

BAB I

PENDAHULUAN

1.1 Latar Belakang

Game aksi merupakan *game* yang membutuhkan respon yang cepat dari pemain dengan dipadukan oleh koordinasi antara mata dan tangan (Promsutipong and Kotrajaras, 2017). Genre ini memiliki tempo pertarungan yang cepat dengan menekankan pada tantangan fisik dan kecepatan refleks yang tinggi (Saputra and Taurusta, 2022). Gerakan yang aktif dan juga banyaknya adegan pertarungan membuat *game* aksi memiliki animasi yang cukup rumit, sehingga membutuhkan manajemen pergerakan animasi yang baik agar animasi berjalan sesuai harapan (Mustofa, Sidiq, 2018).

Animasi merupakan salah satu dari elemen visual yang sangat penting bagi *video game*. *Game* menjadi lebih hidup dan nyata dengan hadirnya animasi (Rivaldi, Insanudin and Susanti, 2020). Animasi dalam *video game* tidak hanya satu gerakan, tetapi memiliki beberapa gerakan yang harus disatu padukan menjadi sebuah sistem pada objek tertentu (Li *et al.*, 2020). Terutama pada karakter pemain yang memiliki masukan secara langsung dari pemain. Animasi yang menjadi hasil dari keluaran harus memiliki gerakan yang sesuai dengan masukan yang diberikan, sehingga perpindahan animasi cenderung lebih rumit (Mustofa, Sidiq, 2018).

Penelitian yang dilakukan (Hidayat, Rachman and Azim, 2019) yang berjudul “Penerapan *Finite State Machine* pada *Battle Game* Berbasis *Augmented Reality*” dengan hasil bahwa salah satu metode yang bisa digunakan untuk membuat

sistem kontrol animasi yaitu *Finite State Machine (FSM)*. *FSM* diterapkan pada karakter 3D dalam *augmented reality* berbasis marker. Didapatkan bahwa pengendalian animasi diterapkan menggunakan *FSM* sebagai strategi *action animation prediction* untuk menentukan desain awal aksi pada animasi yang dibuat ketika karakter masuk ke dalam keadaan bertarung.

Penelitian lain yang mengambil metode yang sama dalam kendali animasi yaitu yang dilakukan oleh (Febrianto and Fatimah, 2022) berjudul “Penerapan Metode Finite State Machine Game 2d Adventure Kebokicak Dan Surontanu Berbasis Android” mendapatkan hasil penerapan *FSM* pada agen cerdas untuk kendali animasinya. *FSM* pada agen cerdas memiliki *state* yang akan berpindah berdasarkan deteksi jarak karakter pemain. Kendali animasi berhasil dijalankan dan dapat berjalan dengan baik, perilaku ditampilkan dengan menjalankan animasi yang sesuai dengan interaksi pemain.

FSM adalah metode yang diterapkan untuk membuat sistem pergerakan adaptif, yang mampu mengenali dan merekam tindakan pemain, apa yang dilakukan pemain dan apa tidak dilakukan pemain, dan kemudian memutuskan sebuah tindakan (Adimah, 2019), terbukti berhasil diterapkan dan berjalan dengan baik pada kendali animasi pada kedua penelitian tersebut. Menurut (Naharu *et al.*, 2021) pada penelitiannya yang berjudul “Penerapan *Hierarchical Finite State Machine* untuk Pengambilan Keputusan *Non-Player Character* (Studi Kasus: Gim *Hack and Slash*)” menyebutkan bahwa tidak menutup kemungkinan *FSM* masih memiliki kekurangan, yaitu sistem yang semakin rumit seiring bertambahnya *state* dan transisi yang diakibatkan oleh tingkah laku yang begitu banyak. Kekurangan

tersebut dapat dicegah dengan menggunakan *Hierarchical Finite State Machine (HFMS)*.

HFMS memiliki *super-state* atau *clustering*, yaitu menggabungkan beberapa *state* menjadi satu *state* (Mahendranta *et al.*, 2019). *HFMS* sebagai metode pengembangan dari *FSM* dapat menyederhanakan proses komputasi dan menjaga batasan dengan meminimalkan sejumlah bit pengkodean dan menyesuaikan kode untuk keadaan (Fauzi *et al.*, 2019). Melakukan perubahan dalam *HFMS* hanya perlu mengedit pada tingkat hirarki tertentu ke bawah. *HFMS* melibatkan hierarki di mana *FSM* bersarang di dalam *FSM* lainnya. Jumlah transisi dapat dikurangi dengan mengelompokkan *state* bersama dengan transisi keluar yang sama (Jagdale, 2021). Berdasarkan pemaparan tersebut maka akan dilakukan penerapan *HFMS* pada kontrol animasi karakter 2D pemain dalam *game* yang sudah dikembangkan. *HFMS* akan diintegrasikan pada antarmuka *animator* yang ada dalam Unity. *Animator* yang merupakan antarmuka bawaan Unity bertugas sebagai pengendali animasi pada sebuah objek dengan mengakses skrip *HFMS* pada objek karakter pemain. Kontrol animasi yang diterapkan dengan metode *HFMS* akan diproses kemudian transisi antar *state* akan berlangsung berdasarkan masukan yang dilakukan oleh pemain. *Clustering* yang ada dalam *HFMS* akan membuat sistem lebih rapih dan mudah untuk dipelihara. Berdasarkan uraian tersebut *HFMS* akan diimplementasikan sebagai pengendali animasi karakter pemain dan diukur kinerjanya menggunakan parameter *Frame Rate*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan masalah yaitu :

1. Bagaimana menerapkan *HFSM* sebagai pengendali animasi untuk karakter pemain dalam game aksi 2D?
2. Bagaimana mengukur kinerja *HFSM* yang diterapkan pada karakter pemain sebagai pengendali animasi?

1.3 Batasan Masalah

Adapun beberapa batasan masalah pada penelitian ini, yaitu:

1. Pengendali animasi akan diimplementasikan untuk karakter pemain pada *game* bergenre aksi yang sudah dikembangkan berbasis *Desktop Windows 10*.
2. Kontrol animasi pada pemain sekaligus sebagai kontrol perilaku pemain menggunakan masukan dari *keyboard* dan *mouse*.
3. Bahasa pemrograman yang digunakan untuk menerapkan algoritma yaitu *C#*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini, yaitu:

1. Menerapkan *HFSM* untuk pengendalian animasi menggunakan *game engine* Unity dan mengintegrasikannya dengan antarmuka *animator* yang merupakan pengendali animasi bawaan dari Unity.

2. Mengukur kinerja *HFSM* menggunakan parameter *Frame Rate* dengan cara membandingkan rata-rata *Frame Rate* yang dihasilkan *HFSM* dengan *FSM* kemudian dihitung persentase kenaikan kinerja dari algoritma tersebut.

1.5 Manfaat Penelitian

Beberapa manfaat yang didapat dari penelitian ini berdasarkan latar belakang yang telah diuraikan, antara lain:

1. Metode yang diterapkan dapat membantu dalam proses pengembangan *video game* dari sisi penggabungan teknis dengan artistik, dalam melakukan kontrol animasi dan perilaku pada karakter.
2. Meminimalisir *bug* yang diakibatkan oleh alur percabangan animasi yang banyak dengan memanfaatkan *clustering* dalam *HFSM*.
3. Mengetahui kinerja dari *HFSM* sebagai pengendali animasi karakter pemain.
4. Sebagai pembanding dan dasar metode yang lebih efisien untuk kontrol animasi bagi penelitian berikutnya.