

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini sudah banyak sekali jenis *game puzzle* yang asik untuk dimainkan yang juga sekaligus akan mengasah kemampuan berpikir dalam menyelesaikan suatu permasalahan. Salah satu *game puzzle* yang menarik adalah permasalahan pada *game* labirin (*maze*). Menurut Octara Pribadi (2015), labirin merupakan *puzzle* yang terdiri dari jalur yang rumit, berliku-liku, serta memiliki banyak jalan buntu, dimana penyelesaian *game* tersebut harus mencari jalan keluar dari labirin.

Tetapi masih sering ditemukan permainan labirin yang jalurnya tetap sama setiap kali *game* mulai dimainkan, khususnya pada level yang sama dikarenakan jalur labirin memang sudah dibuat *static* pada setiap level permainan tersebut. Akibatnya pemain yang memainkan permainan akan menemukan kerumitan labirin yang sama sehingga akan membuat pemain cepat merasa bosan dan mengurangi rasa ingin tahu untuk menelusuri labirin itu kembali.

Supaya setiap labirin mempunyai bentuk yang tidak sama persis setiap kali permainan labirin dimainkan, maka diperlukan penerapan algoritma pembangkit labirin atau yang biasa dikenal dengan sebutan *maze generator*. Sehingga setiap kali pemain memulai permainan labirin, permainan akan secara otomatis mengacak kembali bentuk labirin walaupun pemain kemungkinan akan menemui labirin yang sama.

Untuk menghasilkan bentuk labirin yang berbeda-beda dibutuhkan penerapan *maze generator* sebagai pembangkit labirin. Beberapa algoritma yang dapat digunakan yaitu algoritma *Depth-First Search* (DFS), algoritma Kruskal, dan algoritma Prim. Berdasarkan penelitian sebelumnya yang dilakukan oleh Octara Pribadi (2015), algoritma DFS dapat menghasilkan labirin *perfect*, yaitu labirin yang tidak memiliki jalan yang berulang (*loop*) dan tidak ada sel yang terisolasi didalam labirin. Sedangkan algoritma Kruskal dan algoritma Prim menghasilkan bentuk labirin yang mirip, cenderung menghasilkan banyak cabang yang pendek dan buntu. Dalam penelitian yang dilakukan oleh Meiki Kurniawam (2015) juga menggunakan algoritma DFS sebagai *maze generator* agar pemain tidak terlalu banyak memasuki cabang yang pendek dan buntu yang akan menyulitkan pemain dalam melakukan penelusuran balik pada percabangan sebelumnya untuk mencari jalan keluar yang benar.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka rumusan masalah yang didapat adalah bagaimana menghasilkan bentuk labirin yang tidak statis dengan mengimplementasikan algoritma *Depth-First Search* (DFS) sebagai *maze generator* dalam 3 tingkat kompleksitas yang berbeda.

1.3 Ruang Lingkup

Adapun ruang lingkup permasalahan dalam penelitian ini, yaitu:

1. Menggunakan algoritma *Depth-First Search* untuk membangkitkan peta labirin.

2. Terdapat 3 level kompleksitas labirin, yaitu :
 - a. Sempel dengan ukuran labirin 8x8
 - b. Sedang dengan ukuran labirin 12x12
 - c. Kompleks dengan ukuran labirin 16x16
3. Proses *generate* labirin dilakukan sebelum permainan memasuki *scene*.
4. Menganalisis *resource* seperti CPU terpakai, GC Alloc dan waktu yang dibutuhkan oleh *device* pada saat proses *generate* peta labirin.
5. Analisis data *resource* dilakukan dengan menggunakan *Unity Profiler*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menghasilkan bentuk labirin yang bersifat dinamis dengan memiliki jalur keluar yang benar, serta menganalisis *resource* yang terpakai pada saat proses *generate* labirin berdasarkan level kompleksitas.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah bentuk labirin yang dihasilkan bersifat dinamis, setiap kali permainan dimulai ulang maka bentuk labirin akan berubah, sehingga user tidak merasa bosan dan lebih tertantang dengan bentuk labirin yang berbeda-beda.