

BAB V KESIMPULAN

A. Kesimpulan

Berdasarkan hasil implementasi *Finite State Machine* (FSM) menggunakan Bahasa Lua sebagai bahasa skrip utama pada *platform* Roblox Studio, dapat disimpulkan bahwa FSM berhasil diterapkan untuk mengelola perilaku NPC melalui *state Idle, Patrol, Chase, Attack, dan Dead. Script* pada Roblox Studio dengan memanfaatkan sejumlah fungsi dan struktur Bahasa Lua, antara lain variabel *state* untuk menyimpan kondisi NPC yang aktif, struktur kondisional *if-else* untuk memeriksa syarat perpindahan antar *state*, fungsi perulangan (*loop*) yang dikombinasikan dengan *task.wait()* untuk mengatur jeda waktu antar aksi, perhitungan jarak (*magnitude*) untuk mendeteksi posisi *player* dalam radius *DETECTION_RADIUS* maupun *ATTACK_RADIUS*, serta *event* seperti *Touched* dan *RemoteEvent* untuk menangani interaksi seperti tembakan bola api dan mekanisme *stomp*, hingga fungsi *model:Destroy()* untuk menghapus NPC ketika memasuki *state Dead*. Setiap *state* dihubungkan melalui aturan transisi yang jelas, sehingga NPC mampu berpindah perilaku secara otomatis sesuai kondisi permainan, mulai dari berpatroli, mengejar, menembakan api, hingga mati ketika dikalahkan pemain. Implementasi dilakukan melalui *Server Script* agar *state* NPC tetap konsisten dan tersinkronisasi bagi seluruh pemain dalam lingkungan *multiplayer*. Dengan demikian, metode FSM terbukti efektif sebagai solusi sistematis untuk mengendalikan perilaku NPC secara dinamis dan terstruktur pada game “Super Qui”.

Hasil evaluasi FSM pada NPC game “Super Qui” menggunakan *Harness Test* menunjukkan bahwa seluruh skenario pengujian berhasil dijalankan dengan status *passed* tanpa kegagalan, yang membuktikan bahwa logika transisi *state* telah berfungsi sesuai rancangan. Hasil pengujian *logging* memperkuat temuan tersebut dengan menunjukkan perpindahan *state* yang konsisten dan sesuai kondisi permainan. Evaluasi lebih lanjut melalui *Game Experience Questionnaire* (GEQ) menunjukkan bahwa implementasi NPC berbasis FSM memperoleh penilaian yang signifikan lebih tinggi daripada nilai tengah skala, sehingga menunjukkan bahwa *gameplay* yang menggunakan FMS diterima secara positif oleh pemain, yang ditandai dengan nilai signifikan lebih tinggi pada komponen *Immersion, Competence, dan Positive Affect* ($p < 0,05$), serta nilai signifikan lebih rendah pada komponen *Tension* dan *Negative Affect*, sedangkan komponen *Flow* dan *Challenge* berada pada tingkat netral. Dengan demikian, hipotesis alternatif (H_a) yang menyatakan bahwa penerapan FSM pada NPC berpengaruh secara signifikan terhadap *gameplay* diterima, sehingga dapat disimpulkan bahwa implementasi FSM mampu menghasilkan perilaku NPC yang dinamis dan responsif serta meningkatkan kualitas pengalaman bermain pada game “Super Qui”.

B. Saran

Berdasarkan hasil implementasi *Finite State Machine* (FSM) pada *Non-Player Character* (NPC), pengujian fungsional menggunakan *Harness Test*, serta evaluasi pengalaman bermain menggunakan *Game Experience Questionnaire* (GEQ), penulis memberikan beberapa saran yang dapat dijadikan sebagai acuan dalam pengembangan penelitian selanjutnya maupun pengembangan game sebagai berikut:

1. Pengembangan Perilaku NPC

Implementasi FSM pada penelitian ini menggunakan lima *state*, yaitu *Idle*, *Patrol*, *Chase*, *Attack*, dan *Dead*, yang mampu menghasilkan perilaku NPC sesuai dengan kondisi permainan. Untuk penelitian selanjutnya, disarankan menambahkan *state* atau mekanisme transisi yang lebih kompleks sehingga NPC dapat memberikan respons yang lebih adaptif dan meningkatkan variasi perilaku NPC selama permainan berlangsung.

2. Pengembangan Metode Kecerdasan Buatan

Penelitian ini menerapkan metode FSM sebagai pengendali perilaku NPC. Pada penelitian selanjutnya, FSM dapat dikombinasikan dengan metode kecerdasan buatan lainnya, sehingga NPC tidak hanya mengikuti aturan yang telah ditentukan, tetapi juga mampu mengambil keputusan secara lebih dinamis berdasarkan kondisi permainan.