimal mungkin dan mencakup area seluas mungkin.

## 1.2. Perumusan Masalah

Berdasarkan batasan masalah diatas dapat diambil kesimpulan bahwa permasalahan yang ada adalah:

1. Bagaimana merancang dan membangun sistem rendering grafis 3D yang terpisah dari perhitungan matematika dan algoritma dalam membantu memudahkan pengembangan aplikasi berbasis 3D bagi developer, dengan fungsi-fungsi antara lain mengelola perangkat rendering (*render device*), manajemen verteks (*vertex cache manager*) dan serta manajemen skin (*skin manager*) secara efektif dan efisien?
2. Bagaimana merancang dan membangun sistem perhitungan matematika dan algoritma-algoritma 3D yang terintegrasi dan kompatibel menggunakan

teknologi SIMD untuk membantu kelancaran proses rendering grafis 3D yang sesuai dengan harapan pihak developer?

3. Bagaimana merancang dan membangun sistem untuk manajemen *scene*, kamera, kontrol pergerakan dan juga beserta informasi-informasi tentang proses rendering grafis 3D?

### 1.3. Pembatasan Masalah

Batasan masalah yang dilakukan adalah sebagai berikut:

1. Aplikasi yang dibuat meliputi pengembangan rendering grafis 3D yang memakai API DirectX.
2. Aplikasi ini menggunakan algoritma dan perhitungan matematika 3D yaitu meliputi Vektor, Matriks, *Ray*, *Plane*, *Polygon*, *Polylist*, *Euler Angle*, *Quaternion*, AABB, OBB, *Octree* dan *BSP Tree*.
3. Aplikasi ini terbatas hanya dapat digunakan untuk membantu pengembangan aplikasi berbasis 3D dalam rendering grafis 3D.

### 1.4. Tujuan

Dengan batasan masalah diatas maka tujuan yang akan dicapai dalam penyusunan tugas akhir ini adalah:

1. Merancang dan membuat suatu aplikasi rendering grafis 3D untuk membantu pengembangan aplikasi berbasis 3D, yang berfungsi mengelola perangkat rendering, manajemen verteks dan serta manajemen skin secara efektif dan efisien

2. Merancang dan membuat suatu aplikasi sistem perhitungan matematika dan algoritma 3D menggunakan teknologi SIMD yang terintegrasi dan kompatibel dengan sistem rendering grafis 3D.
3. Merancang dan membuat suatu aplikasi untuk manajemen scene, kamera, kontrol pergerakan dan menampilkan informasi yang dibutuhkan dalam proses rendering grafis 3D.

### 1.5. Sistematika Penulisan

Laporan Tugas Akhir (TA) ini ditulis dengan sistematika penulisan sebagai berikut:

#### **BAB I        PENDAHULUAN**

Pada bab pertama ini akan dijelaskan mengenai latar belakang masalah dan penjelasan masalah secara umum, perumusan masalah serta batasan masalah yang dibuat, tujuan dari pembuatan tugas akhir dan sistematika penulisan dari buku ini.

#### **BAB II        LANDASAN TEORI**

Pada bab kedua ini berisi tentang pembahasan landasan teori yang berhubungan dan mendukung dalam pembuatan tugas akhir. Landasan teori yang digunakan antara lain: *software framework*, komputer grafis 3D, Direct3D API, matematika 3D, vektor, sistem koordinat kartesius, matriks dan transformasi, orientasi *euler angle*, *plane* dan *ray*, dan *bounding volume*.

#### **BAB III       PERANCANGAN SISTEM**

Bab ini berisi penjelasan mengenai analisa identifikasi permasalahan, algoritma manajemen *scene*, perancangan sistem

*class* diagram, *component* diagram, dan *flowchart* fungsi rendering.

#### **BAB IV IMPLEMENTASI DAN EVALUASI**

Bab keempat ini berisi tentang implementasi dari framework rendering engine. Serta melakukan pengujian terhadap sistem rendering, pengujian implementasi manajemen scene, dan evaluasi sistem untuk mengetahui bahwa aplikasi tersebut telah dapat menyelesaikan permasalahan yang dihadapi sesuai dengan harapan.

#### **BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran. Saran yang dimaksud adalah saran terhadap kekurangan dari aplikasi yang ada kepada pihak lain yang ingin meneruskan topik TA ini. Tujuannya adalah agar pihak lain tersebut dapat menyempurnakan aplikasi sehingga bisa menjadi lebih baik dan berguna.