

BAB 5. KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan, mulai dari perancangan hingga tahap evaluasi pada algoritma, dapat disimpulkan bahwa penilitan yang berjudul “Analisis Perbandingan Kinerja Algoritma *Pathfinding* Pada *Non-Playable Character* Dalam Bentuk *Maze Solving* Dengan Menggunakan Unity” memberikan beberapa kesimpulan terhadap kinerja algoritma antara lain :

1. Berdasarkan hasil pengujian dan dengan kriteria pengujian yang dilakukan, estimasi waktu yang didapatkan dari masing – masing algoritma berbeda satu dengan yang lain. Algoritma *greedy best first search* menjadi algoritma yang tercepat dalam menyelesaikan *maze problem* dengan rata – rata waktu estimasi yaitu 0.759 ms, dan diikuti dengan algoritma A* yang menjadi algoritma tercepat kedua dengan waktu estimasi yaitu 0.950 ms. A* pada kasus ini terkadang mengevaluasi lebih banyak *node* sehingga mengurangi waktu untuk menemukan tujuan. Sedangkan untuk algoritma dijkstra, *backtracking* dan BFS memiliki estimasi waktu yang jauh lebih tinggi dengan waktu estimasi 2.928 ms untuk dijkstra, 2.452 ms untuk BFS, dan 4.269 ms untuk *backtracking*. Hal ini dikarenakan algoritma - algoritma ini mengevaluasi sangat banyak *node* untuk menemukan tujuan dan hal ini dapat dilihat pada *maze* dengan ukuran yang besar.
2. Berdasarkan hasil pengujian dan dengan kriteria pengujian yang dilakukan, untuk Panjang jalur, algoritma A*, dijkstra dan BFS selalu memberikan jalur paling optimal dengan total keseluruhan panjang jalur akhir yaitu, 2093 *node*. Hal ini sesuai dengan kriteria algoritma tersebut yang memberikan jalur paling optimal, sedangkan algoritma *greedy best first search* terkadang menggunakan jalur yang kurang optimal dengan total keseluruhan panjang jalur akhir yaitu, 2139 *node*. Hal dikarenakan algoritma ini hanya menggunakan jalur yang dievaluasi oleh algoritma tersebut. Sedangkan untuk algoritma *backtracking*, jalur yang dihasilkan sangat tidak optimal dengan total keseluruhan panjang jalur akhir yaitu, 10725 *node*. Hal ini dikarenakan algoritma *backtracking* bergantung pada jalur yang dijelajahi. Untuk banyaknya jalur yang dijelajah, algoritma A* dan *greedy best first search* merupakan algoritma yang paling sedikit menjelajah *node* dengan total banyaknya *node* yang dijelajah, yaitu 4626 *node* untuk algoritma *greedy* BFS dan 6000 *node* untuk algoritma A*. Hal ini dikarenakan algoritma ini merupakan algoritma *informed* yang menggunakan perhitungan *cost* untuk mengevaluasi jalur yang akan dijelajah, sedangkan untuk algoritma dijkstra, *backtracking*, dan BFS dengan total banyaknya *node* yang

dijelajah, yaitu 25305 *node* untuk algoritma dijkstra dan BFS, dan 17830 *node* untuk algoritma *backtracking*. Hal ini dikarenakan algoritma – algoritma ini merupakan algoritma *uninformed* yang dimana memerlukan penjelajahan *node* yang lebih banyak untuk mencapai tujuan.

3. Berdasarkan hasil pengujian dan dengan kriteria pengujian yang dilakukan, untuk penggunaan *memory* dan penggunaan CPU, algoritma *greedy best first search* menggunakan penggunaan CPU dan *Memory* yang cukup optimal dengan rata – rata penggunaan CPU sebanyak 4.551% dan dengan rata-rata penggunaan *memory* sebanyak 115.183 *bytes* dan diikuti dengan algoritma A* dengan rata – rata penggunaan CPU sebanyak 5.700% dan dengan rata-rata penggunaan *memory* sebanyak 149.567 *bytes*, hal ini disebabkan karena kedua algoritma tidak menyimpan banyak *node* dan waktu berjalan algoritma yang cepat sehingga penggunaan *memory* dan CPU relatif rendah. Sedangkan untuk algoritma dijkstra menggunakan rata – rata penggunaan CPU sebanyak 17.570% dan dengan rata-rata penggunaan *memory* sebanyak 516.150 *bytes*, *backtracking* menggunakan rata – rata penggunaan CPU sebanyak 25.603% dan dengan rata-rata penggunaan *memory* sebanyak 490.767 *bytes*, dan BFS menggunakan rata – rata penggunaan CPU sebanyak 14.699% dan dengan rata-rata penggunaan *memory* sebanyak 456.975 *bytes*, algoritma ini menyimpan lebih banyak *node* dan waktu berjalan algoritma yang lebih lama sehingga menggunakan *memory* dan CPU lebih banyak dibandingkan dengan algoritma *greedy best first search* dan A*.
4. Berdasarkan data yang didapat dari pengujian dengan perbedaan kepadatan dari suatu *map*, kepadatan ini mempengaruhi hasil akhir dari pengujian walaupun dengan ukuran *map* yang sama.

5.2. Saran

Untuk penelitian selanjutnya disarankan untuk menguji algoritma dengan ukuran *map* yang jauh lebih besar untuk mendapatkan *worst case* dari algoritma yang diuji dan jenis *map* lain yang lebih dinamis, serta mempertimbangkan algoritma tambahan seperti DFS, JPS, atau kombinasi algoritma. Pengujian juga sebaiknya dilakukan pada perangkat dengan spesifikasi rendah agar hasil lebih aplikatif. Selain itu, visualisasi hasil *pathfinding* dapat dikembangkan agar lebih informatif dan lebih mudah untuk dipahami.