

BAB I

PENDAHULUAN

1.1 Latar Belakang

Artificial intelligence (AI) dewasa ini telah menjadi garis depan dalam industri *game*. Hampir semua *game* membutuhkan AI untuk membuat komputer tampak cerdas (Sutojo *et al.*, 2011). Salah satu objek pada *game* yang dapat disematkan AI adalah *non-playable character* (NPC) yang umumnya bertindak sebagai musuh di dalam *game*. AI membantu NPC *game* untuk dapat menjadi lebih realistis di dalam lingkungannya, seperti bergerak dan mencari rute optimalnya sendiri (*pathfinding*). *Pathfinding* memiliki dua macam sifat target pencarian, yaitu statis dan dinamis. Topik *pathfinding* ini cukup menarik perhatian peneliti dari berbagai bidang untuk menjadikannya sebagai objek penelitian.

Dari penelitian 5 tahun terakhir, mayoritas peneliti menggunakan target *pathfinding* yang bersifat statis. Hal ini menyebabkan aplikasi maupun *game* yang dibangun menjadi konstan dan dapat dengan mudah diprediksi. Dalam beberapa penelitian seperti yang dilakukan oleh Wibowo *et al.* (2011) serta Widodo dan Hadiq (2013), *game pathfinding* dibangun dengan tujuan menerapkan algoritma pencarian tertentu, namun nyatanya algoritma pencarian tidak dimanfaatkan secara maksimal karena hanya berfungsi sebagai fitur bantuan yang dapat dipilih pengguna untuk menyelesaikan *game*. Algoritma pencarian bahkan menjadi tidak berfungsi jika pengguna tidak memilih untuk menggunakan fitur bantuan tersebut. Untuk menghindari permasalahan tersebut terulang kembali, penelitian yang baru dapat dilakukan dengan menggunakan target bersifat dinamis pada *game pathfinding* dan menerapkan algoritma pencarian kepada NPC agar berfungsi maksimal. Salah satu *game* yang dapat digunakan adalah *game Space Spider 3D*.

Game Space Spider adalah *game* 3D yang dibangun menggunakan *tools* Unity dengan desain lingkungan berbasis labirin dan target *pathfinding* yang bersifat dinamis. *Game* dilengkapi dengan karakter *player* dan *non-player* (NPC) sebagai musuh yang melakukan pencarian jalur menuju *player* yang bergerak

dinamis sesuai masukan pengguna. NPC akan memanfaatkan algoritma pencarian jalur optimum sebagai penentu arah gerak menuju target selama *game* berjalan.

Pencarian jalur optimum membutuhkan algoritma pencarian yang tepat (Suyanto, 2011). Terdapat dua teknik pencarian, yaitu pencarian buta (*blind*) dan pencarian terbimbing (*heuristic*) (Witanti dan Rahayu, 2013). Pencarian buta umumnya tidak memenuhi kriteria *optimality*, sedangkan pencarian terbimbing dapat memenuhi kriteria *completeness* dan *optimality* (Suyanto, 2011). Untuk mendapatkan jalur optimum, maka akan digunakan teknik pencarian terbimbing.

Teknik pencarian terbimbing terdiri dari beberapa algoritma seperti *Generate and Test*, *Hill Climbing*, *A**, *Greedy*, *Branch and Bound*, *Brute Force*. Algoritma *Greedy* tidak memperhatikan konsekuensi pemilihan solusi (Zulhidayanti dan Yulianti, 2013). Algoritma *Generate and Test* memakan waktu untuk pengujian setelah penemuan solusi (Welianto *et al.*, 2011). Algoritma *Brute Force* tidak optimal karena membutuhkan jumlah langkah yang besar (Widodo dan Hadiq, 2013). Secara tradisional algoritma yang digunakan dalam *game pathfinding* adalah algoritma *A**. Algoritma *A** berhasil menarik perhatian ribuan peneliti. Sulit menemukan algoritma yang lebih baik dari *A** (Cui dan Shi, 2011).

Berdasarkan latar belakang di atas, maka perlu dibangun *game pathfinding Space Spider 3D* yang dengan target pencarian dinamis. Teknik pencarian yang digunakan adalah teknik pencarian terbimbing (*heuristic*) dengan algoritma *A**. Penelitian dilakukan untuk menciptakan *game pathfinding* yang interaktif, tidak mudah diprediksi, serta dapat menampilkan hasil pencarian jalur optimal oleh algoritma *A** secara *real time* selama *game* berjalan.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka rumusan masalah dalam penelitian ini secara rinci adalah sebagai berikut.

1. Bagaimana membangun sebuah *game pathfinding Space Spider 3D* dengan target pencarian yang bersifat dinamis?
2. Bagaimana memanfaatkan algoritma *A** dalam melakukan *pathfinding* pada *game Space Spider 3D* secara maksimal selama *game* berjalan?

1.3 Batasan Masalah

Untuk menjaga agar fokus penelitian tidak keluar dari tujuan utama, maka terdapat beberapa batasan masalah sebagai berikut.

1. *Game* berbasis *desktop*, tidak terhubung ke jaringan LAN ataupun internet.
2. *Game* terdiri dari 3 zona permainan (labirin) dengan tingkat kesulitan (level) yang berbeda namun dengan konfigurasi karakter yang sama dalam batasan waktu 60 hingga 120 detik yang berbeda pada tiap zona permainan.
3. NPC berjumlah 3 buah tanpa karakteristik khusus.
4. Hasil akhir *game* hanya menang atau kalah, tanpa perhitungan skor.
5. *Game* hanya menunjukkan jalur optimal hasil pencarian algoritma untuk pergerakan NPC, namun tidak membahas waktu tempuh dan memori yang digunakan untuk proses pencarian.

1.4 Tujuan dan Manfaat Penelitian

Tujuan dan manfaat yang ingin dicapai dalam melakukan penelitian dan pengembangan sistem *game pathfinding Space Spider 3D* ini, secara terperinci dijelaskan sebagai berikut.

1.4.1 Tujuan penelitian

Berdasarkan latar belakang dan permasalahan yang ada, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Membangun sebuah *game pathfinding Space Spider 3D* yang interaktif dan tidak dapat dengan mudah diprediksi, dengan menggunakan target pencarian yang bersifat dinamis.
2. Menerapkan algoritma A* pada NPC *game Space Spider 3D* untuk menentukan arah pergerakan NPC saat berinteraksi dengan *player* sebagai target pencarian secara *real time* selama *game* berjalan.

1.4.2 Manfaat penelitian

Berdasarkan permasalahan dan tujuan di atas, penelitian ini dilaksanakan dengan harapan dapat memberi beberapa manfaat sebagai berikut.

1. *Game Space Spider 3D* dapat menunjukkan performa algoritma A* dalam melakukan pencarian dengan target yang bersifat dinamis, serta *game* menjadi lebih interaktif dan tidak mudah diprediksi.
2. Algoritma A* dapat dimanfaatkan secara maksimal pada NPC *game* dalam melakukan pencarian jalur optimal sebagai penentu arah gerak NPC saat berinteraksi dengan *player* secara *real time* selama *game* berjalan.

1.5 Metodologi Penelitian

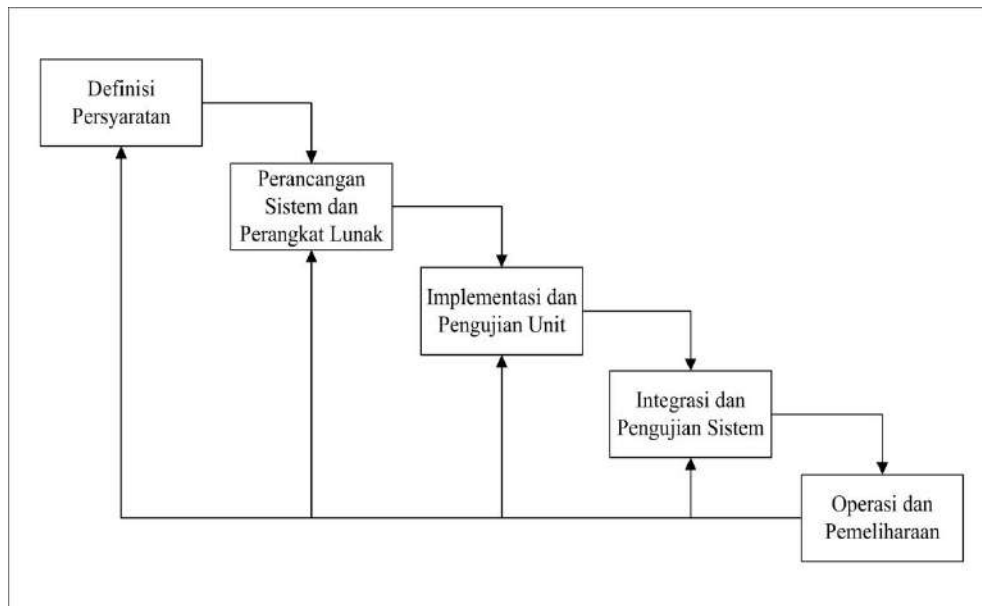
Metodologi penelitian yang digunakan dalam melakukan penelitian dan pengembangan sistem *game pathfinding Space Spider 3D* ini, secara terperinci dijelaskan sebagai berikut.

1.5.1 Jenis penelitian

Berdasarkan klasifikasi penelitian berdasarkan tujuan dan manfaatnya, penelitian ini merupakan sebuah penelitian terapan. Penelitian ini dilakukan untuk menerapkan dan menguji teori *pathfinding* dengan algoritma A* berdasarkan permasalahan keterbatasan dan kelemahan penelitian-penelitian sebelumnya. Penelitian dilakukan dengan menggunakan metode eksperimen lab dalam membangun sebuah *game* berjudul *Space Spider 3D*. *Game* berbasis *desktop* yang dibangun menggunakan *tools* Unity dan MonoDevelop. Algoritma A* diterapkan pada NPC musuh dalam *game* dengan target pencarian bersifat dinamis.

1.5.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode/model air terjun atau yang lebih dikenal dengan model *waterfall*. Model *waterfall* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis. Model ini mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, kemudian merepresentasikannya sebagai fase-fase proses berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan pemeliharaan (Sommerville, 2003). Ilustrasi model *waterfall* dapat dilihat pada Gambar 1.1.



Gambar 1.1 Model Waterfall
(Sumber: Sommerville, 2003)

Berikut merupakan penjelasan mengenai fase-fase sistematis tersebut menurut Sommerville (2003).

1. Definisi Persyaratan

Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan pengguna sistem. Persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

Tahap ini dilaksanakan tanpa konsultasi dengan pengguna, melainkan dengan melakukan pencarian informasi mengenai spesifikasi minimum sebuah *game* secara umum dengan batasan sistem ditentukan sesuai kebutuhan penelitian.

2. Perancangan Sistem dan Perangkat Lunak

Proses perancangan sistem membagi persyaratan dalam sistem. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi sistem perangkat lunak yang mendasar.

Tahap ini dilaksanakan dengan memodelkan sistem, merancang alur cerita, peraturan dan prosedur permainan, pengumpulan gambar, audio, dan objek (*asset*) pada *unity store*, serta mendesain tata letak objek dalam lingkungan *game* sehingga membentuk sebuah labirin sebagai area permainan.

3. Implementasi dan Pengujian Unit

Pada tahap ini, perancangan perangkat lunak direalisasikan dan diuji sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

Tahap ini dilaksanakan dengan merealisasikan desain kedalam bentuk kode program dalam bahasa *Javascript* dan *C#* menggunakan *tools* MonoDevelop.

4. Integrasi dan Pengujian Sistem

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi.

Tahap ini dilaksanakan dengan melakukan pengujian *black box* pada fungsionalitas sistem, pengujian *white box* pada struktur kontrol desain prosedural, dan pengujian statistik terhadap *game*.

5. Operasi dan Pemeliharaan

Pemeliharaan mencakup korelasi dari berbagai *error* yang ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan baru ditambahkan.

Tahap ini tidak akan dilaksanakan karena pengoperasian *game* tidak dilakukan secara publik sehingga tidak akan terjadi perubahan eksternal dan tidak membutuhkan proses pemeliharaan.

Metode *waterfall* baik dan umum digunakan karena memiliki berbagai keunggulan. Metode ini menuntut pengembang bekerja disiplin. Setiap tahapan hanya dapat dilaksanakan saat tahap sebelumnya telah selesai (Pressman, 2002).

Pemilihan metode *waterfall* dalam pembangunan *game Space Spider* ini didasari oleh berbagai pertimbangan, salah satunya karena metode *waterfall* sering dibahas dan digunakan dalam berbagai penelitian, sehingga banyak penelitian yang dapat dijadikan referensi. Metode *waterfall* bersifat klasik, mudah dipahami, dan memiliki tahapan dan alur yang jelas sehingga proses pembangunan sistem dapat dilakukan secara disiplin. Metode *waterfall* dapat menghindari kesalahan estimasi waktu pengembangan sistem. Hal ini cukup penting karena waktu pengembangan sistem ini tergolong singkat.

1.6 Sistematika Penulisan

Penulisan laporan dilakukan secara sistematis menggunakan beberapa bab. Gambaran umum mengenai isi laporan penelitian secara keseluruhan diuraikan secara singkat dengan sistematika sebagai berikut.

BAB I PENDAHULUAN

Bab ini meliputi latar belakang permasalahan yang diangkat dalam penelitian, perumusan masalah, tujuan dan manfaat penelitian, metodologi penelitian, serta sistematika penulisan laporan penelitian.

BAB II LANDASAN TEORI

Bab ini meliputi teori-teori dasar yang digunakan sebagai landasan dan referensi dalam melakukan penelitian. Teori tersebut mencakup teori mengenai kecerdasan buatan, *game*, metode pencarian, serta algoritma pencarian yang digunakan dalam penelitian.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini meliputi analisis kebutuhan sistem pada *game Space Spider* yang akan dibangun, beserta pemodelan UML, desain arsitektur sistem, dan desain antar muka sistem.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini meliputi pembahasan dari proses pembuatan sistem, seperti realisasi desain dalam tampilan antar muka berbasis 3D, hasil penerapan algoritma pencarian, serta pengujian fungsionalitas sistem.

BAB V KESIMPULAN DAN SARAN

Bab ini meliputi kesimpulan yang dapat diambil dari penelitian yang dilakukan, serta saran-saran yang berguna untuk penelitian dan pengembangan sistem di masa yang akan datang.