

ABSTRAK

Game Tower Defense memerlukan mekanisme pencarian jalur atau biasa disebut sebagai metode *pathfinding* yang efisien agar unit musuh dapat bergerak menuju tujuannya secara cerdas. Algoritma Dijkstra merupakan salah satu metode fundamental yang dapat menjamin penemuan jalur terpendek dalam sebuah peta permainan berbasis graf dan sebuah *grid* berbentuk peta dua dimensi. Penelitian ini bertujuan untuk (1) mengimplementasikan algoritma Dijkstra pada sebuah prototipe game Tower Defense yang dibangun menggunakan Unity Engine dengan bahasa C#, dan (2) menganalisis performa algoritma tersebut dalam menentukan jalur yang optimal. Metode penelitian yang digunakan adalah eksperimental dengan pendekatan komparatif, di mana sistem yang diimplementasikan diuji pada skenario peta yang identik dengan sistem referensi berbasis Python/Pygame. Analisis dilakukan dengan membandingkan hasil visual, metrik kuantitatif, dan analisis kode sumber dengan metode *white box*. Hasil penelitian menunjukkan bahwa implementasi algoritma Dijkstra berhasil. Meskipun terdapat perbedaan visual pada rute yang dihasilkan antara sistem peneliti dan sistem referensi, kedua sistem terbukti menemukan jalur yang sama-sama optimal dengan panjang identik memiliki 48 langkah. Analisis lebih lanjut menyimpulkan bahwa akar penyebab perbedaan rute tersebut adalah perbedaan urutan prioritas dalam pengecekan simpul pada tetangga ketika algoritma menghadapi kondisi *tie-breaking*. Dengan demikian, disimpulkan bahwa algoritma Dijkstra telah berhasil diimplementasikan dan divalidasi kinerjanya, di mana perbedaan visual pada jalur tidak menandakan kesalahan atau sub-optimalitas, melainkan merupakan karakteristik yang dapat diprediksi dari detail implementasi algoritma.

Kata Kunci: *Pathfinding, Algoritma Dijkstra, Game Tower Defense, Unity, Analisis Komparatif.*

ABSTRACT

Tower Defense games require an efficient pathfinding mechanism to enable enemy units to move intelligently towards their goal. Dijkstra's algorithm is a fundamental method that guarantees the discovery of the shortest path in graph-based game maps and two-dimensional grid maps. This research aims to (1) implement Dijkstra's algorithm in a Tower Defense game prototype built using the Unity Engine with the C# language, and (2) analyze the algorithm's performance in determining the optimal path. The research method employed is experimental, with a comparative approach, where the implemented system is tested against a Python/Pygame-based reference system on an identical map scenario. The analysis was conducted by comparing visual results, quantitative metrics, and source code through white-box analysis. The results show that the implementation of Dijkstra's algorithm was successful. Although there were visual differences in the generated routes between the researcher's system and the reference system, both were proven to find equally optimal paths with an identical length of 48 steps. Further analysis concluded that the root cause of these route differences was the variation in priority order when checking neighboring nodes during tie-breaking conditions in the algorithm. Therefore, it is concluded that Dijkstra's algorithm was successfully implemented and its performance validated. The visual differences in the paths do not indicate errors or sub-optimality, but rather are a predictable characteristic of the algorithm's implementation details.

Keywords: *Pathfinding, Dijkstra's Algorithm, Tower Defense Game, Unity, Comparative Analysis.*