

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penerapan algoritma Depth-First Search (DFS) pada maze generator sebelumnya telah dilakukan oleh Meiki Kurniawan (2015), dan Octara Pribadi (2015) untuk mengetahui apakah algoritma DFS ini berhasil membuat bentuk labirin. Dan hasil yang didapatkan oleh kedua penelitian tersebut bahwa algoritma DFS ternyata berhasil membentuk labirin yang sifatnya sempurna (*perfect maze*), yaitu labirin dengan satu jalur tanpa ada jalur yang terhubung, dan tidak ada sel yang terisolasi di dalam labirin.

Kemudian algoritma DFS ini juga pernah digunakan untuk membuat game tebak kata oleh Salamuddin (2016). Dimana algoritma ini ternyata berhasil membantu dalam menjawab soal yang diberikan. Banyaknya huruf yang diberikan mempengaruhi kecepatan algoritma DFS dalam mencari jawaban.

Sedangkan Albin Karlsson (2018) melakukan penelitian tentang kompleksitas labirin yang dihasilkan dari *maze generator* menggunakan algoritma *Recursive Backtracking*, *Recursive Division*, dan algoritma Prim. Komponen yang diteliti dalam labirin meliputi distribusi jalur percabangan, distribusi koridor, dan panjang jalur penyelesaian. Penelitian ini menghasilkan bahwa algoritma *Recursive Backtracking* merupakan algoritma yang membentuk labirin paling kompleks diantara algoritma lainnya.

Tabel 2.1 Tinjauan Pustaka

No	Sumber	Objek	Kasus	Algoritma	Bahasa Pemrograman / Teknologi
1.	Salamuddin (2016)	Game tebak kata	Membuat game tebak kata	DFS	Java, Android Studio
2.	Meiki Kurniawan (2015)	Maze generator	Membangun jalur labirin	DFS	Csharp, Unity 3D
3.	Octara Pribadi (2015)	Maze generator	Membangun jalur labirin	DFS, BFS	C++
4.	Albin Karlsson (2018)	Kompleksitas labirin	Membanding kan kompleksitas dalam penerapan algoritma	Algoritma Prim, Recursive Backtracking	Csharp, Unity 3D
5.	Muhammad Aprizal Abyan (diusulkan)	Maze generator	Analisis resource dalam proses maze generation	DFS	Csharp, Unity 3D

2.2 Dasar Teori

2.2.1 Pengertian Game

Berikut adalah sebagian definisi dan pengertian dari *game* menurut beberapa ahli :

- a. Menurut John von Neumann (1953)

Game merupakan Permainan yang terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun dengan untuk memaksimalkan kemampuan sendiri atau pun meminimalkan kemenangan lawan.

b. Menurut Agustinus Nilwan (1995)

Game merupakan permainan komputer yang dibuat dengan teknik dan metode animasi.

c. Menurut Ivan C Siberio (2010)

Game merupakan salah satu aplikasi yang paling banyak dipakai dan dinikmati oleh pengguna media elektronik saat ini.

d. Menurut Samuel Henry (2013)

Game adalah bagian yang tidak bisa dipisahkan dari keseharian anak-anak.

2.2.2 Puzzle Game

Game puzzle merupakan *genre game* yang menekankan pada pemecahan teka-teki. Jenis-jenis teka-teki dapat menguji banyak keterampilan pemecahan masalah termasuk logika, pengenalan pola, pemecahan urutan, dan penyelesaian kata. Pemain mungkin memiliki waktu yang tidak terbatas atau upaya tak terbatas untuk memecahkan teka-teki, atau mungkin ada batas waktu, atau teka-teki yang lebih sederhana dapat menjadi sulit dengan harus menyelesaikannya secara *real time*, seperti dalam Tetris. (Cahyo 2011)

2.2.3 Labirin

Kata “labirin” memaksudkan tempat yang penuh dengan jalan dan lorong berliku-liku serta simpang siur. Jalur ini biasanya mengarah ke bagian tengah labirin atau ke luar dari pintu lain. Sistem lorong, halaman, kamar, dan serambi Labirin sedemikian rumit dan membingungkan sehingga seorang asing tidak akan pernah menemukan jalan masuk atau keluar tanpa bantuan pemandu. Sebagian dari Labirin itu gelap gulita, dan konon ada pintu-pintunya yang, sewaktu dibuka, mengeluarkan bunyi yang sangat menakutkan bagaikan guntur. (Sedarlah, 1999)



Gambar 2.1 "Classic" atau "Cretan" Labirin

(Sumber : <http://www.labyrinthcreations.com/labyrinth-history.html>)

2.2.4 Maze Generator

Maze Generator juga disebut sebagai pembangkit labirin, yaitu metode yang digunakan untuk membangun bentuk labirin. Labirin dapat dihasilkan dengan memulai susunan sel yang telah ditentukan sebelumnya (paling umum kotak persegi panjang tetapi pengaturan lain juga dimungkinkan) dengan dinding di antara mereka. Susunan yang telah

ditentukan ini dapat dianggap sebagai grafik yang terhubung dengan ujung-ujungnya mewakili situs dinding yang mungkin dan simpul yang mewakili sel. Tujuan dari algoritma pembuatan labirin kemudian dapat dianggap membuat subgraf di mana sulit untuk menemukan rute antara dua node tertentu. (Bond 1981)

Beberapa algoritma yang digunakan sebagai *maze generator* adalah algoritma *Depth-First Search* (DFS), algoritma Kruskal, algoritma Prim, algoritma Wilson, dan algoritma Recursive Division.

2.2.5 Algoritma Depth First Search (DFS)

Menurut Hafid Inggiantowi (2008) DFS adalah pencarian yang berjalan dengan meluaskan anak akar pertama dari pohon pencarian yang dipilih dan berjalan dalam dan lebih dalam lagi sampai simpul tujuan ditemukan, atau sampai menemukan simpul yang tidak punya anak. Kemudian, pencarian *backtracking* akan kembali ke simpul yang belum selesai ditelusuri.

Berikut merupakan beberapa hal tentang DFS menurut Octara Pribadi (2015) yaitu :

1. Traversal (penjalaran simpul) dari suatu graf G akan :
 - a. Mengunjungi semua simpul dan sisi dari G
 - b. Menentukan apakah G terhubung
 - c. Memperhitungkan komponen terhubung dari G
 - d. Memperhitungkan pohon merentang dari G

kemudian DFS akan melakukan *backtrack* kembali ke simpul atas dan seterusnya hingga kembali lagi ke simpul paling atas. (Pribadi 2015)

Menurut Teneng (2008) pembangkitan labirin menggunakan DFS yaitu dengan menghitung jumlah total sel pada baris dan kolom yang dilewati oleh garis atau jalur solusi. Jumlah sel yang dilewati pada baris dan kolom itulah yang nantinya akan menjadi nilai-nilai baris dan kolom sebagai panduan pada bentuk labirin. Labirin sendiri akan dibentuk dengan menghilangkan tembok atau *wall* pada *maze* sempurna sehingga akan kembali berbentuk seperti grid biasa, hanya saja sudah disediakan pintu masuk dan pintu keluar serta nilai-nilai pada baris dan kolom sebagai panduan untuk menemukan jalur solusi. Algoritma dalam pembangkitan labirin menggunakan DFS sebagai berikut :

- a. Mulai
- b. Masukkan ukuran *grid* ($i \times j$)
- c. Selama Total sel pada *grid* $<$ sel yang telah di kunjungi lakukan langkah d, e, dan f
- d. Pilih secara acak sel tetangga (kanan, kiri , atas, bawah)
- e. Cek apakah sel sudah pernah dikunjungi, jika ya kembali ke langkah d, jika tidak lakukan langkah f
- f. Hapus *wall* atau pembatas pada sel
- g. Cek apakah semua sel sudah dikunjungi, jika ya lakukan langkah h, jika tidak kembali ke langkah d
- h. Selesai

2.2.6 Unity Profiler

Unity Profiler adalah alat yang dapat digunakan untuk mendapatkan informasi kinerja tentang aplikasi. *Unity Profiler* dapat dihubungkan ke perangkat di jaringan Anda atau perangkat yang terhubung ke *Unity Engine* Anda untuk menguji bagaimana aplikasi Anda berjalan pada platform rilis yang Anda inginkan. Anda juga dapat menjalankannya di Editor untuk mendapatkan ikhtisar alokasi sumber daya saat Anda sedang mengembangkan aplikasi Anda. (Unity 2019)

Profiler mengumpulkan dan menampilkan data tentang kinerja aplikasi Anda di berbagai bidang seperti CPU, memori, renderer, dan audio. Ini adalah alat yang berguna untuk mengidentifikasi area untuk peningkatan kinerja dalam aplikasi Anda, dan beralih ke area tersebut. Anda dapat menentukan hal-hal seperti bagaimana script Anda, aset, pengaturan *scene*, kamera rendering, dan pengaturan yang memengaruhi kinerja aplikasi Anda. Ini menampilkan hasil dalam serangkaian grafik, sehingga Anda dapat memvisualisasikan di mana lonjakan kinerja aplikasi Anda terjadi. (Unity 2019)