BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dihasilkan dari perbandingan algoritma A* dan *Breadth first search* adalah

1. Algoritma *Breadth first search* melakukan pencarian dengan mengunjungi satu persatu jalur di setiap percabangan untuk menuju lokasi akhir, sehingga algoritma *Breadth first search* memiliki waktu eksekusi yang lebih cepat dibanding algoritma A* karena memiliki proses yang lebih sederhana, hal ini terjadi apabila memiliki waktu yang singkat untuk mengunjungi setiap percabangan simpul atau simpul-simpul anak.
2. Sedangkan algoritma A* menemukan lokasi akhir dengan cara melakukan perhitungan nilai heuristik dan melakukan proses runut balik apabila jalur tidak ditemukan untuk menuju lokasi akhir, akibatnya algoritma ini hanya memilih simpul yang paling optimal untuk menuju lokasi akhir, sehingga algoritma A* mengunjungi jalur yang lebih sedikit bila dibandingan algoritma *Breadth first search.*
3. Berdasarkan hasil penelitian yang dihitung dari aspek ketuntasan hasil pencarian (*completeness)* yang terdapat pada tabel 4.3, waktu eksekusi (*running time)* yang terdapat pada tabel 4.5, data ruang yang didapatkan dari jumlah simpul yang dikunjungi (*space)* pada tabel 4.4 dan optimasi (*optimized)* yang berasal dari tabel 4.3 dan proses yang terjadi pada bagian Analisis dan Perancangan Sistem. Algoritma A*

unggul dalam aspek ketuntasan hasil pencarian (*completeness)*, ruang (*space)* dan optimasi (*optimized)* sedangkan algoritma *Breadth first search* hanya unggul pada ketuntasan hasil pencarian (*completeness)* dan waktu (*running time)*. Sehingga, Algoritma A* memiliki kinerja yang lebih baik dibandingkan algoritma *Breadth first search.*

5.2 Saran

Saran untuk pengembangan penelitian pada aplikasi yang menggunakan pencarian algoritma A*, pencarian algoritma *Breadth first search*, dan aplikasi untuk membandingkan algoritma A* dan algoritma *Breath first search* yaitu

1. Apabila menggunakan algoritma A* untuk mempercepat pencarian pada *block maze* maka yang harus dilakukan untuk menggunakan algoritma A* adalah lakukan optimasi simpul yang dipilih pada runut balik untuk menuju simpul tujuan untuk memperbaiki jumlah proses pada algoritma A*.
2. Ketika menggunakan algoritma *Breadth first search*, algoritma ini memiliki waktu eksekusi yang cepat karena prosesnya yang sederhana. Algoritma ini memiliki kelemahan yaitu ketika melakukan pencarian maka algoritma ini sangat boros akan penyimpanan data ke memori karena mengunjungi simpul-simpul yang tidak optimal untuk menuju lokasi tujuan atau setiap percabangan untuk menuju simpul tujuan, tetapi karena prosesnya yang singkat apabila memiliki waktu yang singkat untuk mengunjungi simpul-simpul anaknya, maka algoritma *Breadth first search* cocok untuk melakukan pencarian pada *block maze*.
3. Kasus *block maze* yang digunakan pada aplikasi ini adalah kasus berupa pencarian peta pada *block maze* yang hanya memiliki 1 solusi untuk menemukan simpul tujuan. Sebaiknya kasus yang digunakan untuk pencarian pada *block maze nanti* adalah kasus yang memiliki lebih dari 1 solusi untuk mencapai simpul tujuan, supaya dapat terlihat

keoptimalan yang dihasilkan dari algoritma A* apabila dibandingkan dengan algoritma *Breadth first search*.

4. Aplikasi yang telah dibuat hanya mampu menyelesaikan dan membuat block maze hingga memiliki kolom 100 dan baris 100, dengan menggunakan JPanel pada bahasa pemrograman Java. Sebaiknya gunakan manajemen memori yang lebih baik dan memilih komponen untuk pembuatan simpul yang lebih menghemat memori, sehingga aplikasi dapat melakukan pencarian pada *block maze* yang dengan jumlah kolom dan baris yang lebih banyak.
5. Pada aplikasi yang digunakan untuk membandingkan algoritma A* dan algoritma *Breadth first search* pada *block maze*. Pada proses pencarian dengan menggunakan algoritma A*, proses ini melakukan banyak perulangan yang tidak efisien, seperti pemeriksaan kedekatan simpul induk dari simpul-simpul anak untuk dijadikan jalur menuju simpul tujuan, sehingga jumlah proses yang terjadi pada aplikasi menjadi lebih banyak daripada proses yang seharusnya dilakukan oleh algoritma A*. Sebaiknya, proses yang digunakan pada pencarian dengan menggunakan algoritma A* ini dibuat menjadi lebih efisien dengan mengurangi perulangan pemeriksaan simpul yang tidak diperlukan.