

# Pembuatan Mekanisme Strategi Perang pada *Game Majapahit's Conquest*

Jimmy Muliawan<sup>1</sup>, Gregorius Satia budhi.<sup>2</sup>, Liliana<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Telp (031) – 2983455, Fax. (031) - 8417658

jimmy\_758595@yahoo.com, greg@petra.ac.id<sup>2</sup>, lilian@petra.ac.id<sup>3</sup>

## ABSTRAK

Dalam pembuatan media visual yang mendukung minat pembelajaran sejarah Indonesia, *game based learning* merupakan sebuah solusi untuk masalah ini. *Game* yang dibuat harus dapat mencakup aspek-aspek pada sejarah sehingga dapat mencapai tujuan awal dari pembuatan *game*. Oleh karena itu, dalam penelitian ini, *game* akan dibuat untuk mencakup salah satu aspek politik yaitu peperangan. *Game* yang akan dibuat akan mengambil aspek politik pada zaman Kerajaan Majapahit dan diberi nama *Majapahit's Conquest*.

*Game* yang dibuat mengambil tema *Turn Based Tactical* dimana *player* dan komputer akan menjalankan *troops* secara bergiliran. Terdapat juga 3 event penting yang dibuat secara khusus sebagai wadah untuk visualisasi strategi perang yang dipakai pada zaman Majapahit, yaitu Pemberontakan Ra Kutu, Ekspedisi Bali, dan Perang Bubat. *AI* yang dipakai untuk menjalankan *troops* dari komputer menggunakan 2 metode yaitu *Fuzzy State Machine* dan *Minimax*. *Game* ini akan dibuat menggunakan *Unity Engine* dengan scripting menggunakan bahasa *C#* dengan *MonoDevelop* sebagai *IDE*.

Pengujian dilakukan pada sistem yang mendukung strategi perang seperti penyerangan dan cara berpindah tempat. Dari hasil pengujian tersebut sistem ini dipengaruhi oleh variabel-variabel yang telah dikonfigurasi pada setiap *troops*.

**Kata Kunci:** *Fuzzy State Machine, Minimax, Game Based Learning, Sejarah Indonesia, Majapahit's Conquest*

## ABSTRACT

*In order to make a visual media that can make student to take interest in history, game based learning is one of the solutions. The game first must have all aspect of the history itself in order to fulfill the goal. In this research, game that can cover the politic aspect especially war is made. The game will cover the politic aspect of Majapahit Kingdom and entitled Majapahit's Conquest.*

*This game use Turn Based Tactical as the genre which player and computer side moves his troops turn by turn. 2 methods of AI will be used in this game, which are Minimax and Fuzzy State Machine. This game will be made with Unity Engine, and scripting language will be done with C# as the language and MonoDevelop as IDE.*

*The experiment is conducted to check the game system that support the war strategy of the game such as attack command or move/run command. The result of the experiment declare that all configured variables in troops influence the system.*

**Keywords:** *Fuzzy State Machine, Minimax, Game Based Learning, Indonesian History, Majapahit's Conquest.*

## 1. PENDAHULUAN

Pelajaran sejarah merupakan salah satu pelajaran yang mempunyai peran untuk mengajarkan siswa di sekolah nilai patriotisme, nilai nasionalisme dan menghargai negara sendiri [1]. Namun pelajaran sejarah ini juga yang menjadi salah satu pelajaran yang dihindari oleh siswa sekolah karena susah dimengerti dan kompleks sehingga nilai tersebut tidak tersampaikan dengan baik [4]. Pembelajaran sejarah di Indonesia hanya dapat ditemui dalam bentuk buku pelajaran dan kurang media pendukung pembelajaran Sejarah Indonesia sehingga mereka cenderung lebih meminati sejarah luar negeri yang telah terekspos dengan media pendukung yang juga bersifat komersial.

*Game* komputer dapat menjadi salah satu media alternatif dalam pembelajaran Sejarah Indonesia lewat visualisasi yang disediakan oleh *game* yang dibuat. *Game* yang akan dibuat mengambil topik sejarah kerajaan pada zaman Majapahit. Dengan menggunakan *game* komputer sebagai media untuk memberikan visualisasi, orang yang memainkan *game* ini tidak hanya mendapatkan ilmu pengetahuan saja namun juga mendapatkan hiburan, sehingga unsur *game* yang ada juga tidak dihilangkan dengan materi sejarah yang dibawa. *Game* ini juga perlu dibuat sebagai media pembelajaran alternatif untuk sejarah Indonesia dan memberikan visualisasi sejarah Indonesia melalui media yang memiliki banyak peminat

## 2. LANDASAN TEORI

### 2.1 Sejarah Indonesia Zaman Majapahit

Sejarah Kerajaan Majapahit tidak lepas dari pengaruh Gajah Mada, seseorang yang pada awalnya menjabat sebagai panglima perang Bhayangkara hingga menjadi Mahapatih pada masa pemerintahan Ratu Tribhuwanatunggadewi. Gajah Mada mulai menata karirnya lewat keberhasilannya menyelamatkan Raja Jayanegara dan mengambil kembali istana dari Pemberontakan Ra Kutu. Pada tahun 1343, Kerajaan Majapahit dibawah pimpinan Gajah Mada dan bantuan dari Arya Damar berhasil menaklukan Kerajaan Bendahulu (Bali), Sumbawa, dan Lombok. Setelah itu penaklukan pada wilayah Kalimantan juga dilakukan oleh Kerajaan Majapahit [3].

Pada tahun 1350 M. Hayam Wuruk naik takhta menggantikan ibunya untuk memerintah Majapahit. Pada masa pimpinannya, Kerajaan Majapahit menaklukan daerah Timur seperti Logajah, Gurun, Sukun, Taliwung, Sapi, Gunungapi, Seram, Hutankadali, Sasak, Bantayan, Luwu, Buton, Banggai,

Kunir, Galiyan, Muar, Wandan, Ambon, Timor, Dampo. Pada tahun 1357 M, Perang Bubat terjadi dan membunuh keluarga kerajaan Sunda Galuh .

## 2.2 Game Desain by Gamification

Gamification adalah proses dimana menggunakan *game thinking* dan *game mechanic* untuk menyelesaikan masalah dan berinteraksi dengan user [2].

Untuk menciptakan sebuah game dengan gamification yang tepat, diperlukan sebuah game elemen yang dapat diurutkan dari level konkret hingga abstrak. Level-level tersebut adalah:

- *Interface design pattern*
- *Game design pattern* atau *Game mechanic*
- *Design principles*
- *Conceptual model of game design unit*
- *Game design method* dan *design process*

Dengan pembagian level diatas, sebuah game yang akan dibuat dapat diposisikan hingga game elemen yang akan dibuat dapat disesuaikan sehingga game yang dibuat dapat memberikan *output* yang diharapkan secara maksimal.

## 2.3 Turn Based Tactical

*Turn-Based Tactical* adalah *subgenre Turn-Based* dimana *player* bermain dengan bergantian dalam 1 *map*. Dalam 1 giliran, *player* dapat menggerakkan beberapa pion untuk berjalan atau menyerang. *Game* akan selesai jika *objective* dari *game* telah terpenuhi. Contoh *objective* adalah seperti mengalahkan semua musuh yang ada atau mengalahkan jendral musuh. Contoh *game* dengan genre ini adalah *Elven Legacy*, *X-Com* dan *Guardian of Graxia* (Gambar 1).

*AI (Artificial Intelligence)* adalah salah satu elemen penting dalam membuat *game* dengan genre ini. *AI* yang terlalu mudah maupun yang terlalu susah akan membuat *player* menjadi cepat bosan karena *game* yang dimainkan menjadi tidak menarik. Metode *AI* yang paling umum digunakan adalah *Minimax* atau *Rule Based System*. Metode *AI* tersebut memiliki kesamaan dimana sebelum melakukan sebuah pergerakan, *AI* akan memikirkan terlebih dahulu prioritas dan pola gerakan mana yang lebih menguntungkan.



Gambar 1. Tampilan game dengan genre Turn-Based Tactical

## 2.4 Fuzzy Logic

*Fuzzy Logic* adalah konsep yang ditemukan pada tahun 1965. Algoritma ini dapat menerjemahkan pernyataan rumit dari bahasa sehari-hari menjadi formula matematika. *Fuzzy Logic* telah banyak dikembangkan di berbagai negara, terutama di Jepang untuk mengatur beberapa alat elektronik seperti mesin cuci buatan Matsushita, Televisi buatan Nissan dapat mengatur sendiri kontras, warna yang dikeluarkan.

Ada beberapa langkah yang diperlukan oleh *Fuzzy* untuk memproses suatu kegiatan, yaitu membuat menentukan *fuzzy set*,

*memberships function*, membuat *fuzzy rules*, membuat *degree of fuzzy membership function*, melakukan defuzzifikasi.

Tabel 1. Tabel Fuzzy Set untuk Margin HP

Nama	Jarak
<i>PlayerHigh</i>	-100-(-30)
<i>Even</i>	-25-25
<i>EnemyHigh</i>	30-100

*Fuzzy Set* digunakan menjabarkan secara jelas *fuzzy range* yang akan digunakan. Contoh dari *Fuzzy Set* dapat dilihat pada Tabel 1. Dari contoh pada Tabel 1, ketika Margin HP bernilai -80 maka dapat dikatakan bahwa nilai tersebut termasuk *PlayerHigh*, ketika Margin HP bernilai 27 maka nilai dapat dikatakan *Even* dan *EnemyHigh*

*Fuzzy Rule* adalah *rule* yang ditulis dengan menggunakan bentuk *if.then*. *Rule* ini akan digunakan untuk memproses beberapa *fuzzy input*. Contoh *fuzzy rule* adalah seperti ini:

- if MarginHP is *Even* and JarakAntarTroops is *Medium* then Move is *Far*
- if MarginHP is *Even* and JarakAntarTroops is *Near* then Move is *Short*
- if MarginHP is *AttackerHigh* and JarakAntarTroops is *Medium* then Move is *Middle*

*Degree of Membership Function* digunakan untuk menentukan nilai akhir yang akan dimasukkan dalam matrix untuk dihitung/fuzzifikasi. Sebagai contoh dari hasil input dari *fuzzy input* adalah -75 untuk *MarginHP* dan 6 untuk jarak antar objek. Untuk mengerjakan langkah selanjutnya yang harus dilakukan pertama kali adalah *fuzzification*. *Fuzzification* adalah proses merubah crisp value dari *fuzzy input* menjadi *fuzzy word*.

Defuzzifikasi atau mendapatkan nilai crisp pada output dapat dilakukan dengan metode *Centroid*. *Centroid* adalah cara mendapatkan nilai crisp output dilihat dari nilai tengah pada grafik.

## 2.5 Fuzzy State Machine

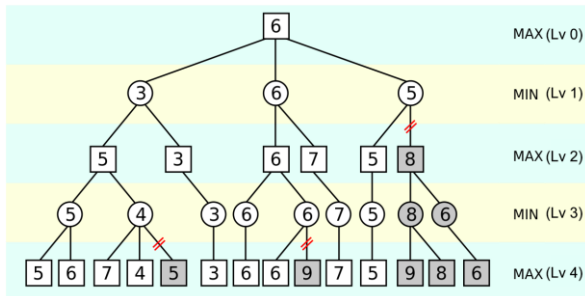
*Fuzzy State Machine* adalah metode *AI* yang menggunakan prinsip dari logika *fuzzy* dalam mengambil sebuah keputusan. *Fuzzy State Machine* menggunakan dasar dari *Finite State Machine* yang digabung logika *fuzzy* sehingga menghasilkan *state machine* yang lebih variatif dengan state yang lebih sedikit dari *Finite State Machine*. *AI* ini biasa digunakan oleh *object* untuk menentukan senjata apa yang digunakan atau memilih melakukan apa sesuai dari grafik *fuzzy* yang dipakai oleh *object*. Cara *AI* ini melakukan penerjemahan grafik menjadi sebuah perintah adalah mengubah *crisp* yang telah didesain menjadi *fuzzy rules* dan diterjemahkan lagi menjadi perintah yang dapat dibaca oleh *AI* [6].

## 2.6 Minimax

*Minimax* adalah metode *AI* yang biasa digunakan untuk membuat *game* strategi. Cara kerja dari *AI* ini adalah dengan membuat *tree* kemungkinan untuk setiap langkah yang dapat dilakukan oleh *AI* dan dihitung bobot yang diperlukan (Gambar 2). Dari hasil kemungkinan-kemungkinan tersebut, *AI* akan memilih langkah terbaik berdasarkan dari bobot terbaik yang ada. Untuk sebuah

game berbasis *Turn-Based Strategy*, pembuatan node pada tree *Minimax* dibatasi agar AI tidak berpikir terlalu lama [5].

*Alpha Beta Pruning* adalah salah satu metode *Minimax* yang dapat digunakan untuk mengurangi *roots* yang tidak terpakai sehingga cara kerja *Minimax* menjadi lebih cepat. Dalam prosesnya AI tidak akan memasuki *root* yang memiliki nilai lebih kecil atau lebih besar dari nilai *alpha/beta* yang ada pada *depth* yang dimasuki oleh AI [7].



Gambar 2. Tampilan kerja Minimax

### 3. DESAIN SISTEM

#### 3.1 Desain Game

Game yang akan dibuat bernama “Majapahit’s Conquest”. Game ini memiliki tujuan untuk memberikan realisasi peristiwa-peristiwa sejarah yang ada pada zaman Majapahit melalui peristiwa-peristiwa yang dibuat dalam game. Game ini mengambil latar belakang cerita tentang perkembangan Kerajaan Majapahit dalam menguasai nusantara.

Pada game ini terdapat 3 event dengan *storyline* yang berbeda yang dapat diakses. Event yang diambil didasarkan pada literatur-literatur yang telah diteliti dan memiliki informasi yang lengkap sehingga dapat memberikan komponen yang lengkap saat diadaptasi pada game.

Event pertama adalah Pemberontakan Ra Kutu. Peristiwa ini diambil karena merupakan salah satu pemberontakan besar yang membahayakan Kerajaan Majapahit pada zaman pemerintahan Raja Jayanegara. Event kedua adalah cerita tentang Ekspedisi Bali. Kerajaan Majapahit dipimpin oleh Gajah Mada mulai melakukan penyerangan di wilayah Bali. Serangan tersebut dilakukan dari 3 arah. Pada event ini, *player* harus memenangkan ketiga peperangan tersebut untuk menguasai Kerajaan Bali. Event ketiga adalah cerita tentang perang Bubat. Peristiwa ini terjadi karena kesalahpahaman yang menyebabkan keluarga Kerajaan Sunda tersinggung dan terjadi peperangan di lapangan Bubat. Pada event ini *player* memiliki *objective* untuk mengalahkan keluarga kerajaan dalam beberapa *turn*.

#### 3.2 Desain Game Komponen

Beberapa *game component* diperlukan untuk pembuatan game ini. *Game component* ini akan digunakan untuk menyusun game yang akan dibuat. *Game component* yang dipakai pada game ini meliputi:

- *Stage*: Merupakan tempat *player* menempatkan *troops*, berbentuk beberapa macam terrain seperti rawa (*swamp*), daratan (*land*), tepi pantai (*shore*), dan hutan (*forest*). Terrain tersebut terdiri dari beberapa panel yang berbentuk persegi. Setiap panel memiliki beberapa tipe dan effect yang berbeda

seperti pada panel lumpur, maka jarak gerak *troops* akan dibagi menjadi 2.

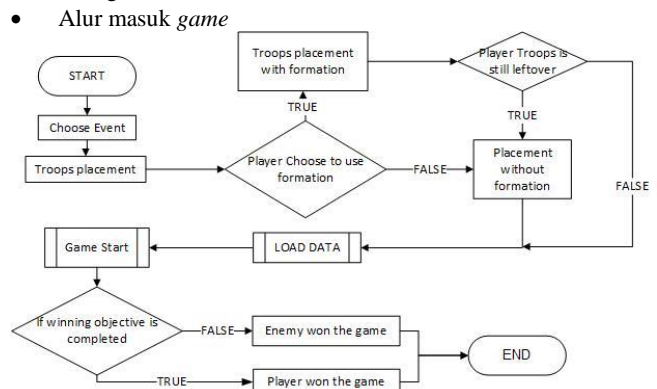
- *Environment*: *Environment* ini berupa object yang menempati map seperti pohon dan bangunan. *Troops* yang dijalankan oleh *player* tidak dapat menempati tempat yang sama dengan object tersebut.
- *Troops*: Sebutan untuk gabungan dari beberapa pasukan yang memiliki *job* yang sama. Agar *troops* yang dijalankan oleh komputer dapat mengambil keputusan dengan sendirinya, *troops* akan diberi AI agar dapat berpikir sendiri. Pada game ini, AI yang dipakai adalah *Minimax* dan *Fuzzy State Machine*. Dengan kombinasi dua AI tersebut, setiap game memasuki *enemy turn*, AI akan merancang strategi perang untuk melawan *player*.

Pada game ini, *troops* akan dibagi menjadi beberapa jenis sesuai dengan asal kerajaannya sesuai dengan literatur yang telah dipelajari. *Troops* dapat dibagi menjadi 2 berdasarkan jarak serang yang dimiliki yaitu *short-range* dan *long-range*. *Troops* yang memiliki jenderal perang didalamnya akan memiliki status yang lebih besar. Jenderal perang yang berbeda juga akan mempengaruhi jumlah *troops* yang dapat diberi perintah pada setiap turnnya. Untuk membedakan *troops-troops* tersebut, AI yang akan digunakan akan mengalami sedikit perubahan yaitu pada *membership function* dan *fuzzy rules* yang dipakai pada AI *Fuzzy State Machine*. Perubahan pada 2 elemen tersebut akan memberikan *effect* yang berbeda ketika *troops* akan mengambil keputusan sehingga menampilkan perbedaan. Untuk Kerajaan Majapahit, *troops* akan dibagi menjadi 3 yaitu *short sword* (keris) *troops* untuk tipe *short range*, *archer troops* untuk tipe *long range*, dan *spear troops* untuk tipe *long range*.

Untuk *Enemy Troops* ini terdiri dari 2 Kerajaan yang menjadi musuh dari Majapahit, yaitu Kerajaan Bali dan Kerajaan Sunda. Untuk Kerajaan Bali, *troops* dengan tipe *short range* adalah *Kris Troops*, *troops* dengan tipe *long range* adalah *Archer Troops* dan *Spear Troops*. Untuk Kerajaan Sunda, *troops* dengan tipe *short range* adalah *Whip Troops*, untuk tipe *long range* adalah *Archer Troops* dan *Spear Troops*. *Troops* yang memiliki pemimpin pasukan didalamnya akan memiliki status yang lebih besar.

#### 3.3 Desain Gameplay

Pada alur game ini, akan dibagi menjadi 2 bagian yaitu alur masuk ke game dan alur dari *gameplay* itu sendiri. Detail dari kedua bagian tersebut adalah:



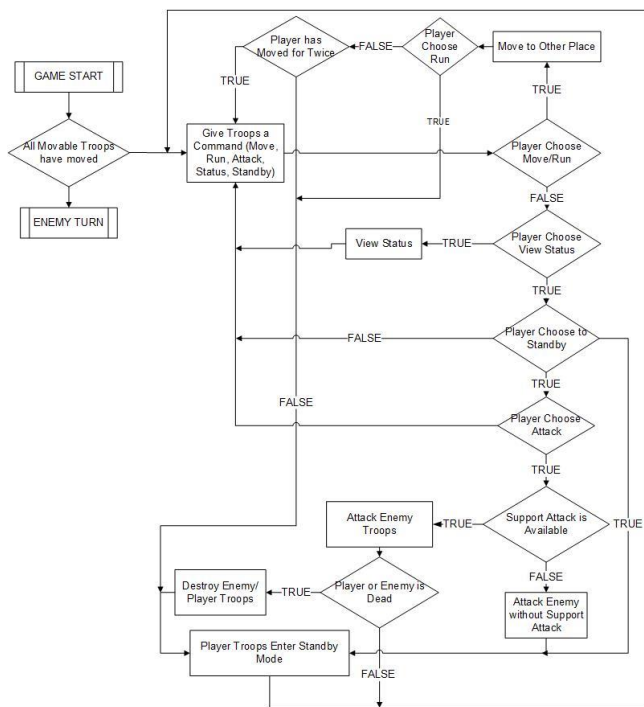
Gambar 3. Flowchart alur masuk game

Ketika memulai bagian *game* dari *free campaign* (Gambar 3), *player* memulai dari dari memilih event yang akan dimainkan. Pada setiap *event* komposisi dari *troops* yang akan dimainkan oleh *player* akan berbeda-beda. Dari *troops* yang telah dipilih, *player* dapat memilih untuk membuat formasi perang atau menempatkan pasukannya sesuai yang diinginkan oleh *player*.

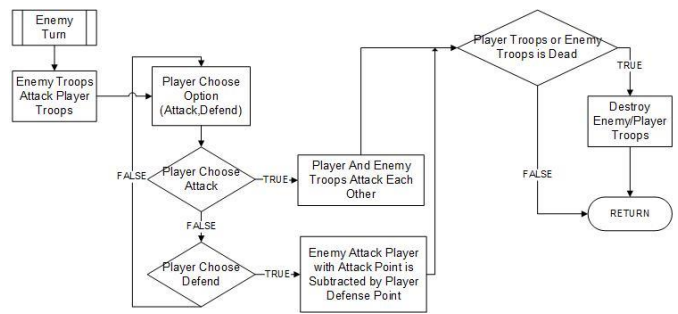
• Alur *gameplay*

Setelah *player* memasuki *game*, secara garis besar *gameplay* terbagi menjadi 2 bagian yaitu, *player turn* (Gambar 4) dan *enemy turn* (Gambar 6). Pada *player turn*, *player* dapat menggerakkan beberapa *troops* yang telah dipilihnya. Setiap *troops* dapat digerakkan dengan beberapa pilihan perintah yaitu *move*, *run*, atau *attack*. Pada dasarnya dapat bergerak 2 kali, dengan beberapa pola seperti *move-run*, *run-run*, *move-move*, *move-attack*, atau *run-attack*. Jika *player* telah memilih untuk menggunakan perintah *attack* maka *player* tidak dapat melakukan gerak lagi. Ketika *player* memilih perintah *attack* dan posisi *troops* yang dijalankan terdapat *troops* yang sama di dekatnya (sebelah kiri, kanan, atau bawah) dan *troops* tersebut dapat ikut menyerang, maka *troops* tersebut dapat melakukan *support attack*.

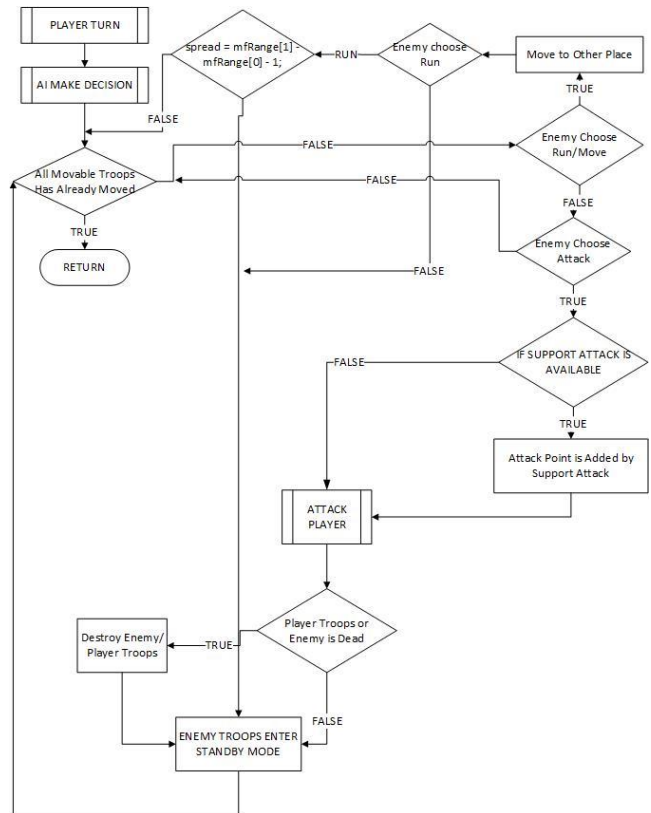
Pada *enemy turn*, *enemy* akan menjalankan *troops* dengan aturan yang sama seperti *player*. Hanya saja, *AI* yang akan menentukan *troops* untuk berpindah tempat atau menyerang jika jarak yang ada memenuhi. Ketika *enemy* memilih untuk menyerang *player* (Gambar 5), *player* diberi pilihan untuk bertahan atau ikut menyerang. Jika *player* tidak dapat melakukan penyerangan balik maka *player* dianggap memilih untuk bertahan.



Gambar 4. Flowchart alur *gameplay* *player turn*



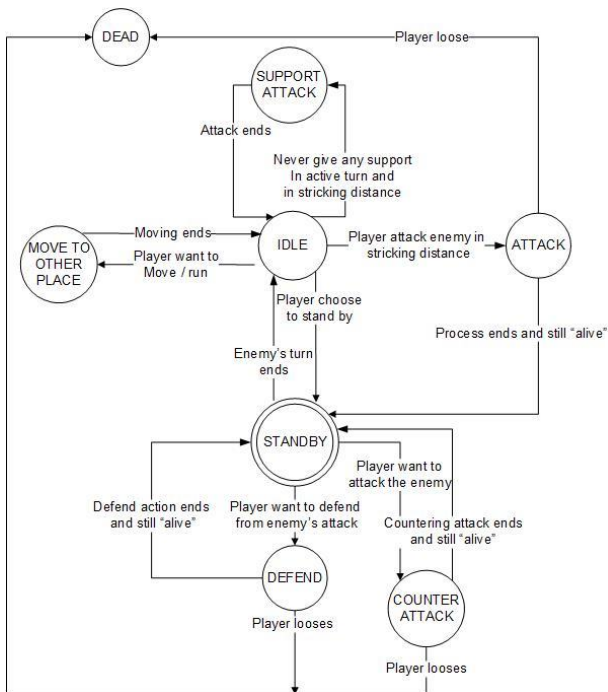
Gambar 5. Flowchart alur *gameplay* *enemy* menyerang *player*



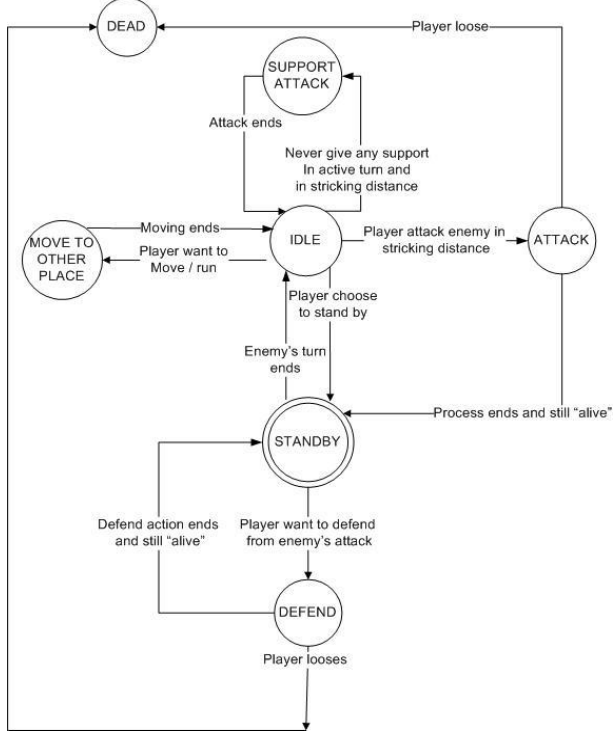
Gambar 6. Flowchart alur *gameplay* *enemy turn*

### 3.4 Desain Troops

Pada setiap *troops* terdapat beberapa *state* yang dimiliki oleh *troops*. *State* tersebut bergantung pada *job* yang dimiliki oleh *troops*. Pada dasarnya setiap *troops* dapat berpindah tempat dengan perintah *move/run* dan menyerang *troops* lain dengan perintah *attack*. *Troops* memasuki *state support attack* jika *troops* tersebut memang dapat melakukan *support attack* dan *troops* tersebut dipilih oleh *player*. Setelah *player* melakukan perintah, *player* memasuki *standby state*. Pada *state* ini, *player* tidak dapat berpindah tempat atau menyerang. *Troops* tetap memasuki *standby state* sampai *enemy turn* selesai. Pada *state* ini, *troops* ini dapat diserang oleh *troops* musuh. *Troops* dengan tipe *short range* (*short sword troops*) seperti pada Gambar 7 dapat melakukan penyerangan balik ketika diserang oleh *troops* musuh, namun *player* dengan tipe *long range* (*archer troops*) seperti pada Gambar 8 tidak dapat melakukan serangan balik sehingga hanya dapat bertahan. Ketika *troops* ini dianggap mati maka *troops* memasuki *dead state* dan tidak dapat lagi dipakai.



Gambar 7. Finite State Machine short range troops



Gambar 8. Finite State Machine long range troops

### 3.5 Desain AI

Cara kerja AI ini menggunakan 2 metode AI yaitu *Minimax* dan *Fuzzy State Machine*. AI akan dipakai sebagian besar pada waktu *enemy turn* untuk AI memikirkan perintah apa yang akan dilakukan, dan ketika musuh diserang oleh *player*, AI akan mengambil keputusan untuk bertahan atau menyerang balik. Pada *enemy turn* *Minimax* akan memilih *troops* mana saja yang akan

dijalankan dengan *Fuzzy State Machine* akan memikirkan perintah/*command* apa yang akan dijalankan.

## 4. PENGUJIAN

Pengujian yang dilakukan adalah pengujian sistem dan *state* pada setiap jenis *troops* berdasarkan jarak serang yang dimiliki, yaitu:

- Pengujian sistem

Pengujian ini dilakukan untuk mengetahui apakah alur game telah dijalankan atau tidak.



Gambar 9. Scene awal masuk game

Gambar 9 menunjukkan ketika *player* memasuki *stage* yang telah ditentukan. *Player* diminta untuk memilih apakah ia mau menempatkan *troops* menggunakan formasi yang telah ditentukan oleh *game* atau menempatkan sendiri *troops* yang dimilikinya.



Gambar 10. Scene setelah *player* memilih menggunakan formasi

Gambar 10 menunjukkan ketika *player* memilih menempatkan *troops* menggunakan formasi. Dengan menggunakan formasi, *player* tidak perlu menempatkan satu persatu *troops* yang ada.



Gambar 11. Scene setelah *player* memilih untuk menempatkan pasukan secara bebas

Gambar 11 menunjukkan bahwa player dapat menempatkan secara bebas pasukan yang telah dimilikinya. Terdapat *window* yang menunjukkan jumlah pasukan yang belum ditempatkan dan button yang dipencet untuk mengganti job *troops* yang akan ditempatkan.

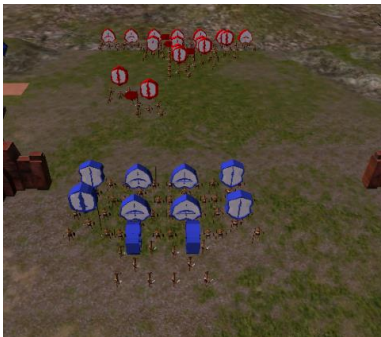


Gambar 12. Scene pada *player turn*

Setelah *player* menempatkan semua *troops* yang dimilikinya, *player* dapat menjalankan *troops* tersebut dengan beberapa perintah yang dapat diakses (*move*, *run*, *attack*) seperti pada Gambar 12.



Gambar 13. Scene ketika *player* mengakhiri gilirannya



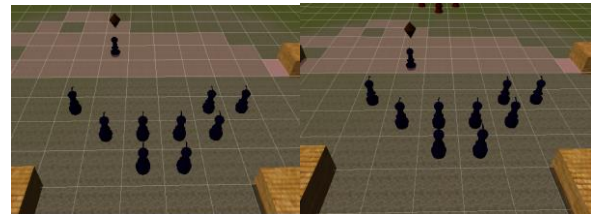
Gambar 14. Scene pada *enemy turn*

Ketika *player* telah menjalankan *troops* yang dapat dijalankan, *player* dapat mengakhiri gilirannya (Gambar 13) dan game akan memasuki *enemy turn* sehingga *enemy* dapat menggerakkan *troops* yang dimilikinya. *Troops* dan *command* yang akan dikerjakan dipilih oleh AI (Gambar 14).

- Pengujian *state* pada *troops*
- Pengujian ini dilakukan ketika *troops* diberi perintah baik dari *player* ataupun AI.



Gambar 15 Scene ketika *troops* memasuki state berpindah tempat



Gambar 16. Scene ketika *troops* telah berpindah tempat

Gambar 15 menunjukkan ketika *troops* diberi perintah untuk *move/run*, akan muncul tanda yang menandai *tile* yang dapat ditempati *troops*. Jika *player* memilih perintah *run* seperti pada Gambar 15 bagian (b), jarak pindah lebih besar jika dibandingkan perintah *move* (Gambar 15 (a)). Ketika *troops* telah mendapatkan koordinat kemana *troops* harus berpindah, maka *troops* akan berpindah tempat seperti pada Gambar 16. Jika *troops* memilih untuk menggunakan *command move* maka *troops* masih dapat menggunakan *command move* atau *attack* jika ada *troops* yang memasuki jarak serang. Jika *troops* memilih untuk menggunakan *command run* maka *troops* akan menempati *state standby* setelah berpindah tempat.



Gambar 17. Scene ketika *troops* menyerang *troops* lain

Gambar 17 menunjukkan ketika *troops* memasuki *state attack*. Jika ada *troops* yang dapat membantu penyerangan maka *attack point* yang diterima *troops* musuh akan bertambah (Gambar 17 (a)).

Ketika *troops* musuh tidak dapat melakukan penyerangan balik karena jarak yang tidak mencukupi atau *troops* tersebut termasuk tipe *long range*, maka *attack point* yang dimiliki akan menjadi 0 dan hanya dapat bertahan. Gambar 17 (b) menunjukkan ketika *troops* dapat melakukan penyerangan balik kepada *troops* yang menyerangnya.

## 5. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan beberapa kesimpulan yaitu:

- *State* yang dapat diakses *troops* bergantung pada tipe jarak serang yang dimiliki oleh *troops*.
- Perintah yang dijalankan oleh *enemy troops* dikerjakan oleh AI.

Pemilihan keputusan untuk menyerang balik atau bertahan dipengaruhi oleh jarak serang suatu troops dan tipe *troops* (*long range* atau *short range*)

## 6. REFERENSI

- [1] Azan. N, Zin M, Jaafar. A, Yue. W.S. 2009. *Digital Gamebased learning (DGBL) model and development methodology*. *WSEAS TRANSACTIONS on COMPUTERS*, 322-333.
- [2] Deterding. S, Dixon. D, Khaled. R, Nacke. L.E. 2011. *Gamification: Toward to Definition*. *CHI 2011*, 1-4.
- [3] Ilahi, M. T. 2009. *Gelegar sumpah Gajah Mada, sang proklamator Nusantara*. Yogyakarta: Garailmu.
- [4] Joseph, S. 2011. *What are Upper Secondary School Students Saying About History? Caribbean Curriculum*, 1-26.
- [5] Millington, I. 2006. *Artificial Intelligence for Games*. San Fransico: Elsevier.
- [6] Pirovano, M. 2012. *The use of Fuzzy Logic for Artificial Intelligence in Games*. 1-8.
- [7] Saffidine. A, Finnsson H, Buro. M. 2012. *Alpha-Beta Pruning for Games with Simultaneous Moves*. *Association for the Advancement of Artificial*, 1-7.