

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Penemuan jalur (*pathway finding*) merupakan masalah fundamental dalam kecerdasan buatan dan robotika dengan berbagai aplikasi luas, seperti navigasi, perencanaan, dan desain game. Dengan kata lain, *pathway finding* adalah proses mencari jalur terbaik untuk mencapai tujuan dari suatu titik awal, dengan mempertimbangkan batasan-batasan yang ada di lingkungan [1]. Salah satu algoritma yang sangat penting dalam AI adalah algoritma A* (A-star) dalam pengembangan *pathway-finding* baik itu untuk game ataupun yang lainnya[2].

Penelitian Duchoň et al.[4] menyatakan bahwa Keunggulan utama algoritma A* adalah sederhana, relatif cepat, dan dapat dimodifikasi. Algoritma ini digunakan untuk menemukan jalur terpendek antara dua titik dalam suatu ruang pencarian dengan mempertimbangkan berbagai kendala atau hambatan. Penelitian Candra et al. [5] juga menemukan bahwa A* merupakan algoritma best-first search, yang mampu menangani pencarian jalur terpendek dengan waktu lebih cepat namun tidak selalu optimal dibandingkan dengan algoritma Dijkstra dengan hasil optimal dalam waktu pencarian lebih lama. Dengan beberapa keunggulannya, penggunaan algoritma A* mencakup berbagai aplikasi dalam kehidupan sehari-hari, seperti sistem navigasi GPS, game, perencanaan pergerakan robot, dan banyak lagi. Selain itu, kelebihan algoritma A* terletak pada efisiensinya. Algoritma ini memanfaatkan heuristik untuk mempersempit pencarian, fokus pada simpul yang paling mungkin mengarah ke tujuan, sehingga menghemat waktu dan sumber daya komputasi. Di balik kelebihannya, algoritma ini juga memiliki beberapa kekurangan yang perlu dipertimbangkan. Kinerjanya sangat bergantung pada heuristik yang digunakan. Heuristik yang buruk dapat menghasilkan jalur suboptimal, bahkan gagal menemukan jalur sama sekali. [7]. Beberapa modifikasi telah dilakukan untuk meningkatkan efisiensi dari algoritma A* ini, seperti yang telah dilakukan

oleh Liu et al. [8]. Performa A* sangat bergantung pada fungsi heuristik yang dipilih. Selain itu, A* perlu menyimpan semua jalur yang sedang terbuka (open set) dan jalur tertutup (closed set). Hal ini dapat menghabiskan banyak memori, terutama untuk grafik yang besar. Meskipun A* adalah algoritma pencarian yang ampuh, ia memiliki keterbatasan. Memilih heuristik yang tepat dan mengelola kompleksitas komputasi sangat penting untuk penerapan yang efektif.

Pada game 3D Algoritma A* dapat digunakan untuk meningkatkan kualitas game 3D. Namun, penting untuk mempertimbangkan kendala dan tantangan yang dapat dihadapi saat menggunakan algoritma ini. Kendala dan tantangan ini meliputi, Kompleksitas algoritma, Kinerja algoritma, Ketersediaan data. Badzrotul Mufida menyatakan bahwa, algoritma A* dapat menjadi tidak efisien dalam game 3D dengan dunia permainan yang besar dan kompleks [9]. Penelitian Neto et al. [10] Menunjukkan bahwa rute yang dihasilkan oleh algoritma 2D sekitar 6% lebih panjang dibandingkan dengan pendekatan 3D, namun dengan waktu eksekusi rata-rata sembilan kali lebih rendah.

Salah satu teknik optimasi yang dapat dilakukan untuk mencapai akurasi yang tinggi menggunakan heap sort. Metode ini merupakan metode yang paling cepat dibandingkan dengan metode median sort dan rapid sort [11]. Heapsort merupakan algoritma pengurutan yang bekerja dengan cara membangun tumpukan (heap). Tumpukan adalah struktur data yang menyimpan data dalam urutan khusus, yaitu urutan terurut dari nilai minimum. Heap sort adalah algoritma pengurutan yang dapat digunakan untuk meningkatkan kinerja algoritma A* [9].

Berdasarkan hal tersebut maka penelitian ini akan dilakukan penerapan metode heapsort untuk mencari rute terpendek dengan melakukan optimalisasi terhadap algoritma A* pada *game pathway-finding* 3D.

1.2 Rumusan Masalah

Berdasarkan permasalahan penelitian yang telah dijelaskan dalam latar belakang sebelumnya, dapat diidentifikasi permasalahan sebagai berikut:

1. Bagaimana kinerja algoritma A* dalam pencarian rute terpendek pada game 3D?
2. Bagaimana kinerja algoritma A* dengan optimasi heapsort dalam pencarian rute terpendek pada game 3D?

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah dalam membangun sebuah game 3D yang di mana *enemy* pada game ini terdapat algoritma A* yang dioptimasi oleh *heap sort* agar *enemy* dapat mencari rute terpendek untuk menghampiri *player*. Berdasarkan latar belakang masalah yang telah dijelaskan sebelumnya, maka tujuan penelitian ini adalah sebagai berikut:

1. Mengukur kinerja algoritma A* dalam pencarian rute terpendek pada game 3D.
2. Mengukur kinerja algoritma A* dengan optimasi heapsort dalam pencarian rute terpendek pada game 3D.

1.4 Batasan Masalah

Adapun batasan-batasan masalah yang ada di dalam penelitian ini meliputi:

1. Pengukuran kinerja dilakukan dengan mengukur parameter – parameter jumlah node, waktu eksekusi, nilai heuristik, jalur robustness, optimalisasi jalur.
2. Penelitian ini berfokus pada pengembangan algoritma A* yang lebih efisien untuk game pathway-finding pada game 3D.
3. Penelitian ini berfokus pada evaluasi kinerja algoritma A* dengan optimasi heapsort untuk game pathway-finding dengan berbagai ukuran lingkungan dan jenis rintangan.
4. Game yang dibangun pada penelitian ini menggunakan *game engine* Unity dengan bahasa pemrograman C# (CSharp).

1.5 Metodologi Penelitian

Metodologi penelitian merupakan suatu proses untuk memecahkan sebuah permasalahan secara logis, dimana memerlukan data – data literatur dari beberapa jurnal, artikel, dan buku. yang mendukung untuk melaksanakan suatu penelitian. Metodologi yang digunakan dalam penelitian ini menggunakan Metode Eksperimen. Metode Eksperimen merupakan metode kuantitatif yang digunakan untuk mempelajari suatu variabel independen (percobaan) terhadap variabel dependen (hasil) dalam kondisi yang terkontrol. Pendekatan eksperimen memungkinkan peneliti untuk secara sistematis menguji dan membandingkan efektivitas kedua algoritma di bawah kondisi yang terkontrol. Penelitian ini melibatkan serangkaian pengujian yang dilakukan pada tiga level dengan ukuran grid yang berbeda-beda, yaitu 50x50, 65x65, dan 100x100. Setiap level diuji dengan beberapa parameter iterasi dan pengujian untuk mengukur waktu eksekusi, jumlah node, kelancaran jalur Robustness, nilai heuristik dan optimalisasi jalur dalam berbagai kondisi. [33]

1.5.1 Pengumpulan Data

Pada teknik pengumpulan data yang digunakan terdiri dari :

1. Studi Literatur

Studi Literatur merupakan kegiatan dengan melakukan pencarian dan pengumpulan data pustaka yang menunjang penelitian yang akan dikerjakan. Data pustaka tersebut berupa buku, artikel, jurnal, dan laporan akhir yang berkaitan dengan judul penelitian.

2. Studi Kasus

Studi kasus adalah metode pengumpulan data dengan cara mempelajari secara mendalam suatu kasus tertentu. Studi kasus dapat dilakukan secara kuantitatif.

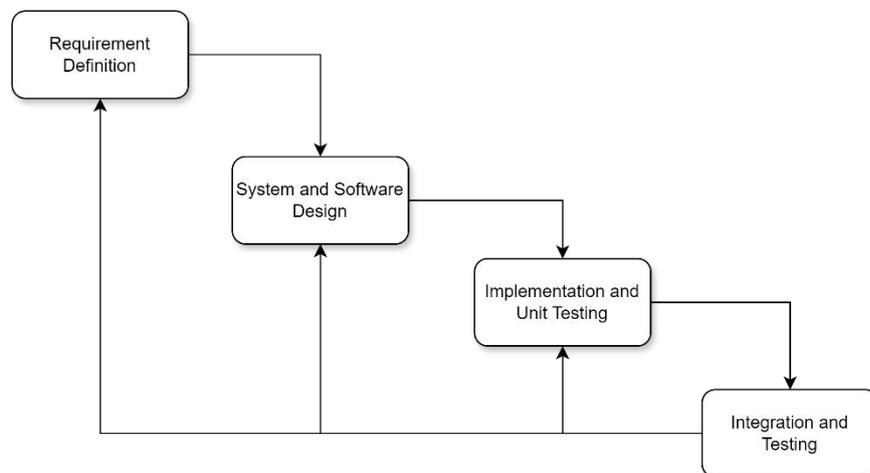
3. Eksperimen

Eksperimen adalah metode pengumpulan data dengan cara melakukan suatu percobaan untuk menguji suatu hipotesis. Eksperimen dalam penelitian ini dilakukan dengan menguji algoritma A* yang dioptimasi

dengan algoritma heapsort yang telah dibuat dan diimplementasikan ke dalam game 3D.

1.5.2 Pembangunan Perangkat Lunak

Metode waterfall adalah salah satu metode pengembangan perangkat lunak yang paling umum digunakan. Metode ini menggunakan pendekatan berurutan, di mana setiap tahap pengembangan harus diselesaikan sebelum tahap selanjutnya dimulai.



Gambar 1.1 Metode Waterfall

Sumber Gambar : Ian Sommerville.2011. *Software Engineering 9th Edition*. Addison-Wesley.[14]

Metode waterfall terdiri dari beberapa tahap, yaitu:

1. Definisi Kebutuhan

Pada tahap ini, peneliti harus mengumpulkan dan memahami kebutuhan pengujian, serta menentukan variabel yang akan diukur. Variabel yang akan diukur dalam penelitian ini adalah jumlah node, waktu eksekusi, nilai heuristik, kelancaran jalur, dan optimalisasi jalur yang dibutuhkan oleh algoritma A* untuk mencari jalur terpendek dalam ruang 3D.

2. System and Software Design

Pada tahap ini, peneliti harus merancang sistem yang akan dibangun untuk memenuhi kebutuhan pengujian, serta menentukan metode pengumpulan dan analisis data. Metode pengumpulan data yang dapat digunakan dalam penelitian ini adalah eksperimen. Dalam eksperimen, peneliti akan menguji algoritma A* dengan menggunakan optimasi heapsort dan algoritma A* tanpa optimasi heapsort.

3. Implementasi and Unit Testing

Tahapan implementasi dan testing dalam sebuah penelitian, terutama yang berfokus pada pengembangan algoritma A* (A-Star) dengan optimasi heapsort, merupakan proses yang krusial untuk memastikan bahwa solusi yang dihasilkan memenuhi tujuan yang telah ditetapkan.

4. Integration and System Testing

Pada tahap ini, peneliti harus menguji parameter yang telah dibangun untuk memastikan bahwa pengujian tersebut dapat mengukur perbandingan algoritma A* dengan optimasi heapsort dan tanpa optimasi serta bebas dari kesalahan. Dalam penelitian ini, peneliti dapat menggunakan metode pengujian yang sesuai dengan variabel yang diukur, yaitu jumlah node, waktu eksekusi, nilai heuristik, kelancaran jalur dan optimalisasi heapsort.

1.6 Sistematika Penulisan

Untuk memberikan gambaran secara umum mengenai penulisan tugas akhir yang dilakukan maka sistematika penulisannya adalah sebagai berikut :

BAB 1 PENDAHULUAN

Pada bab ini berisi uraian latar belakang masalah, rumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, tahap pengumpulan data, model pengembangan perangkat lunak dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini akan membahas berbagai konsep konsep dasar dan teori-teori pendukung yang berhubungan dengan game 3D, algoritma A* dengan optimasi heapsort, dan pencarian jalur terpendek.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan membahas tentang analisis masalah, gambaran umum, analisis pengembangan aplikasi game, arsitektur system, analisis kebutuhan, penerapan algoritma A* dengan optimasi Heapsort, dan perancangan antarmuka.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini berisi hasil implementasi analisi dari BAB 3 dan perancangan aplikasi yang dilakukan, serta hasil kinerja aplikasi untuk mengetahui apakah aplikasi yang dibangun sudah memenuhi kebutuhan.

BAB 5 KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang diperoleh seberapa akurat dan efektif pada algoritma A* dengan optimasi heapsor dalam pencarian rute terpendek aplikasi pada game 3D , serta kesimpulan dan saran untuk pengembangan aplikasi yang telah dirancang.