BAB IV
KESIMPULAN

4.1 Kesimpulan

Berdasarkan rumusan masalah "Bagaimana Finite State Machine dapat digunakan untuk mengatur transisi animasi karakter pada game Legacy of the Sunstone?", maka dapat disimpulkan bahwa:

1. Finite State Machine dapat digunakan untuk mengatur transisi animasi karakter pada game Legacy of the Sunstone dengan cara membagi perilaku karakter ke dalam state-state independen seperti *Idle*, *Movement*, *Jump*, *Falling*, *Crouch*, *Aim*, *Melee*, dan *Takedown*. Setiap state memiliki animasi yang sudah ditentukan dan akan diputar saat state tersebut aktif, sesuai dengan karakteristik Moore Machine di mana output animasi ditentukan oleh state yang sedang berjalan.
2. Transisi antar state terjadi berdasarkan input pemain seperti tombol keyboard dan mouse serta kondisi tertentu seperti posisi karakter di tanah atau di udara. Untuk menghasilkan perpindahan animasi yang mulus, digunakan teknik *CrossFadeInFixedTime* yang memungkinkan blending antara animasi sebelumnya dengan animasi baru dalam durasi waktu yang dapat dikonfigurasi.
3. Implementasi Finite State Machine menggunakan arsitektur Hierarchical Finite State Machine dengan State Pattern pada bahasa pemrograman C# dan Unity Engine. Hasil pengujian *Black Box* Testing terhadap 13 skenario menunjukkan bahwa seluruh fitur berfungsi sesuai harapan, membuktikan bahwa Finite State Machine mampu mengatur transisi animasi karakter secara efektif dan menghasilkan pengalaman bermain yang responsif.

4.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan lebih lanjut:

1. Menambahkan state baru seperti Swimming atau Vehicle untuk memperkaya mekanik gameplay.
2. Mengembangkan visual debugging tools pada Unity Editor untuk menampilkan state aktif dan transisi yang terjadi secara real-time guna mempermudah proses pengembangan.
3. Mengembangkan visual debugging tools pada Unity Editor untuk menampilkan state aktif dan transisi yang terjadi secara real-time guna mempermudah proses pengembangan.
4. Menambahkan unit testing untuk memastikan setiap state berfungsi dengan benar secara independen sebelum diintegrasikan.
5. Implementasi pada Enemy *Artificial Intelegence* Arsitektur Finite State Machine yang sama dapat diterapkan pada karakter musuh untuk mengelola perilaku *Artificial Intelegence* seperti *Patrol*, *Chase*, *Attack*, dan *Retreat*.