

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi di abad-21 sangatlah pesat, teknologi yang ada terus berkembang dan menciptakan berbagai inovasi baru yang semakin modern. Teknologi ini diciptakan oleh manusia dengan tujuan yaitu untuk mempermudah kehidupan manusia. Berbagai manfaat dari teknologi yaitu sarana komunikasi, sarana hiburan, sarana memperoleh informasi [4]. Sarana hiburan pada teknologi salah satu contohnya yaitu sebuah *video game*.

*Video game* mengalami perkembangan yang sangat cepat. Kini *video game* hadir dengan berbagai jenis genre ataupun jenis seperti *action*, *fighting*, *racing*, *education*, *adventure*, *maze*, *strategy*, *simulation*, *puzzle*, dan masih banyak lagi. Game dengan genre *maze* merupakan sebuah tempat yang memiliki jalur yang berbelok-belok dan bercabang serta dalam jalurnya sering memiliki jalur buntu. Game *maze* ini juga biasanya memiliki satu titik awal mulai dengan satu tujuan [5] serta dalam *maze* biasanya memiliki NPC (*Non Player Character*) yang menghalang pemain dalam mencapai tujuan tersebut, namun salah satu hal yang menantang dalam pembuatan game *maze* adalah membuat sebuah pergerakan NPC agar dapat mengejar pemain tanpa terjebak dalam game *maze* itu sendiri. Oleh karena itu, solusi untuk masalah ini adalah mengimplementasikan *pathfinding* kedalam NPC game.

*Pathfinding* merupakan sebuah algoritma yang dirancang untuk menemukan dua node terpedek yang sudah ditentukan dengan cara memulai pada node awal dan kemudian berlanjut hingga menuju node akhir secara berulang ulang untuk mendapatkan jarak antara kedua node ini [6]. Salah satu algoritma yang banyak digunakan dalam *pathfinding* merupakan algoritma A-star karena algoritma ini dapat berjalan lebih baik dari algoritma pencarian heuristik lainnya seperti Greedy Best First Search dan IDA\* [7].

Namun walaupun algoritma A-star banyak dijadikan bahan penelitian *pathfinding*, dalam Unity juga terdapat sebuah sistem yaitu NavMesh (*Navigation Mesh*) yang dapat dipakai untuk melakukan *pathfinding*. NavMesh merupakan sebuah representasi pencarian ruang yang menggunakan ruang dua dimensi (2D) untuk merepresentasikan jalur yang bisa dilewati (*walkable*) [8]. Dengan adanya

fitur NavMesh pengguna Unity dapat mengimplementasikan *pathfinding* dengan lebih mudah.

Terdapat beberapa penelitian yang menjadi dasar untuk penelitian ini, yaitu penelitian Penerapan Algoritma A-star pada Game Labirin [9] dan Penerapan Navmesh Dengan Algoritma A Star Pathfinding Pada Game Edukasi 3d Go Green [10]. Untuk penelitian ini, terdapat perbedaan dengan penelitian lainnya yaitu pada peneliti sebelumnya menggunakan A-Star serta NavMesh dalam implementasi gamenya serta mencari nilai kepuasan sedangkan penelitian ini akan dilakukan untuk mengimplementasi serta membandingkan antara algoritma A-star dengan NavMesh pada Unity dalam hal kecepatan *pathfinding* dalam *Maze*. *Maze* digunakan dapat penelitian ini karena pada dasarnya *Maze* memiliki banyak jalur serta cukup rumit sehingga dengan menggunakan *pathfinding*, menyelesaikan *maze* akan menjadi lebih praktis [11]. Perbandingan ini dilakukan untuk mencari tahu apakah NavMesh yang biasa dipakai pada Unity dapat lebih cepat atau lebih lambat dari algoritma A-star yang dikenal memiliki performa yang baik sehingga dapat memberi wawasan kepada pada peneliti atau programmer game mengenai pencarian jalur terpendek yang lebih efisien.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah yang terdapat dalam penelitian ini adalah:

1. Bagaimana cara implementasi algoritma A-star dan NavMesh untuk *Pathfinding* NPC dalam permainan *maze*?
2. Bagaimana perbandingan kecepatan waktu algoritma A-star dengan NavMesh Unity pada NPC dalam permainan *maze*?

## 1.3 Batasan Permasalahan

Batasan masalah yang ada pada penelitian ini adalah :

1. Map yang akan dibuat berjumlah lima dan simulasi akan dijalankan sepuluh kali untuk setiap map kemudian dicari waktu *pathfinding* NPC rata-rata
2. Map satu hingga lima akan diurutkan berdasarkan jumlah jarak dan belokan
3. Semua map yang dibangun memiliki ukuran 15x15 blok.

4. Setiap map memiliki NPC berjumlah satu dan memiliki satu target destinasi.
5. Simulasi algoritma A-star dan NavMesh akan dijalankan secara terpisah.

#### 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disusun, maka tujuan penelitian yang ingin dicapai adalah:

1. Mengimplementasikan algoritma A-star dan NavMesh untuk *pathfinding* NPC dalam permainan *maze*
2. Membandingkan kecepatan waktu algoritma A-star dengan NavMesh pada NPC unity dalam permainan *maze*

#### 1.5 Manfaat Penelitian

Manfaat dari penelitian yang akan dilakukan adalah:

1. Menunjukkan perbandingan kecepatan waktu *pathfinding* antara algoritma A-star dengan NavMesh pada Unity dalam permainan *maze*.
2. Sebagai rujukan untuk pengembang lain dalam menentukan *pathfinding* yang lebih efisien untuk digunakan.
3. Sebagai acuan untuk peneliti lain agar dapat menambah pengetahuan mengenai algoritma A-star ataupun NavMesh Unity yang ingin dijadikan sebagai topik penelitian.

#### 1.6 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN

Bab satu akan menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan laporan.

- Bab 2 LANDASAN TEORI

Bab kedua akan menjelaskan tentang landasan teori yang berkaitan dengan penelitian ini, yaitu Unity, Algoritma A-Star, dan Navigation Mesh Unity.

- Bab 3 METODOLOGI PENELITIAN

Bab tiga akan menjelaskan tentang metodologi penelitian berupa tahapan yang akan dilaksanakan dalam penelitian serta rancangan berupa mockup dan flowchart.

- Bab 4 HASIL DAN DISKUSI

Bab empat akan menjelaskan tentang hasil penelitian yang telah dilakukan berupa implementasi sistem yang telah dirancang serta hasil perhitungan yang telah dilakukan.

- Bab 5 KESIMPULAN DAN SARAN

Bab lima akan berisi mengenai kesimpulan dari pengujian yang telah dilakukan serta berisi saran bagi pembaca atau pengembang berikutnya.

