

## BAB V

### KESIMPULAN & SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian, implementasi, dan analisis performa terhadap algoritma A\* dan A\* dengan *Jump Point Search* (JPS) pada *game* berbasis *grid movement*, beberapa kesimpulan dapat diambil sebagai berikut:

1. *Game* berbasis *grid movement* berhasil dikembangkan menggunakan Unity dengan memanfaatkan *tilemap* sebagai representasi lingkungan dan sistem pergerakan berbasis *grid* sebagai fondasi utama. *Game* ini berfungsi tidak hanya sebagai permainan, tetapi juga sebagai *platform* studi kasus untuk mengimplementasikan dan menguji algoritma *pathfinding*. Sistem visualisasi *open list*, *closed list*, dan jalur akhir juga berhasil diintegrasikan sehingga proses pencarian jalur dapat diamati secara jelas.
2. Algoritma A\* dan A\* yang dioptimalkan dengan JPS berhasil diimplementasikan dalam *game* menggunakan struktur data *array* sebagai representasi peta. JPS mampu diterapkan dengan baik pada peta *grid* dengan mendeteksi titik-titik *jump points* yang memungkinkan pemangkasan simpul pencarian tanpa menghilangkan optimalitas hasil. Implementasi kedua algoritma berjalan stabil dan dapat diintegrasikan langsung dengan sistem *movement* pada Unity.
3. Pengujian dilakukan melalui modul *benchmark* yang dirancang khusus, dengan memanfaatkan *dataset* peta uji Sturtevant (2012) dan lebih dari satu juta skenario pencarian jalur. Evaluasi dilakukan menggunakan metrik *completeness*, *optimality*, *time complexity*, dan *space complexity*. Seluruh skenario diuji dengan kondisi *start-goal* yang sama untuk memastikan perbandingan yang adil. Data pengujian disimpan secara sistematis untuk dianalisis secara kuantitatif.
4. Hasil perbandingan kinerja antara algoritma A\* dan A\* dengan *Jump Point Search* (JPS) pada lebih dari 1,35 juta skenario uji menunjukkan

bahwa keduanya memiliki tingkat *completeness* dan *optimality* sebesar 100%, tanpa adanya deviasi antara *path cost* dan *optimal cost* pada seluruh pengujian. Perbedaan utama terletak pada efisiensi komputasi, di mana dari sisi *time complexity*, A\* memiliki rata-rata waktu eksekusi sebesar 17,10 ms, sedangkan JPS lebih cepat dengan rata-rata 5,92 ms, menunjukkan peningkatan performa yang signifikan. Dari sisi *space complexity*, perbedaan juga terlihat jelas, di mana rata-rata *open node* pada A\* mencapai 1.764 *node*, sedangkan JPS hanya 83 *node*, serta rata-rata *closed node* pada A\* sebesar 45.569 *node* dibandingkan 3.395 *node* pada JPS, yang menunjukkan bahwa JPS mampu secara drastis mengurangi jumlah eksplorasi simpul. Selain itu, pada *parameter* penggunaan memori, A\* menggunakan rata-rata 9.092.482 *bytes* ( $\approx 9,09$  MB), sedangkan JPS menggunakan 7.793.450 *bytes* ( $\approx 7,79$  MB), sehingga terjadi penghematan memori yang cukup signifikan. Secara keseluruhan, hasil ini membuktikan bahwa JPS mampu mempertahankan optimalitas solusi yang sama dengan A\*, sekaligus memberikan peningkatan efisiensi yang signifikan baik dari segi waktu eksekusi maupun penggunaan sumber daya. Hasil uji ANOVA yang dilakukan pada seluruh metrik performa, termasuk waktu eksekusi, jumlah *open node*, jumlah *closed node*, dan penggunaan memori menunjukkan bahwa seluruh nilai *f* hitung mengungguli *f* kritis dengan *p-value* berada jauh di bawah ambang signifikansi ( $\alpha = 0,05$ ), sehingga seluruh  $H_0$  ditolak. Dengan demikian, terdapat perbedaan yang secara statistik signifikan antara performa A\* dan JPS pada semua *parameter* yang dianalisis. Secara keseluruhan, hasil penelitian ini membuktikan bahwa JPS mampu mempertahankan optimalitas solusi yang sama dengan A\*, sekaligus memberikan peningkatan efisiensi yang signifikan baik dari segi waktu eksekusi maupun sumber daya komputasi.

## 5.2 Saran

Berdasarkan hasil penelitian ini, beberapa saran dapat diberikan untuk pengembangan lebih lanjut:

1. Implementasi dapat diperluas dengan menambahkan algoritma lain seperti D\* Lite, Theta\*, atau algoritma berbasis heuristik adaptif untuk memberi perbandingan lebih beragam pada berbagai jenis peta.
2. *Game* studi kasus dapat dikembangkan dengan peta dinamis, peta bergerak (*moving obstacles*), atau peta dengan perubahan waktu nyata untuk menguji robustitas algoritma terhadap kondisi lingkungan yang berubah.
3. Melakukan analisis lebih dalam terhadap karakteristik peta. Walaupun penelitian ini tidak mengategorikan peta berdasarkan tipe strukturnya (terbuka, tertutup, acak, teratur dsb.) analisis berbasis morfologi peta dapat membantu memahami penyebab performa algoritma *pathfinding* berbeda antar kategori peta.
4. Studi lanjutan dapat menguji performa algoritma pada situasi *real-time*, misalnya *multi-agent movement*, *dynamic avoidance*, atau skenario permainan yang memerlukan pembaruan jalur secara kontinu.