

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di bidang informasi meningkat dari tahun ke tahun sesuai dengan perkembangan jaman. Dengan adanya perkembangan teknologi informasi ini mendorong manusia untuk memanfaatkannya dalam berbagai bidang. Secara langsung atau tidak langsung, teknologi informasi telah menjadi bagian penting dalam berbagai kehidupan. Salah satunya adalah kemudahan dalam aspek berbagi informasi atau data dari satu pengguna dengan pengguna lainnya. Komputer grafik adalah bagian dari komputer sains yang mengatasi proses menghasilkan gambar dengan bantuan komputer (**Eck, 2015**). Komputer grafik merupakan teknologi utama dalam bidang digital fotografi, perfilman, video game, perangkat seluler dan tampilan komputer. Semua aspek tersebut bertujuan untuk berbagi informasi atau data dan komunikasi dalam bentuk visual.

Menurut **Kard, dkk (1998)** dan **Qin (2008)** visualisasi adalah rekayasa dalam merepresentasikan data visual menggunakan komputer sebagai pendukung untuk memperkuat sebuah informasi. Visualisasi mengeksplor, mentransformasikan, dan mengamati data dalam bentuk gambar dan grafik. Data tersebut dapat berasal dari berbagai sumber dan digunakan untuk berbagai bidang. Contohnya visualisasi 3 dimensi pada data CT-scan, dan MRI yang sangat membantu dalam bidang medis, menampilkan model objek 3D dalam dunia perfilman atau memvisualisasikan objek geometri yang membantu dalam pemahaman di bidang akademik.

Dari berbagai bentuk geometri yang ada, *torus* adalah permukaan yang mempunyai 1 *genus* dengan *genus* adalah jumlah banyaknya lubang pada suatu permukaan. Sehingga *torus* merupakan bangun permukaan yang memiliki satu lubang. *Torus* pada ruang 3 dimensi berbentuk seperti donat, dan *torus* sangat lah bermanfaat untuk dimensi yang lebih tinggi. Salah satu penggunaan umum untuk *n-D torus* adalah Sistem Dinamikal. (**Tabor, 1989**).

Untuk memvisualisasikan suatu objek dalam komputer grafik diperlukan perangkat bantuan (*interface*) salah satunya ialah OpenGL (*OpenGraphic Library*). OpenGL (*Open Graphics Library*) biasa dianggap sebagai API (*Application Programming Interface*) yang menyediakan bermacam-macam fungsi yang dapat digunakan untuk memanipulasikan grafik dan gambar. Menurut **Cirne & Pedrini (2013)** teknik visualisasi dengan melakukan proyeksi dua dimensi pada volumetric dataset biasa disebut sebagai Volume Rendering. Volumetrik dataset tersebut biasa direpresentasikan oleh sekumpulan elemen volume yang disebut *voxel*. Masing-masing *voxel* memiliki nilai tertentu pada *grid* yang terletak didalam ruang tiga dimensi. Volume Rendering ini diklasifikasi menjadi dua kategori. *Direct Volume Rendering (DVR)* dan *Indirect Volume Rendering* atau *Surface Fitting (SF)*. *DVR* ditandai dengan pemetaan elemen langsung ke layar tanpa menggunakan geometri primitif sebagai representasi perantara, sedangkan *SF* ditandai dengan mengekstraksi dan merepresentasikan *isosurface* (permukaan yang mewakili sekumpulan titik dengan nilai scalar yang sama) yang kemudian dirender untuk di visualisasikan. Isosurface ini dapat ditentukan dari permukaan primitif seperti poligon atau dengan fungsi tertentu. Contoh algoritma untuk *DVR* adalah *Shear-Warp*, *3D Texture Mapping*, dan *Ray Casting*, sedangkan algoritma untuk *SF* yaitu *Contour Connection* dan *Marching Cube*.

Di dalam jurnalnya **Theisel (2002)** menyatakan bahwa metode yang terkenal dalam membangun *isosurface* dari volume dataset adalah algoritma *Marching Cube*. Volume Dataset adalah kumpulan data yang berisikan 3 dimensi *pixel grid* yang dipetakan sesuai dengan tingkat keabuan pada *pixel* (**Qin, 2008**). Menurut **Lorensen & Cline (1987)**, algoritma ini membangun *isosurface* dengan pendekatan “*divide and conquer*”, yaitu memecah-mecahkan masalah menjadi bagian-bagian kecil sehingga lebih mudah diselesaikan. Algoritma ini membentuk kubus kecil (*voxel*) sehingga diperoleh 8 *pixel* yang berasal dari dua irisan *grid* yang berdekatan, lalu dikalkulasikan pada setiap *pixel* dari kubus tersebut sehingga menghasilkan pendekatan segitiga yang terbentuk dari volume dataset dan dilanjutkan ke kubus selanjutnya. Algoritma ini membentuk permukaan tiga dimensi yang beresolusi tinggi dengan mempertimbangkan sifat dasar pada

interpolasi linier bidang skalar dan menemukan representasi permukaan yang lebih baik dengan pendekatan segitiga yang dihasilkan. Algoritma ini tidak hanya terkenal dan populer, menurut **Han (2005)** dibandingkan dengan pendekatan yang lain, hasil pendekatan dari algoritma *Marching Cube* dapat ditampilkan pada berbagai sistem grafik dan aplikasi pemodelan 3D dengan hanya dengan algoritma *rendering* yang standar dan menghasilkan permukaan pendekatan yang baik dengan model secara *real time*.

Berdasarkan uraian yang telah dijelaskan, penulis tertarik untuk melakukan Visualisasi 3D pada Objek *Torus* dengan menggunakan Algoritma *Marching Cube*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, permasalahan yang akan dibahas adalah sebagai berikut :

1. Bagaimana memvisualisasikan Objek 3D dengan menggunakan Algoritma *Marching Cube*?
2. Bagaimana membuat program Visualisasi 3D dengan menggunakan Algoritma *Marching Cube*?
3. Bagaimana implementasi program Visualisasi 3D dengan Algoritma *Marching Cube* pada Objek *Torus* ?

1.3 Tujuan

Berdasarkan rumusan masalah yang diatas, tujuan yang akan dicapai adalah sebagai berikut:

1. Menerapkan Algoritma *Marching Cube* dalam memvisualisasikan Objek 3D.
2. Membuat program Visualisasi 3D dengan menggunakan Algoritma *Marching Cube*.
3. Mengimplementasi program Visualisasi 3D dengan menggunakan algoritma *Marching Cube* pada Objek *Torus*.

1.4 Manfaat

Beberapa manfaat yang diharapkan dapat diperoleh dari penelitian ini adalah:

1. Menambah pengetahuan dan wawasan mahasiswa tentang penerapan konsep algoritma *Marching cube* dalam memvisualisasikan objek 3 dimensi *torus*.
2. Sebagai referensi dalam menerapkan algoritma lain untuk visualisasi objek 3 dimensi