BAB 1
PENDAHULUAN

1.1. Latar Belakang Masalah

Game Arcade adalah salah satu jenis *game* yang mengandalkan ketangkasan pemain dalam memegang kontrol [1]. *Game* jenis ini biasanya mempunyai *level* yang singkat, kontrol yang mudah, dan mempunyai *gameplay* yang sederhana. *Game Arcade* yang akan dibangun merupakan *game* kejar-kejaran sambil bertarung. Pada *game* perkelahian dan kejar-kejaran tentunya memerlukan rintangan yang baik untuk membuat *game* tersebut menarik. Selain bentuk arena kejar-kejaran yang rumit, NPC (*Non Playable Character*) musuh harus dapat bereaksi sesuai dengan keadaan atau kondisi yang terjadi di dalam permainan.

Game yang serupa yaitu *game "Tom & Jerry in House Trap"* yang ada pada PS One yang mengambil latar cerita seperti pada kartun *"Tom & Jerry"* [2]. Dimana karakter kucing dan tikus saling kejar-kejaran di dalam rumah dan bisa menyerang satu sama lain hingga *health point* lawannya habis. *Game "Tom & Jerry in House Trap"* tersebut merupakan perpaduan dari jenis *game* labirin dan *game* perkelahian. Pada game tersebut pergerakan NPC dalam mengejar pemain masih kurang tepat, karena NPC hanya akan mengejar pemain apabila pemain sudah terlihat (masuk jangkauan penglihatan NPC) dan apabila pemain tidak terlihat maka NPC akan bergerak secara random menuju ruangan lain. Selain itu perilaku NPC nya pun masih monoton, terkadang ketika NPC sudah memiliki senjata tidak langsung menyerang pemain yang ada dihadapannya tetapi malah berputar-putar untuk mencari senjata yang lain atau mencari jebakan.

NPC musuh harus bisa mengenali arena dengan baik sehingga bisa mencari senjata dengan cepat, memasang jebakan, kabur dari pemain ketika tidak memiliki senjata, dan mengejar pemain ketika memiliki senjata. Sehingga NPC musuh tersebut membutuhkan suatu metode algoritma yang bisa membantu dalam mencari jalur untuk pergerakannya dan juga untuk menentukan keputusan aksi yang perlu dilakukannya.

Jump Point Search adalah algoritma pencarian jalan ke suatu target dimana algoritma bergerak dengan sangat cepat di *uniform-cost grid maps* dengan

melompati banyak tempat yang dirasa perlu untuk dilewati [3]. Algoritma ini adalah hasil pengembangan dari algoritma A* dengan menambahkan fungsi *jump points* ketika melakukan *path-finding*. *Jump Point Search* dapat digambarkan dalam dua aturan pemangkasan sederhana yang diterapkan secara rekursif selama pencarian, aturan khusus untuk langkah lurus dan untuk langkah diagonal. Kunci penting dalam kedua kasus ini adalah untuk memangkas set tetangga dekat di sekitar *node* dengan mencoba untuk membuktikan bahwa ada jalur optimal dari parent *node* saat ini untuk setiap tetangga. Berdasarkan penelitian sebelumnya [3] diketahui bahwa algoritma *Jump Point Search* yang diterapkan pada permainan dengan jenis labirin seperti Pacman mampu menghasilkan rata-rata jumlah *visited node* terbaik dibandingan dengan algoritma A*, IDA* dan PEA*. Dengan hasil rata-rata *visited node* untuk algoritma Jump Point Search 11,7, algoritma IDA* 15,57, algoritma A* 27,8, dan algoritma PEA* 28,63. Algoritma *Jump Point Search* merupakan algoritma yang sederhana namun efektif, optimal tanpa menggunakan banyak memori, dan sangat cepat performanya.

*Finite State Machine*s (FSM) adalah sebuah metode perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja suatu sistem dengan yang menggunakan tiga hal, yaitu : *state*, *event* dan *action* [4]. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri (misal interupsi *timer*). Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif rumit. *Finite State Machine* ini dapat digunakan pada NPC sebagai pembuat keputusan sehingga NPC bisa bekerja dengan maksimal. Berdasarkan penelitian sebelumnya [4] diketahui bahwa metode *Finite State Machine* yang diterapkan pada game dengan jenis simulasi *Role-Playing Game* (RPG) The Relationship dapat menghasilkan respon NPC yang berbeda-beda menyesuaikan dengan kondisi yang terjadi di dalam permainan dan semakin banyak respon atau perilaku yang diterapkan pada NPC maka semakin menarik pula permainannya karena dapat menghasilkan alur cerita yang lebih menarik.

Oleh karena itu dalam penelitian ini algoritma *Jump Point Search* akan digunakan untuk diimplementasikan pada NPC musuh sebagai AI yang berfungsi untuk menentukan jalur pergerakannya dan *Finite State Machine* sebagai pembuat keputusan.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka dapat dirumuskan pokok permasalahannya adalah bagaimana menggunakan algoritma *Jump Point Search* dan metode *Finite State Machine* sebagai kecerdasan buatan kepada NPC musuh dalam memilih jalur pergerakannya dan sebagai pembuat keputusan.

1.3. Maksud dan Tujuan

Maksud dari penelitian tugas akhir ini adalah untuk penelitian pada Implementasi algoritma *Jump Point Search* dan *Finite State Machine* pada NPC musuh dalam *game arcade*.

Adapun tujuan dari penelitian tugas akhir ini adalah sebagai berikut:

1. Mengukur performa penggunaan memori pada algoritma *Jump Point Search* dalam mencari jalur terpendek.
2. Metode *Finite State Machine* dapat digunakan sebagai kecerdasan buatan kepada NPC musuh sebagai pembuat keputusan dalam pergerakan atau aksi yang akan dilakukan oleh NPC sesuai dengan kondisi yang ada.

1.4. Batasan Masalah

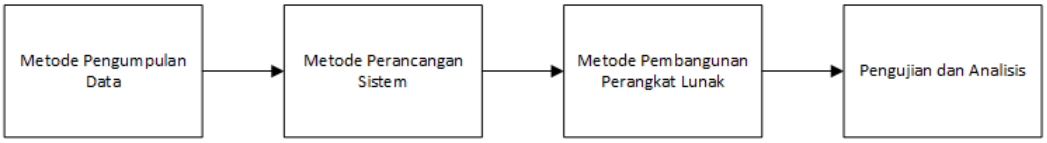
Untuk menghindari pembahasan yang meluas, maka penulis membatasi permasalahan-permasalahan dalam penelitian ini sebagai berikut:

1. Pada game yang akan dibangun NPC musuh akan melakukan pencarian jalur terpendek menuju pemain, senjata, dan jebakan.
2. Untuk simulasi proses pencarian jalur terpendek akan ditampilkan dalam bentuk *minimap* permainan.
3. NPC musuh akan memiliki perilaku untuk memasang jebakan, mengambil senjata, mengejar pemain, dan menyerang pemain.

4. Parameter yang akan digunakan untuk metode FSM yaitu : *health point*, jarak, *damage*, senjata musuh, jebakan, kecepatan, dan *durability*.
5. Permainan merupakan satu lawan satu (pemain lawan NPC musuh).
6. Ada tiga tingkat kesulitan pada *game* yang akan dibangun, yaitu *easy*, *medium*, dan *hard*.
7. *Game* menggunakan grafik 2,5 dimensi.

1.5. Metodologi Penelitian

Metode penelitian yang akan digunakan dalam penelitian tugas akhir ini adalah metode penelitian kualitatif. Metode penelitian kualitatif lebih mengacu kepada data-data yang diperoleh dari lapangan dan bersifat *real*.



Gambar 1.1 Metodologi Penelitian

1.5.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini adalah studi literatur. Hal tersebut dikarenakan oleh jenis data, dan maksud serta tujuan dari penelitian tugas akhir ini. Studi literatur adalah pengumpulan data dengan cara pengumpulan literatur, jurnal dan bacaan yang ada kaitannya dengan permasalahan yang terjadi.

1.5.2 Metode Perancangan Sistem

Pada tahap ini berisi perancangan antarmuka dan konsep *game* yang akan dibangun. Pada tahap ini juga akan dilakukan analisis kebutuhan algoritma dan metode pada *game* yaitu, bagaimana cara kerja algoritma *Jump Point Search* sebagai penentu jalur terpendek pada NPC dan bagaimana cara kerja metode *Finite State Machine* sebagai penentu prilaku NPC.

1.5.3 Metode Pembangunan Perangkat Lunak

Pada tahap pembangunan perangkat lunak yang dilakukan adalah mengimplementasikan algoritma *Jump Point Search* dan *Finite State Machine* ke dalam *game*.

1.5.4 Pengujian dan Analisis

Setelah algoritma *Jump Point Search* dan *Finite State Machine* diimplementasikan ke dalam *game*, maka dilakukan pengujian terhadap sistem yang telah dibuat dan menganalisis hasil implementasi algoritma *Jump Point Search* dan *Finite State Machine* pada sistem.

1.6. Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran secara umum mengenai permasalahan dan pemecahannya. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini membahas mengenai latar belakang , identifikasi masalah, maksud dan tujuan, batasan masalah, metode penelitian, serta sistematika penulisan untuk menjelaskan pokok – pokok pembahasannya.

BAB 2 LANDASAN TEORI

Pada bab ini akan menjelaskan mengenai objek dari penelitain, dan teori – teori pendukung yang berhubungan dengan pembangunan sistem yaitu, *game arcade*, *artificial intelligence*, algoritma A*, algoritma *Jump Point Search*, metode *Finite State Machine*, OOP (*object oriented programming*), UML, dan C#.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas mengenai analisis dan perancangan *game* yang dilakukan. Analisis bagaimana penerapan algoritma *Jump Point Search* dan *Finite State Machine* pada *game* yang akan dibangun. Dalam tugas akhir ini analisis dan kebutuhan perangkat lunak dibangun dengan konsep *object oriented programming*.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi mengenai implementasi dari analisis yang telah dilakukan, serta pengujian dari implementasi algoritma *Jump Point Search* dan *Finite State Machine* yang telah dibuat.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi mengenai kesimpulan yang diperoleh dari hasil pengujian implementasi algoritma *Jump Point Search* dan *Finite State Machine* berdasarkan tujuan yang ingin dicapai sebelumnya serta saran dari masalah-masalah yang ditemukan selama proses penelitian untuk pengembangan sistem ke depan.