

Implementasi *Artificial Intelligence* pada *Game Defender of Metal City* dengan Menggunakan *Finite State Machine*

Billy, Imam Kuswardayan, Wijayanti Nurul Khotimah

Departemen Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: imam@its.ac.id

Abstrak—*Game* adalah jenis perangkat lunak yang populer dewasa ini. Pada *game* biasa terdapat *Artificial Intelligence* (AI) yang mengendalikan pihak komputer sehingga manusia yang bermain dapat merasakan seolah mereka sedang melawan pemain manusia lain saat bermain. Salah satu jenis *game* yang ada ialah *tower defense* dimana pemain perlu mempertahankan *tower* atau markas nya agar tidak dihancurkan oleh pihak lawan. Untuk membuat AI untuk *game* berjenis *tower defense*, diperlukan algoritma yang baik untuk mengatur gerakan pasukan lawan. Penelitian ini mengusulkan penerapan AI pada *game* berjenis *tower defense* menggunakan *rule based Finite State Machine*. Setiap pasukan memiliki AI dengan beberapa *state*, pada saat terjadi suatu kondisi tertentu, maka pasukan akan berpindah dari satu *state* ke *state* lainnya berdasarkan *rule* yang telah ditentukan, hanya ada satu *state* yang bisa aktif pada suatu waktu. Pada penelitian ini AI akan dibagi menjadi 3 level yaitu *easy*, *medium* dan *hard*. Pengujian dilakukan dengan mempertandingkan antar level AI pada semua kemungkinan tingkat kesulitan, lalu dibuat tabel derajat kemenangan atau *winning rate* dari AI yang ada. Berdasarkan hasil pengujian *Finite State Machine* dapat diterapkan pada *game* berjenis *tower defense* dengan baik untuk membentuk AI yang kuat.

Kata Kunci—*Game*, *Artificial Intelligence*, *Tower Defense*, *Unity*.

I. PENDAHULUAN

PERKEMBANGAN teknologi yang sangat pesat akhir-akhir ini membawa banyak perubahan dalam perkembangan perangkat lunak, salah satu jenis perangkat lunak yang sangat populer dewasa ini adalah *game*, *game* merupakan perangkat lunak yang digunakan sebagai sarana hiburan baik oleh kaum anak-anak ataupun orang dewasa.

Salah satu fitur yang umum dan dapat ditemui pada hampir semua *game* yang ada pada zaman sekarang adalah *Artificial Intelligence* atau AI, AI adalah kepintaran buatan yang dipasang pada sistem computer sehingga memungkinkan pemain untuk bermain melawan komputer dengan keadaan seolah-olah ia melawan pemain lain, membuat AI adalah suatu tantangan tersendiri bagi developer *game*.

Finite State Machine atau FSM adalah sebuah model yang digunakan untuk menggambarkan alur dari suatu proses berdasarkan *state* atau keadaan dari suatu proses. Hanya ada satu *state* yang bisa aktif pada suatu waktu sehingga dibutuhkan adanya perpindahan ke *state* lain apabila ingin melakukan aksi yang lain[1]. FSM biasa digunakan untuk mengatur dan

menentukan alur tindakan, yang berguna untuk mengimplementasikan AI dalam *game*.

Defender of Metal City adalah *game* berjenis *tower defense* yang dirancang dengan AI untuk meningkatkan antusiasme pemain dengan adanya lawan AI dengan tingkat kesulitan yang sepadan.

Pada penelitian ini akan mengimplementasikan AI pada *game* berjenis *tower defense* menggunakan *Finite State Machine*. Penelitian ini dibagi menjadi beberapa tahap. Tahap pertama, merancang *game* dengan *genre tower defense* dan aturan permainannya. Kedua, merancang strategi untuk memenangkan *game*. Ketiga, mengimplementasikan strategi untuk menang dan *Finite State Machine* pada AI serta perancangan *level* AI. Langkah terakhir, akan dilakukan uji coba terhadap kemampuan AI dengan mempertandingkan antar setiap kemungkinan tingkat kesulitan AI.

II. TINJAUAN PUSTAKA

A. *Artificial Intelligence*

Artificial Intelligence atau Kecerdasan Buatan adalah kecerdasan yang dimiliki oleh mesin yang dapat bertindak seperti layaknya manusia[1]. Pada *game*, AI bertindak dalam mengendalikan komputer agar dapat bermain melawan pemain manusia.

B. *Tower Defense*

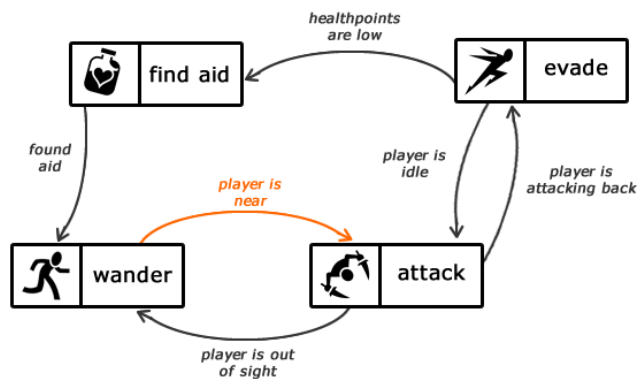
Tower defense adalah *sub-genre* dari *game real-time strategy* dimana tujuan utama pemain adalah melindungi *tower* utamanya dari serangan musuh dengan menempatkan pertahanan di sepanjang jalan menuju *tower* tersebut, apabila musuh berhasil menghancurkan *tower* milik pemain maka pemain akan kalah[2]. Beberapa contoh *game* berjenis *tower defense* antara lain adalah *Kingdom Rush* dan *Bloons TD*.

C. *Finite State Machine*

Finite State Machine atau FSM adalah sebuah model yang digunakan untuk menggambarkan alur dari suatu proses berdasarkan *state* atau keadaan dari suatu proses, hanya ada satu *state* yang bisa aktif pada suatu waktu sehingga dibutuhkan adanya perpindahan ke *state* lain apabila ingin melakukan aksi yang lain[3].

FSM biasa digunakan untuk mengatur dan menentukan alur tindakan, yang berguna untuk mengimplementasikan AI dalam

game. Contoh FSM yang digunakan sebagai AI dari musuh pada game dapat dilihat pada Gambar 1.



Gambar 1. Contoh Penggunaan FSM dalam AI pada Game

D. Unity

Unity adalah sebuah *Game Engine Cross-platform* yang dapat menghasilkan game untuk berbagai macam platform termasuk Windows, Mac, Linux, Android, iOS, dan banyak lagi. Unity pertama kali diterbitkan pada Tahun 2005 dan telah banyak berkembang sejak saat itu. Unity memiliki *Built-in IDE* yang bernama MonoDevelop.

III. DESAIN SISTEM

A. Spesifikasi Game

Pada game ini terdapat dua pihak yang saling menyerang satu sama lain, target nya adalah menghancurkan *tower* lawan sementara mempertahankan agar *tower* nya sendiri jangan sampai hancur, pihak yang *towernya* hancur akan kalah.

Pada game ini terdapat 4 jenis pasukan yang sama pada kedua belah pihak, yaitu *minion*, *warrior*, *ranger*, dan *artillery*. Pada game ini kedua belah pihak berusaha saling menyerang *Tower* milik pihak lawan.

Peraturan game adalah sebagai berikut:

1. Terdapat 2 belah pihak yang saling menyerang satu sama lain.
2. Terdapat 4 jenis pasukan berbeda pada setiap pihak yang memiliki kemampuannya sendiri-sendiri.
3. Terdapat *tower* pada kedua belah pihak yang apabila hancur maka pihak tersebut akan kalah.
4. *Tower* dapat menyerang pasukan musuh yang mendekati
5. Apabila membunuh pasukan milik musuh maka akan mendapat sejumlah uang.
6. Kedua pihak memiliki pilihan untuk meng-*upgrade* pasukan yang dimilikinya agar lebih kuat hingga *level 3*, dengan setiap *upgrade* membutuhkan biaya sejumlah uang.
7. Kedua belah pihak memiliki pilihan untuk membuat pasukan *elite* yaitu pasukan yang jauh lebih kuat dari pasukan biasa dengan membayar sejumlah uang.
8. Terdapat 3 tingkat kesulitan pada game yaitu *easy*, *medium*, dan *hard*, pemain dapat memilih akan melawan AI dengan tingkat kesulitan mana dan akan menggunakan tingkat kesulitan mana untuk AI pada pihak pemain itu sendiri.

Terdapat 4 jenis pasukan pada game ini yaitu:

1. Minion

Tipe pasukan terlemah, memiliki *damage* yang kecil, dapat bergerak dan menyerang dengan cepat, hanya dapat menyerang dari jarak dekat dan memiliki *health point* yang sedikit.

2. Warrior

Tipe pasukan yang bergerak dengan kecepatan sedang, memiliki *health point* yang cukup sehingga dapat menyerang lawan dari jarak dekat, dan juga memiliki *damage* dan kecepatan menyerang yang lebih besar daripada tipe *minion*.

3. Ranger

Tipe pasukan yang menyerang dari jarak jauh, memiliki *damage* dan *health point* yang lebih besar daripada tipe *minion* tetapi masih lebih kecil daripada tipe *warrior* sehingga tidak dapat menyerang dari jarak dekat, bergerak dan menyerang dengan kecepatan yang sama dengan tipe *warrior*.

4. Artillery

Tipe pasukan terkuat yang memiliki *area damage* sehingga dapat menyerang beberapa musuh secara bersamaan, memiliki *damage* dan *health point* paling besar diantara semua pasukan namun kecepatan serangan dan kecepatan Bergeraknya adalah yang paling lambat diantara semua tipe pasukan yang lain.

Pasukan akan diproduksi oleh *tower* secara periodik sehingga pasukan akan terus dihasilkan. Namun untuk pasukan *elite*, pemain harus mengendalikan kapan akan membeli pasukan *elite* yang mana karena pembelian pasukan *elite* harus menggunakan uang.

Pasukan dapat berjalan di *map* pada area yang bukan hutan. Pemain juga dapat mengendalikan pergerakan pasukan dengan kontrol menggunakan *mouse* untuk mengatur kemana pasukan harus pergi atau pasukan musuh mana yang harus diserang.

Kondisi kemenangan game adalah menghancurkan *tower* milik lawan. Kondisi kekalahan game adalah saat *tower* milik pemain dihancurkan oleh lawan.

B. Strategi Memenangkan Permainan

Strategi memenangkan game adalah dengan mengatur agar pasukan dapat seoptimal mungkin membunuh sebanyak mungkin musuh sehingga dapat terus maju sehingga pada akhirnya menyerang *tower* musuh. Juga harus diperhatikan pertahanan terhadap *tower* sendiri karena apabila *tower* hancur maka pemain akan kalah tidak peduli berapa banyak pasukan musuh yang telah dibunuhnya.

Karena setiap pasukan memiliki kemampuan yang berbeda-beda maka penggunaan setiap pasukan juga berbeda-beda:

1. Pasukan tipe *minion* memiliki *damage* yang tidak terlalu besar namun dapat bergerak cepat sehingga dapat dimanfaatkan untuk membunuh pasukan yang sudah lemah atau *health point* nya tersisa sedikit.
2. Pasukan tipe *warrior* dapat digunakan sebagai garis depan karena memiliki pertahanan dan *damage* yang lumayan sehingga dapat melawan musuh.

3. Pasukan tipe *ranger* dapat menyerang dari jarak jauh sehingga dapat digunakan sebagai *backup* dari belakang.
4. Pasukan tipe *artillery* dapat digunakan untuk melawan musuh dalam jumlah banyak karena dapat melakukan area *damage*.

Untuk strategi mengatur kapan melakukan *upgrade* pasukan dan *men-deploy* elite pasukan, *upgrade* harus dilakukan terlebih dahulu karena *upgrade* bersifat permanen, apabila semua pasukan telah di *upgrade* hingga level tertinggi, barulah uang yang tersisa digunakan untuk menghasilkan pasukan *elite*.

C. Penerapan Artificial Intelligence

Tahap penerapan *artificial intelligence* dibagi menjadi beberapa bagian, yaitu:

1. Penerapan AI pada Tower

Tower adalah bagian penting dalam *game* karena apabila *tower* hancur, pihak tersebut kalah, pada *tower*, hanya ada 2 keadaan, *idle* yaitu tidak melakukan apapun, atau *attack* yaitu menyerang dengan menembakkan panah pada saat ada musuh mendekat dengan jarak kurang dari sama dengan 30 meter.



Gambar 2. FSM pada Tower

FSM yang digunakan oleh *tower* dapat dilihat pada Gambar 2. Penjelasan untuk kondisi perpindahan *state* pada FSM pada *tower* dapat dilihat sebagai berikut:

- I1, A2. Tidak ada musuh dalam jarak 30 meter.
- I2, A1. Ada musuh dalam jarak 30 meter.

Semua angka jarak yang berbeda-beda didapat dari hasil percobaan. Dari beberapa kemungkinan angka yang telah dicoba telah disimpulkan bahwa angka – angka yang didapat adalah angka yang terbaik untuk menghasilkan AI *tower* yang kuat.

2. Penerapan Tingkat Kesulitan AI

Pada *game* ini terdapat 3 tingkat kesulitan AI, yaitu *easy*, *medium*, dan *hard*. Terdapat beberapa hal yang membedakan ketiga tingkat kesulitan tersebut.

Pada ketiga tingkat kesulitan yang berbeda, terdapat delay saat mengambil keputusan, pada tingkat kesulitan *easy*, pengambilan keputusan dilakukan setiap 50 *frame*. Pada tingkat kesulitan *medium*, pengambilan keputusan dilakukan setiap 25 *frame*. Pada tingkat kesulitan *hard*, pengambilan keputusan dilakukan setiap 1 *frame*.

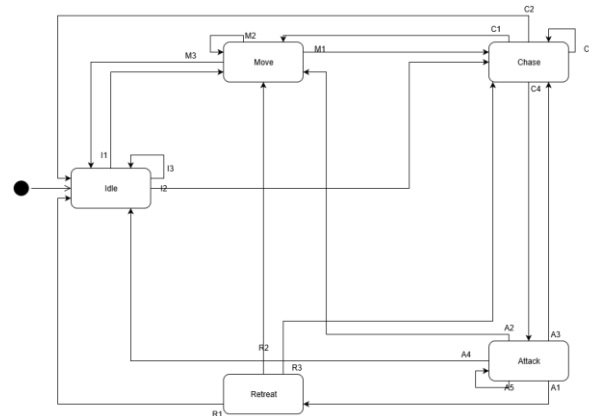
Pada ketiga tingkat kesulitan, juga terdapat kemungkinan pengambilan keputusan secara *random*, dimana AI akan bertindak secara *random* dan tidak mengikuti *rule-base* yang telah ditetapkan, semakin rendah tingkat kesulitan AI maka akan semakin tinggi kemungkinan pengambilan keputusan secara *random* tersebut. Pada tingkat kesulitan *hard*, tidak ada kemungkinan keputusan *random*. Pada tingkat kesulitan *medium*, terdapat 15% kemungkinan pengambilan keputusan

random. Pada tingkat kesulitan *easy*, terdapat 30% kemungkinan pengambilan keputusan *random*. Pada tingkat *medium*, terdapat 4 kemungkinan keputusan *random* yang dapat diambil, yaitu bergerak secara *random*, pulang kembali ke *tower*, mengikuti teman secara *random*, dan menyerang musuh secara *random*. Pada tingkat kesulitan *easy* hanya terdapat 3 kemungkinan keputusan *random*, yaitu bergerak secara *random*, pulang kembali ke *tower*, dan mengikuti teman secara *random*.

Pada bagian pengambilan keputusan untuk meng-*upgrade* pasukan atau *men-deploy* pasukan *elite*, juga terdapat perbedaan. Pada tingkat kesulitan *easy*, pengambilan keputusan dilakukan secara *random* total. Pada tingkat *medium*, akan dilihat apabila ada jenis pasukan yang belum mencapai level maksimal, maka ada 70% kemungkinan pasukan tersebut akan di-*upgrade* terlebih dahulu, dan 30% sisanya adalah kemungkinan mengeluarkan pasukan *elite* secara *random*. Pada tingkat kesulitan *hard*, pasukan akan selalu di-*upgrade* terlebih dahulu hingga mencapai level maksimal, barulah berikutnya akan mengeluarkan pasukan *elite* secara *random*.

Semua angka presentase kemungkinan dan jumlah *frame* untuk semua tingkat kesulitan yang berbeda-beda didapat dari hasil percobaan. Dari beberapa kemungkinan angka yang telah dicoba telah disimpulkan bahwa angka – angka yang didapat adalah angka yang terbaik untuk menghasilkan AI yang sesuai dengan tingkat kesulitannya.

3. Penerapan AI Pasukan Minion



Gambar 3. FSM pada Pasukan Minion

FSM pasukan *minion* dapat dilihat pada Gambar 3. Penjelasan untuk kondisi perpindahan *state* pada FSM pada pasukan *minion* dapat dilihat sebagai berikut:

- I1, M2, R2, C1. Salah satu kondisi berikut terpenuhi:
 - *Health point* lebih dari 25%, tidak ada musuh dalam jarak 20 meter.
 - *Health point* kurang dari sama dengan 25%, dalam jarak 20 meter ada 5 atau lebih teman.
- I2, M1, R3, C3. Salah satu kondisi berikut terpenuhi:
 - *Health point* lebih dari 25%, ada musuh tipe *ranger*.
 - *Health point* lebih dari 25%, tidak ada musuh tipe *ranger*.

- *Health point* kurang dari sama dengan 25%, dalam jarak 20 meter ada 5 atau lebih teman dan ada musuh dalam jarak 15 meter.

I3, R1, C2, M3. *Health point* kurang dari 25%.

C4. Jarak target kurang dari sama dengan 3 meter.

A1. Salah satu kondisi berikut terpenuhi:

- *Health point* kurang dari 25%, musuh tipe *minion* atau *warrior*, tidak ada 5 atau lebih teman tipe *minion* dalam 20 meter dan *health* kurang dari *health point* musuh.
- *Health point* pemain kurang dari 25%, musuh tipe *ranger*, *health point* musuh lebih dari sama dengan 50%.
- *Health point* pemain kurang dari 25%, musuh tipe *artillery*, *health point* musuh lebih dari *health point* pemain.

A2. Salah satu kondisi berikut terpenuhi:

- Target mati atau hilang, *health point* lebih dari 25%, tidak ada musuh dalam jarak 20 meter.
- Musuh mati atau hilang, *health point* kurang dari sama dengan 25%, dalam jarak 20 meter ada 5 atau lebih teman.

A3. Salah satu kondisi berikut terpenuhi:

- Target mati atau hilang, *health point* lebih dari 25%, ada musuh tipe *ranger*.
- Target mati atau hilang, *health point* lebih dari 25%, tidak ada musuh tipe *ranger*.
- Target mati atau hilang, *health point* kurang dari sama dengan 25%, dalam jarak 20 meter ada 5 atau lebih teman dan ada musuh dalam jarak 15 meter.

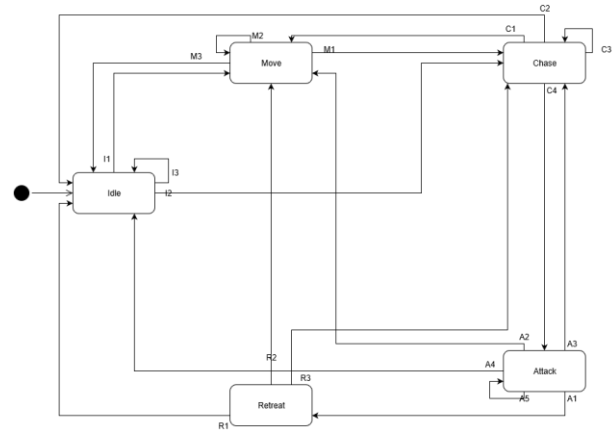
A4. Target mati atau hilang, *health point* kurang dari 25%.

A5. Salah satu kondisi berikut terpenuhi:

- *Health point* lebih dari 25%.
- *Health point* kurang dari sama dengan 25%, musuh tipe *minion* atau *warrior*, dalam jarak 20 meter. terdapat 5 atau lebih teman tipe *minion*.
- *Health point* kurang dari sama dengan 25%, musuh tipe *minion* atau *warrior*, *health point* musuh lebih kecil dari *health point* pemain.
- *Health point* pemain kurang dari 25%, musuh tipe *ranger*, *health point* musuh kurang dari 50%.
- *Health point* pemain kurang dari 25%, musuh tipe *artillery*, dan *health point* musuh kurang dari *health point* pemain.

Semua angka presentase *health point* dan jarak yang berbeda-beda didapat dari hasil percobaan. Dari beberapa kemungkinan angka yang telah dicoba telah disimpulkan bahwa angka – angka yang didapat adalah angka yang terbaik untuk menghasilkan AI pasukan *minion* yang kuat.

4. Penerapan AI Pasukan Warrior



Gambar 4. FSM pada Pasukan Warrior

FSM pasukan *warrior* dapat dilihat pada Gambar 4. Penjelasan untuk kondisi perpindahan *state* pada FSM pada pasukan *warrior* dapat dilihat sebagai berikut:

I1, R2, C1, M2. Salah satu kondisi berikut terpenuhi:

- *Health point* lebih dari 20%, tidak ada musuh dalam jarak 20 meter.
- *Health point* kurang dari sama dengan 20%, ada 5 atau lebih teman dalam jarak 20 meter dan ada musuh dalam jarak 20 meter, tidak ada musuh dalam jarak 20 meter.
- *Health point* kurang dari sama dengan 20%, ada 5 atau lebih teman tipe *minion* dalam jarak 20 meter, tidak ada musuh dalam jarak 20 meter.

I2, M1, R3, C3. Salah satu kondisi berikut terpenuhi:

- *Health point* lebih dari 20%, ada musuh dalam jarak 20 meter.
- *Health point* kurang dari sama dengan 20%, ada 5 atau lebih teman tipe *warrior* atau *ranger* dalam jarak 20 meter dan ada musuh dalam jarak 20 meter.
- *Health point* kurang dari sama dengan 20%, ada 5 atau lebih teman tipe *minion* dalam jarak 20 meter dan ada musuh dalam jarak 20 meter.
- *Health point* kurang dari 20%, tidak ada teman dalam jarak 20 meter, ada musuh tipe *minion* atau *ranger*.

I3, M3, R1, C2. *Health point* kurang dari 20%, tidak ada 5 atau lebih teman dalam jarak 20 meter, tidak ada musuh tipe *minion* atau *ranger*.

C4. Jarak target kurang dari sama dengan 3 meter.

A1. Salah satu kondisi berikut terpenuhi:

- *Health point* pemain kurang dari 20%, musuh tipe *warrior* atau *ranger*, tidak ada teman dalam jarak 15 meter, dan *health point* musuh lebih dari *health point* pemain.
- *Health point* pemain kurang dari 20%, musuh tipe *artillery*, tidak ada teman dalam jarak 15 meter, dan *health point* musuh lebih dari *health point* pemain.

A2. Salah satu kondisi berikut terpenuhi:

- Target mati atau hilang, *health point* lebih dari 20%, tidak ada musuh dalam jarak 20 meter.
- Target mati atau hilang, *health point* kurang dari sama dengan 20%, ada 5 atau lebih teman dalam jarak 20

meter dan ada musuh dalam jarak 20 meter, tidak ada musuh dalam jarak 20 meter.

- Target mati atau hilang, *health point* kurang dari sama dengan 20%, ada 5 atau lebih teman tipe *minion* dalam jarak 20 meter, tidak ada musuh dalam jarak 20 meter.

A3. Salah satu kondisi berikut terpenuhi:

- Target mati atau hilang, *health point* lebih dari 20%, ada musuh dalam jarak 20 meter.
- Target mati atau hilang, *health point* kurang dari sama dengan 20%, ada 5 atau lebih teman tipe *warrior* atau *ranger* dalam jarak 20 meter dan ada musuh dalam jarak 20 meter.
- Target mati atau hilang, *health point* kurang dari sama dengan 20%, ada 5 atau lebih teman tipe *minion* dalam jarak 20 meter dan ada musuh dalam jarak 20 meter.
- Target mati atau hilang, *health point* kurang dari 20%, tidak ada teman dalam jarak 20 meter, ada musuh tipe *minion* atau *ranger*.

A4. Target mati atau hilang, *health point* kurang dari 20%, tidak ada 5 atau lebih teman dalam jarak 20 meter, tidak ada musuh tipe *minion* atau *ranger*.

A5. Salah satu kondisi berikut terpenuhi:

- *Health point* lebih dari 20%.
- *Health point* kurang dari 20%, musuh tipe *minion*.
- *Health point* kurang dari 20%, musuh tipe *ranger* atau *warrior*, dalam 15 meter ada 5 atau lebih teman tipe *warrior* atau *ranger*.
- *Health point* kurang dari 20%, musuh tipe *ranger* atau *warrior*, dalam 15 meter ada 5 atau lebih teman tipe *minion*.
- *Health point* pemain kurang dari 20%, musuh tipe *ranger* atau *warrior*, *health point* musuh kurang dari *health point* pemain.
- *Health point* pemain kurang dari 20%, musuh tipe *artillery*, *health point* musuh kurang dari *health point* pemain.

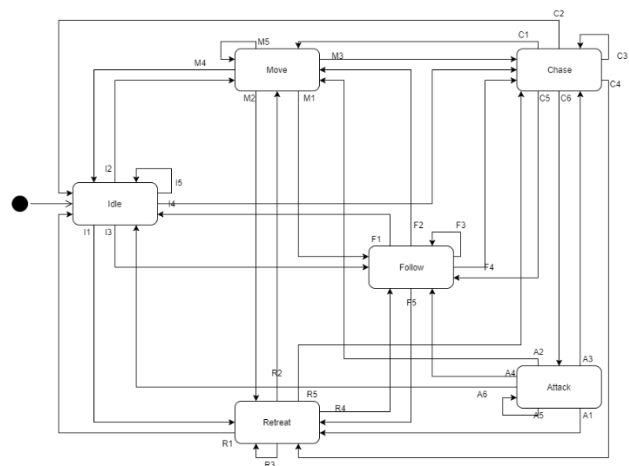
Semua angka presentase *health point* dan jarak yang berbeda-beda didapat dari hasil percobaan. Dari beberapa kemungkinan angka yang telah dicoba telah disimpulkan bahwa angka – angka yang didapat adalah angka yang terbaik untuk menghasilkan AI pasukan *warrior* yang kuat.

5. Penerapan AI Pasukan Ranger

FSM pasukan *ranger* dapat dilihat pada Gambar 5. Penjelasan untuk kondisi perpindahan *state* pada FSM pada pasukan *ranger* dapat dilihat sebagai berikut:

- I1, M2, F5, R3, C4. *Health point* kurang dari sama dengan 20% dan ada musuh dalam jarak kurang dari 5 meter.
- I2, R2, M5, F2, C1. *Health point* lebih dari sama dengan 50%, tidak ada musuh dalam jarak 30 meter.
- I3, M1, F3, R4, C5. *Health point* lebih dari 20% dan kurang dari 50%, dan ada teman tipe *warrior* atau *artillery*.
- I4, M3, F4, R5, C3. Salah satu kondisi berikut terpenuhi:
 - *Health point* lebih dari sama dengan 50%, ada musuh dalam jarak 30 meter.

- *Health point* kurang dari 20% dan ada musuh dalam jarak lebih dari 5 meter dan kurang dari 30 meter.



Gambar 5. FSM pada Pasukan Ranger

I5, M4, R1, F1, C2. Salah satu kondisi berikut terpenuhi:

- *Health point* lebih dari 20% dan kurang dari 50%, dan tidak ada teman tipe *warrior* atau *artillery*.
- *Health point* kurang dari 20%.

C6. Jarak target kurang dari 30 meter.

A1. Salah satu kondisi berikut terpenuhi:

- Jarak target kurang dari 5 meter.
- Target mati atau hilang, *health point* kurang dari sama dengan 20% dan ada musuh dalam jarak kurang dari 5 meter.

A2. Target mati atau hilang, *health point* lebih dari sama dengan 50%, tidak ada musuh dalam jarak 30 meter.

A3. Salah satu kondisi berikut terpenuhi:

- Target mati atau hilang, *health point* lebih dari sama dengan 50%, ada musuh dalam jarak 30 meter.
- Target mati atau hilang, *health point* kurang dari 20% dan ada musuh dalam jarak lebih dari 5 meter dan kurang dari 30 meter.

A4. Target mati atau hilang, *health point* lebih dari 20% dan kurang dari 50%, dan ada teman tipe *warrior* atau *artillery*.

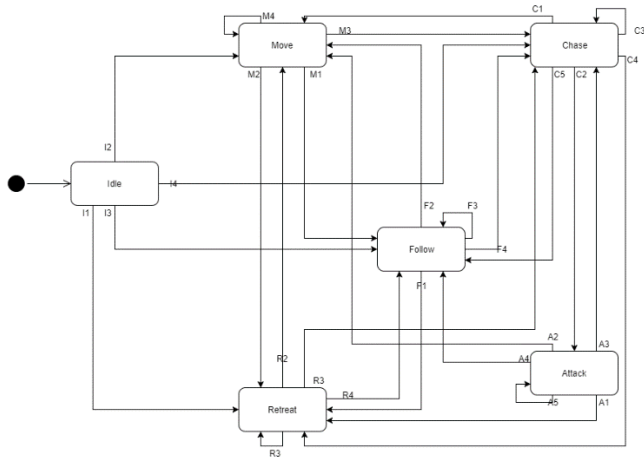
A5. Jarak target lebih dari 5 meter dan kurang dari 30 meter

A6. Salah satu kondisi berikut terpenuhi:

- *Health point* lebih dari 20% dan kurang dari 50%, dan tidak ada teman tipe *warrior* atau *artillery*.
- Target mati atau hilang, *health point* kurang dari 20%.

Semua angka presentase *health point* dan jarak yang berbeda-beda didapat dari hasil percobaan. Dari beberapa kemungkinan angka yang telah dicoba telah disimpulkan bahwa angka – angka yang didapat adalah angka yang terbaik untuk menghasilkan AI pasukan *ranger* yang kuat.

6. Penerapan AI Pasukan Artillery



Gambar 6. FSM pada Pasukan Artillery

FSM pasukan *artillery* dapat dilihat pada Gambar 6. Penjelasan untuk kondisi perpidahan *state* pada FSM pada pasukan tipe *artillery* dapat dilihat sebagai berikut:

- I1, M2, F1, R3, C4. *Health point* kurang dari 10%, tidak ada 5 atau lebih teman dalam jarak 10 meter.
- I2, R2, M4, C1, F2. Salah satu kondisi berikut terpenuhi:
 - *Health point* lebih dari 20%, tidak ada musuh dalam jarak 10 meter.
 - *Health point* lebih dari 10% dan kurang 20%, tidak ada musuh dalam jarak 10 meter.
- I3, F3, M1, R4, C5. *Health point* kurang dari 10%, dalam jarak 10 meter ada 5 atau lebih teman.
- I4, M3, F4, C3, R3. Salah satu kondisi berikut terpenuhi:
 - *Health point* lebih dari 20%, ada musuh dalam jarak 20%.
 - *Health point* lebih dari 10% dan kurang 20%, ada musuh dalam jarak 10 meter.
- C2. Jarak musuh kurang dari sama dengan 5 meter.
- A1. *Health point* kurang dari 10%, tidak ada 5 atau lebih teman dalam jarak 10 meter.
- A2. Salah satu kondisi berikut terpenuhi:
 - Target mati atau hilang, *health point* lebih dari 20%, tidak ada musuh dalam jarak 10 meter.
 - Target mati atau hilang, *health point* lebih dari 10% dan kurang 20%, tidak ada musuh dalam jarak 10 meter.
- A3. Salah satu kondisi berikut terpenuhi:
 - *Health point* kurang dari 20% dan lebih dari 20%, musuh tipe *artillery* dengan *health point* lebih dari pemain, ganti target ke tipe *minion* atau *ranger* atau *warrior* terlemah.
 - *Health point* kurang dari 10%, ada 5 atau lebih teman dalam jarak 10 meter.
 - Target mati atau hilang, *health point* lebih dari 20%, ada musuh dalam jarak 20 meter.
 - Target mati atau hilang, *health point* lebih dari 10% dan kurang 20%, ada musuh dalam jarak 10 meter.
- A4. Target mati atau hilang, *health point* kurang dari 10%, dalam jarak 10 meter ada 5 atau lebih teman.
- A5. Salah satu kondisi berikut terpenuhi:
 - *Health point* lebih dari 10%.

- *Health point* kurang dari 20% dan lebih dari 10%, musuh tipe *warrior*, *ranger*, atau *minion*.
- *Health point* kurang dari 20% dan lebih dari 20%, musuh tipe *artillery* dengan *health point* kurang dari pemain.

Semua angka presentase *health point* dan jarak yang berbeda-beda didapat dari hasil percobaan. Dari beberapa kemungkinan angka yang telah dicoba telah disimpulkan bahwa angka – angka yang didapat adalah angka yang terbaik untuk menghasilkan AI pasukan *artillery* yang kuat.

IV. UJI COBA

Pengujian dilakukan dengan mempertandingkan AI melawan AI pada semua kemungkinan *map* dan tingkat kesulitan dengan mencatat hasil kemenangan dan waktu yang dibutuhkan untuk permainan selesai. Berdasarkan pengujian yang telah dilakukan, maka *winning rate* AI dapat dirangkum pada Tabel 1.

Tabel 1.
Winning Rate AI

	<i>Easy</i>	<i>Medium</i>	<i>Hard</i>
<i>Easy</i>	40%	13.333%	0%
<i>Medium</i>	-	53.333%	0%
<i>Hard</i>	-	-	46.667%

V. KESIMPULAN

1. Penggunaan *Finite State Machine* dalam *game* berjenis *tower defense* yang dikombinasikan dengan strategi untuk menang dan aturan *game* dapat membentuk AI yang kuat.
2. AI yang memiliki tingkat kemungkinan tindakan *random* yang rendah akan lebih kuat daripada AI yang memiliki kemungkinan tindakan *random* yang lebih tinggi.
3. AI yang terbentuk telah sesuai dengan tingkat kesulitannya, dimana AI yang lebih susah akan memiliki kemungkinan kemenangan yang lebih besar.

DAFTAR PUSTAKA

- [1] "Finite-State Machines: Theory and Implementation," *Game Development Envato Tuts+*. [Online]. Available: <https://gamedev.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867>. [Accessed: 12-Feb-2017].
- [2] "What is Artificial Intelligence (AI)? - Definition from Techopedia," *Techopedia.com*. [Online]. Available: <https://www.techopedia.com/definition/190/artificial-intelligence-ai>. [Accessed: 12-Feb-2017].
- [3] "Game Genres Defined: Tower Defense," *Vita Player - the one-stop resource for PS Vita owners*. [Online]. Available: <http://www.vitaplayer.co.uk/game-genres-defined-tower-defense/>. [Accessed: 12-Feb-2017].