BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Artificial Intelligence (AI) agen adalah fitur standar *game* komputer modern, baik sebagai lawan, teman atau tutor dari pemain. Agar tampil otentik, agen tersebut harus mampu menavigasi dunia permainan dan memiliki kemampuan seperti mengejar, melarikan diri, mengawal pemain atau kemampuan lainnya. Semua perilaku ini memerlukan *pathfinding*. Algoritma *pathfinding* menghitung rute terpendek antara dua lokasi yang diberikan, menggunakan representasi grafik struktur dunia *game*. Meskipun ada algoritma yang mapan, seperti Dijkstra dan A*, yang menghasilkan solusi yang optimal, banyak faktor yang mendorong penelitian untuk memperbaiki kinerja baik dari segi waktu eksekusi dan konsumsi memori.

Algoritma harus cukup cepat untuk menjalankan secara *real-time* beberapa agen, jika tidak agen harus berdiri diam sampai pencarian rute selesai. Algoritma *pathfinding* yang paling sering digunakan adalah A*, yang terbukti optimal untuk pencarian [1], walaupun A* adalah algoritma pencarian yang baik, algoritma A* menghabiskan banyak *resources*. Pada peta besar, ratusan atau bahkan ribuan *node* mungkin akan berada dalam *open* dan *closed list*, yang dapat memakan memori lebih banyak dari yang tersedia pada sistem dengan memori yang terbatas, seperti pada perangkat *mobile* yang biasanya memiliki memori lebih kecil dibandingkan dengan perangkat desktop [2]. Untuk mengatasi ini dipilih algoritma yang cocok, salah satunya adalah SMA* (*Simplified Memory-Bounded* A*).

SMA* merupakan pengembangan dari algoritma A* yang mengatasi masalah *storage problem* dengan meletakkan *limit* tetap pada ukuran *open list*. Ketika sebuah *node* baru diproses, jika memiliki nilai yang lebih besar dari setiap *node* yang ada di *list*, maka akan dibuang. Jika tidak, maka akan ditambahkan, dan *node* yang sudah dalam *list* dengan nilai terbesar dihapus [3]. Dengan ini

diharapkan algoritma SMA* dapat menghemat penggunaan memori yang harus dihemat pada perangkat *mobile*, sehingga *game* dapat berjalan dengan lancar dengan penggunaan memori yang minimum.

Berdasarkan latar belakang yang telah dijelaskan di atas maka tugas akhir ini diberi judul yakni ” IMPLEMENTASI ALGORITMA SMA* PADA *GAME* TPS *MONSTER NEST* BERBASIS *MOBILE*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan yang menjadi masalah dalam penelitian ini adalah bagaimana cara mengatasi konsumsi memori pada A* yang menghabiskan banyak *resources* pada *game* dalam pencarian rute untuk NPC (*Non Playable Character*).

1.3 Maksud dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan skripsi ini adalah mengimplementasikan algoritma SMA* pada *game* TPS *monster nest*.

Adapun tujuan yang akan dicapai dalam penelitian ini, yaitu:

1. Mengurangi penggunaan *resources* dengan menggunakan algoritma SMA* yang lebih sedikit dalam penggunaan memori dibandingkan dengan algoritma A* pada *game monster nest*.
2. Mengimplementasikan algoritma SMA* pada *pathfinding game Third Person Shooter monster nest.*

1.4 Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka batasan masalah yang dibuat adalah sebagai berikut :

1. Penggunaan memori pada algoritma SMA* dilihat dengan menghitung jumlah *node* yang ada di *open list* dan *closed list*.
2. *Game* ini memiliki 3 *level* yaitu *level easy*, *level medium* dan *level hard*.
3. *Game* ini berbasis *mobile*.

4. Implementasi algoritma SMA* pada *game* ini diterapkan pada NPC (*non-playable character*) pada *pathfinding*.
5. *Game* ini bergenre TPS (*Third Person Shooter*).
6. *Game* ini hanya dapat dimainkan secara *single player*.
7. *Game* ini dikembangkan dengan menggunakan IDE Unity 3D Pro 3.5.
8. Menggunakan bahasa pemrograman *JavaScript*/*UnityScript*.
9. Perancangan yang digunakan dalam membangun aplikasi ini adalah pendekatan berbasis objek.

1.5 Metodologi Penelitian

Berikut ini adalah suatu metodologi dalam penelitian tugas akhir dengan judul "Implementasi algoritma SMA* pada *game* TPS *Monster Nest* berbasis *mobile*", akan ditempuh langkah-langkah sebagai berikut :

Tahap-tahap Metode yang digunakan dalam pengerjaan laporan penelitian ini adalah:

1. Tahap Pengumpulan data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Pustaka

Studi Pustaka yang dilakukan ialah dengan mempelajari berbagai literatur, seperti buku-buku, artikel-artikel, *e-book*, *website*, dan sumber-sumber yang berkaitan dengan *game* yang akan dibangun, meliputi kecerdasan buatan, desain, *tools* dan juga pemodelan dengan UML.

b. Observasi

Teknik pengumpulan data dengan mengadakan pengamatan langsung dan mengindra terhadap objek atau proses yang di jadikan objek pemasalahan seputar *game* yang dibangun berupa masalah algoritma kecerdasan buatan, desain, *tools* dan juga pemodelan dengan UML.

2. Tahap pembuatan perangkat lunak

Model yang akan digunakan pada kasus ini adalah model *Waterfall* yang dirumuskan oleh Ian Sommerviell di dalam bukunya *Software Engineering* (Rekayasa Perangkat Lunak) [4]. Model ini mengusulkan sebuah pendekatan kepada perkembangan *software* yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut

a. System Engineering

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, manusia, dan *database*. Rekayasa dan anasisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

b. Analysis

Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

c. Design

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.

d. Coding

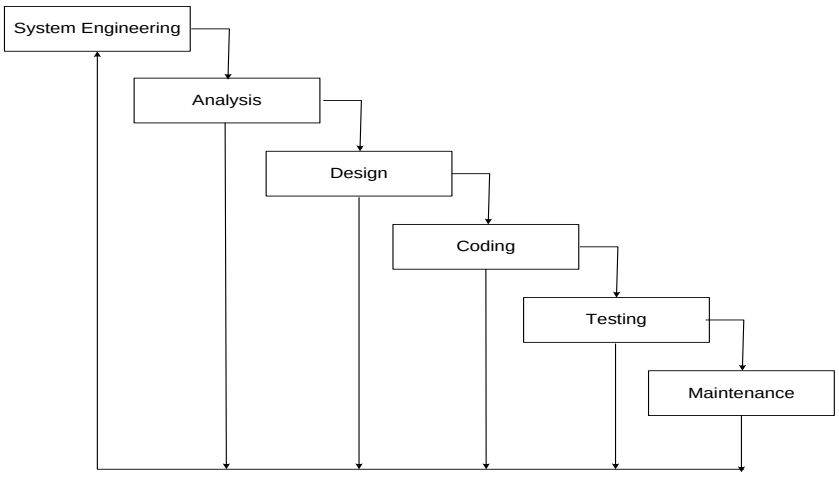
Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

e. Testing

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika *internal software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

f. Maintenance

Software akan mengalami perubahan setelah disampaikan kepada pelanggan (pengecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan yang ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat *peripheral* atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.



Gambar 1.1Diagram Metode *Waterfall* [4]

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian yang kemudian diikuti dengan pembatasan masalah, serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Membahas berbagai konsep dasar teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan yaitu tentang pembuatan aplikasi *game*, kecerdasan buatan yang dalam hal ini algoritma A*, SMA*, OOP (*object oriented programming*), UML (*unified modeling language*), unity 3D.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini membahas analisis terhadap *game* yang dibuat serta bagaimana merancang dengan pemodelan UML, merancang tampilan *game*, serta analisis bagaimana penerapan algoritma SMA* pada *game* TPS *monster nest* yang akan diterapkan pada NPC (*non playable character*) untuk pencarian jalan.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Berisi tentang tahapan-tahapan yang dilakukan untuk menerapkan sistem yang telah dirancang. Melakukan pengujian terhadap aplikasi dan algoritma yang telah dibangun dengan metode pengujian *black box* dan *white box* serta pengujian perbandingan algoritma A* dengan SMA* dengan program simulasi.

BAB 5 KESIMPULAN DAN SARAN

Berisi kesimpulan hasil dari pengujian dan implementasi SMA* berdasarkan tujuan yang ingin dicapai sebelumnya dan memberikan masukan atau saran dari masalah-masalah yang ditemukan selama proses penelitian terhadap algoritma SMA*.