

BAB I

PENDAHULUAN

1.1. Latar Belakang

Permainan merupakan salah satu media bagi seseorang untuk melepaskan rasa penat atau kejenuhan. Dimana permainan terdiri dari 2 jenis yaitu *offline* dan *online*. Namun pada kenyataannya, permainan *online* atau permainan web lebih diminati dan bahkan dipelajari [1]. Hal ini dikarenakan, permainan *online* dapat dimainkan dimana saja selama terdapat jaringan internet. Selain itu pemain juga dapat berinteraksi dengan pemain lain sehingga membuat permainan tersebut lebih menarik [2].

Permainan web yang sering dijumpai umumnya menggunakan HTML5 dan berbasis WebGL (*Web-based Graphics Library*) dengan beberapa penggunaan *framework* yang berbeda [3]. WebGL merupakan *Platform Application Programming Interfaces* (APIs) yang memungkinkan *browser* internet untuk membuat tampilan 3D dengan efisien dan tanpa plug-in.

Pada penelitian ini, dibuat sebuah permainan web tentang permainan *maze*. *Maze*, atau dalam bahasa Indonesia disebut labirin, merupakan sebuah tempat yang penuh dengan jalan dan lorong yang berliku-liku dan simpang siur [4]. Pengertiannya dalam permainan ini, *maze* adalah sebuah dunia yang memiliki sebuah ruang dan pagar yang saling terhubung satu sama lain. Sehingga pemain yang berada dalam dunia tersebut harus membebaskan diri agar mencapai jalan keluar. Permainan ini dikembangkan dengan mengimplementasikan algoritma *maze generation* terbaik dari pilihan-pilihan algoritma yang dibandingkan. Algoritma yang dibandingkan adalah *Binary Tree*, *Aldous-Broder*, *Hunt-and-Kill*, *Growing Tree*, dan *Wilson's*. Pemilihan algoritma ini didasarkan pada jenis *maze* yang dihasilkan dan proses yang unik pada masing-masing algoritma.

Algoritma *Binary Tree* merupakan algoritma yang paling mudah dan sederhana namun menghasilkan *maze* yang sempurna [5]. *Binary Tree* disebut sebagai algoritma termudah karena hanya membutuhkan dua pilihan kemungkinan pada setiap langkah. Pada setiap langkah, hanya perlu melakukan sebuah pilihan acak untuk



menimpa sebuah pagar yang berada pada sebelah kanan/kiri dan bawah/atas.

Selanjutnya, Algoritma *Aldous-Broder* adalah algoritma pembuat *maze* yang dikembangkan oleh dua orang bersaudara yaitu, David Aldous dan Andrei Broder [5]. David Aldous adalah seorang profesor di *University of California*, sedangkan Andrei Broder adalah seorang ilmuwan yang terkenal di Google. Penerapan algoritma ini hampir semudah algoritma *Binary Tree*, namun membutuhkan waktu yang lama untuk menyelesaikan proses sampai akhir [5]. Cara kerja dari algoritma ini adalah memulai proses pada sel mana saja, kemudian pilih tetangga secara acak. Jika tetangga yang dipilih belum pernah dikunjungi maka pagar yang berada pada sel sekarang dan sel tetangga akan ditimpa menjadi sebuah ruang. Jika tetangga yang terpilih sudah pernah dikunjungi, maka tidak akan terjadi apapun. Proses ini berlanjut terus sampai seluruh ruang telah dikunjungi.

Algoritma selanjutnya yang dibandingkan adalah *Hunt-and-Kill*. Algoritma ini memiliki kemiripan seperti algoritma *Aldous-Broder* [5]. Proses algoritma ini dimulai dari penentuan sel awal secara acak. Kemudian, dilakukan sebuah perjalanan acak yang dimulai dari sel awal tadi. Perbedaan antara *Aldous-Broder* dan *Hunt-and-Kill* adalah *Hunt-and-Kill* tidak diperbolehkan mengunjungi sel tetangga yang telah dikunjungi sedangkan *Aldous-Broder* diperbolehkan. Selanjutnya, apabila sel selanjutnya yang akan dilalui sudah pernah dikunjungi, maka akan dilakukan proses yang bernama *hunt* (berburu). Dalam proses *hunt*, dilakukan sebuah pengecekan pada setiap sel sampai ditemukan sebuah sel yang belum pernah dikunjungi yang memiliki tetangga yang telah dikunjungi. Kemudian ulangi proses perjalanan acak. Ulangi kedua proses tersebut sampai seluruh sel telah dikunjungi.

Algoritma *Growing Tree* membutuhkan sebuah *list* dinamis untuk menyimpan posisi sel yang sedang diproses [5]. Langkah awal algoritma ini adalah memilih satu sel secara acak. Masukkan sel tersebut ke dalam *list*. Pilih salah satu sel dari *list*. Ambil secara acak tetangga dari sel tersebut, lalu hubungkan keduanya dan tambahkan tetangga ke dalam *list*. Apabila sel saat ini tidak memiliki tetangga yang belum dikunjungi, maka keluarkan sel tersebut dari *list*. Ulangi semua langkah sampai isi dari *list* kosong.

Algoritma terakhir yang dibandingkan adalah algoritma *Wilson's*. Algoritma ini dikembangkan oleh David Bruce Wilson, seorang peneliti di Microsoft dan profesor bidang matematika di *University of Washington*. Seperti algoritma *Aldous-Broder*, algoritma ini bergantung pada penelusuran jalan secara acak dan dapat memutar balik [5]. Algoritma ini diawali dengan pemilihan satu sel secara acak dan ditandai sebagai telah dikunjungi. Kemudian, pilih satu sel yang belum dikunjungi secara acak. Dari sel terpilih yang belum dikunjungi tersebut, dilakukan penelusuran jalan secara acak sampai menemukan sel yang telah dikunjungi. Setelah menemukan sel sudah dikunjungi, dari jalur awal sampai akhir jadikan sebuah ruang dan tandai sebagai sudah dikunjungi. Proses ini berlanjut sampai tidak ditemukan sel yang belum dikunjungi.

Selain itu, permainan ini juga dibangun menggunakan WebGL untuk pembuatan objek berbasis web. Penyimpanan *cloud* untuk menyimpan akun *player*, *score*, dan *result game*. Ditambah permainan ini dapat dimainkan secara *multiplayer*. Dari penelitian-penelitian saat ini yang pernah dilakukan, pembuatan permainan serupa dengan menggunakan kombinasi penyimpanan *cloud*, WebGL, implementasi *maze generation*, dan *multiplayer* belum pernah dibuat. Dengan pembuatan permainan web *online Maze Survival* ini diharapkan dapat menjadi salah satu solusi untuk permasalahan yang terjadi.

1.2. Perumusan Masalah

Dari latar belakang yang sudah dijelaskan sebelumnya dimunculkan sebuah masalah antara lain:

1. Seberapa cepat algoritma *Binary Tree*, *Aldous-Broder*, *Hunt-and-Kill*, *Growing Tree*, dan *Wilson's* dapat menghasilkan sebuah lintasan *maze*?
2. Seberapa efisien permainan *online Maze Survival* berbasis WebGL dengan penerapan algoritma *Maze Generation* terbaik disimpan pada penyimpanan *cloud* dibandingkan dengan penyimpanan lokal?



1.3. Tujuan

Berdasarkan perumusan masalah di atas, maka tujuan yang ingin dicapai adalah, sebagai berikut:

1. Mengetahui kecepatan masing-masing dari algoritma *Binary Tree*, *Aldous-Broder*, *Hunt-and-Kill*, *Growing Tree*, dan *Wilson's*.
2. Mengetahui tingkat efisien penyimpanan *cloud* dibandingkan penyimpanan lokal pada permainan *online Maze Survival* berbasis WebGL dengan penerapan algoritma *Maze Generation* terbaik.

1.4. Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah dihasilkan permainan *maze* berbasis WebGL dengan penyimpanan *cloud* bernama *Maze Survival* dengan penerapan algoritma terbaik pada perbandingan *Maze Generation*.

1.5. Batasan Masalah

Batasan masalah untuk penelitian ini, antara lain:

1. Algoritma yang dibandingkan adalah *Binary Tree*, *Aldous-Broder*, *Hunt-and-Kill*, *Growing Tree*, dan *Wilson's*.
2. Aplikasi ini dibuat dengan mengimplementasikan algoritma terbaik dari hasil perbandingan.
3. Hanya terdapat maksimal 10 pemain dalam satu room.
4. Dibangun menggunakan *framework* p5.js berbasis WebGL.
5. Penyimpanan menggunakan *Google cloud*.
6. Menggunakan WebSocket untuk interaksi antar user.

1.6. Metodologi Penelitian

Penelitian ini menerapkan jenis metode penelitian bersifat eksperimen. Metode eksperimen yang diterapkan berdasarkan pada langkah-langkah berikut:

- a. Tahap pertama (Observasi)
Pada tahap ini, dilakukan pengamatan dan pengidentifikasian masalah-masalah yang ada sehingga terjadi perubahan.
- b. Tahap kedua (Perancangan sistem)
Pada tahap ini diketahui permasalahan yang ada kemudian dilanjutkan dengan pembuatan rencana tindakan untuk

menyelesaikan permasalahan dengan tepat. Tindakan yang dilakukan dengan membuat perancangan dan penerapan perbandingan algoritma *maze generation* pada pembuatan permainan *Maze Survival*.

- c. Tahap ketiga (Implementasi sistem)
Pada tahap ini, diterapkan implementasi dari rancangan sistem yang telah dibuat.
- d. Tahap keempat (Uji coba dan evaluasi)
Pada tahap ini, dilakukan pengujian dan evaluasi dari hasil yang didapat dari perbandingan algoritma *maze generation* pada pembuatan permainan *Maze Survival*.
- e. Tahap kelima (Penyusunan Laporan)
Pada tahap ini dilakukan penulisan tahap demi tahap mengenai hasil yang didapat dari observasi hingga pengujian dari perbandingan algoritma *maze generation* pada pembuatan permainan *Maze Survival*.

1.7. Sistematika Penulisan

Secara garis besar, sistematika penulisan buku dalam skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang penelitian, perumusan masalah, tujuan, manfaat, batasan masalah, metodologi penelitian serta sistematika penulisan dalam laporan skripsi ini.

BAB II : STUDI PUSTAKA

Bab ini membahas mengenai landasan teori pokok yang digunakan serta penelitian-penelitian sebelumnya.

BAB III : PERANCANGAN SISTEM

Bab ini menjelaskan tentang rancangan sistem dan keseluruhan sistem yang dibuat beserta rancangan alur pada tiap-tiap proses pada sistem.

BAB IV : IMPLEMENTASI SISTEM

Bab ini menjelaskan tentang implementasi sistem berdasarkan rancangan pada bab sebelumnya dan menjelaskan hasil analisa dari sistem yang dibuat.



BAB V : PENUTUP

Bab ini menjelaskan tentang kesimpulan dari uji coba yang telah dilakukan serta saran yang menjadi masukan dalam pengembangan penelitian berikutnya.

