

Penerapan Metode Goal Oriented Action Planning untuk Agent AI pada Turn Based Tactics Video Game

Ryan Chandra Kusuma, Liliana, Hans Juwiantho

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

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

E-mail: rookie3ryan@gmail.com, lilian@peter.petra.ac.id, hans.juwiantho@peter.petra.ac.id

ABSTRAK

Kesulitan yang terdapat pada *game genre turn-based tactics* seringkali terletak di *resources* yang dimiliki musuh. *Player* diwajibkan untuk melakukan aktivitas yang repetitif untuk mengimbangi *resources* musuh. Agar *player* bisa menghabiskan waktu untuk menyusun strategi daripada untuk mengimbangi *resource* musuh, maka diimplementasikan *Goal-Oriented Action Planning* untuk AI. Dengan menggunakan metode GOAP, diharapkan AI GOAP meski tanpa *extra resources* dapat menggantikan AI FSM dengan *extra resources*.

Goal-Oriented Action Planning adalah metode *decision-making* yang membuat suatu karakter tidak hanya melakukan apa yang akan dia lakukan, tetapi juga menentukan bagaimana cara ia melakukannya. A* adalah metode yang mencari *path* dengan *exploring minimum number of nodes* dengan *minimum cost solution*. Penelitian ini menggunakan metode gabungan *Goal-Oriented Action Planning* dan A* Search.

GOAP di penelitian ini memiliki beberapa variasi gerakan berdasarkan *health point* yang dimiliki AI. Hasil penelitian menunjukkan bahwa AI GOAP tanpa *extra resources* memiliki *winrate* 33.33% terhadap AI FSM dengan *extra resources*, dan 83.33% terhadap AI FSM dengan *extra resources* tetapi *power unit* dikurangi. Hasil responden dari berbagai *player* dengan pengalaman yang berbeda menunjukkan bahwa tingkat kesulitan AI FSM dengan *extra resources* lebih tinggi, tingkat kepuasan pemain dan kerealistisan AI lebih tinggi ketika melawan AI GOAP tanpa *extra resources*.

Kata Kunci: *Goal-Oriented Action Planning*, AI, *resources*, *Finite State Machine*, A* Search

ABSTRACT

In *turn-based tactics game*, *difficulties* often placed on *resources* owned by *enemies*. *Players* have to do *repetitive action* to counterbalance *enemy's resources*. To make *players* spent more time on *strategies* rather than counterbalance *enemy's resources*, *goal-oriented action planning* will be implemented for AI. It's expected that AI GOAP even without *extra resources* can replace AI FSM with *extra resources*.

Goal-Oriented Action Planning (GOAP) is a *decision-making method* that capable of making a character not only do what it will do, but also determine how to do it. A* is a method that looks for a *path* by exploring the *minimum number of nodes* with *minimum cost solution*. This research combines GOAP and A* search.

GOAP in this research has several variations of actions based on *health points*. Result of the research shows that AI GOAP without *extra resources* has 33.33% *winrate* against AI FSM with *extra resources*, and 86.66% against AI FSM with *extra resources* but reduced *power unit*. The results of respondents from various

players with different experiences show that the difficulty of AI FSM with *extra resources* is higher, the level of *player satisfaction* and AI's realistic level is higher when fought against AI GOAP without *extra resources*.

Keywords: *Goal-Oriented Action Planning*, AI, *resources*, A* Search, *Finite State Machine*

1. PENDAHULUAN

Game merupakan suatu aktivitas paling menyenangkan dan *enjoyable* apabila *game* tersebut memberikan tantangan yang cukup untuk pemainnya. Contoh tantangan tersebut misalnya, mempelajari cara bermainnya, memecahkan masalah, atau menemukan hal-hal baru ketika bermain *game* tersebut [8]. *Game* merupakan permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang dibuat semenarik mungkin agar pemain bisa mendapatkan sesuatu hingga adanya kepuasan batin [5].

Video games merupakan hiburan dan tantangan. *Video games* tidak akan menyenangkan jika tidak ada tantangannya. Apabila tantangan tersebut terlalu mudah, maka *game* tersebut akan menjadi membosankan. Tetapi jika sebaliknya, *game* tersebut dapat membuat pemain frustrasi. Hal ini berhubungan dengan *flow-state*, yaitu ketika kemampuan pemain dan tantangan dari *game* setara [18]. Di dalam seri permainan *turn-based tactics game*, biasanya musuh AI memiliki fitur tambahan seperti *unit* yang lebih banyak dan *power* yang lebih kuat. Hal ini membuat *player* yang ingin mendapatkan pengalaman bermain yang optimal tanpa mengeluarkan waktu lebih banyak menjadi enggan untuk bermain permainan tersebut meskipun menyukai *genre* tersebut, karena hal-hal tersebut mengharuskan *player* untuk melakukan aktivitas bernama *grinding*. *Grinding* adalah sebuah aktivitas dimana seorang *player* melakukan *action* yang sama berulang kali untuk mendapatkan *resources* [7].

Goal-Oriented Action Planning adalah suatu metode *decision-making* yang dapat membuat suatu karakter tidak hanya melakukan apa yang akan dia lakukan, tetapi juga menentukan bagaimana cara ia melakukannya. Dalam konteks *reward processing*, proses dengan *goal-directed* akan membandingkan potensial hasil dari setiap aksi yang akan dilakukan untuk mendapatkan hasil yang paling diinginkan [6]. Dengan struktur GOAP, GOAP mampu memfasilitasi suatu karakter dengan cara mempertahankan dan menggunakan ulang *behavior* tersebut disesuaikan dengan situasi dimana karakter tersebut berada. Sistem GOAP tidak akan mengganti kebutuhan akan *Finite-state machine*, tetapi lebih menyederhanakan FSM yang dibutuhkan [12]. Di dalam sisi pengimplementasian, dengan menggunakan GOAP, apabila ada *action* yang ingin ditambahkan, pembuat hanya tinggal menambahkan *action* ke dalam program, tanpa perlu mengganti *action* lainnya. Dengan menggunakan metode *Goal-Oriented Action Planning*, *agent* AI akan memiliki aksi

yang cukup bervariasi dan adaptif pada *state* yang dia alami, sehingga *player* dapat bermain dengan *agent* AI yang memiliki *resources* yang sama dengan kesulitan yang cukup menantang untuk *player* sehingga *player* dapat menikmati permainan secara optimal tanpa harus melakukan *grinding*.

Jeff Orkin menganalisa sebuah game bernama *No One Lives Forever 2: A spy in H.A.R.M.'s Way (NOLF2)*, karena game tersebut adalah salah satu contoh yang memiliki *goal-directed autonomous characters*, tanpa kemampuan untuk *planning*. Karakter di dalam *NOLF2* secara konstan melakukan evaluasi ulang tujuan mereka, dan memilih tujuan yang paling relevan untuk mengontrol perilaku mereka. Di dalam penelitiannya, Jeff Orkin ingin menerapkan metode GOAP ke dalam game tersebut dengan cara menggunakan *planner* untuk membuat sebuah *list action* yang akan digunakan untuk memenuhi sebuah *goal*. Hasil penelitian tersebut menyebutkan apa saja keuntungan yang bisa didapatkan jika menggunakan metode GOAP, baik dalam bidang *development*, *runtime behaviour*, dan apa yang bisa dilakukan oleh karakter jika menggunakan metode GOAP, sehingga bisa membuat variasi gerakan, seperti dari *killenemy* menjadi *killenemyarmed*, tanpa mengubah banyak *code*[12].

Magnusson pernah melakukan penelitian serupa pada sebuah *Real-time Strategy Game*. Magnusson membuat sebuah AI bernama AI Ice, yang bertujuan untuk mengalahkan musuh di dalam game tersebut. Untuk melakukan hal tersebut, AI Ice akan mencari informasi di dalam *gameplay* untuk menentukan, membuat, dan menjalankan sebuah tugas sesuai prioritas yang paling tinggi. Hasil dari penelitian tersebut mengatakan bahwa AI Adaptive dengan metode GOAP yang memiliki fitur yang sama dengan pemain manusia (*disabled fog of war, no extra resource, etc*) mampu mengalahkan AI statis yang memiliki fitur yang lebih dari fitur pemain manusia.

Tujuan yang diangkat dari penelitian ini adalah untuk kontribusi pengetahuan lebih lanjut tentang metode *Goal Oriented Action Planning* pada dataset yang berbeda, khususnya *Turn-based Tactics Video Game*, dengan tujuan untuk membuat *Agent* AI yang adaptif sesuai dengan *state* yang dialami AI pada *genre game* tersebut. Dengan aksi AI yang variatif dan adaptif, AI tidak membutuhkan *resources* tambahan, sehingga *player* tidak memerlukan *grinding* untuk mendapatkan *progress* di dalam game tersebut, tetapi masih mendapatkan kesulitan yang cukup menantang ketika melawan AI tersebut. Agar permainan tidak berjalan dengan monoton, maka permainan diberikan sisi *gambling* berupa *hidden personality* di tiap karakter agar gerakan dari karakter bisa berbeda dari satu pertandingan ke pertandingan lainnya. Hasil dari penelitian ini akan berupa uji coba dari pemain melawan *agent* AI GOAP dan uji coba AI GOAP melawan AI Finite-State Machine, untuk mengetahui apakah metode GOAP lebih baik daripada Finite-State Machine pada game *Turn-based tactics*.

2. TINJAUAN STUDI

Jeff Orkin menganalisa sebuah game bernama *No One Lives Forever 2: A spy in H.A.R.M.'s Way (NOLF2)*, karena game tersebut adalah salah satu contoh yang memiliki *goal-directed autonomous characters*, tanpa kemampuan untuk *planning*. Karakter di dalam *NOLF2* secara konstan melakukan evaluasi ulang tujuan mereka, dan memilih tujuan yang paling relevan untuk mengontrol perilaku mereka. Di dalam penelitiannya, Jeff Orkin mengatakan bahwa sebuah karakter akan membuat sebuah rencana secara *real-time* dengan cara memasukkan tujuan di dalamnya kepada sebuah sistem bernama *planner*. *Planner*

tersebut akan mencari *action* yang dibutuhkan untuk sebuah *sequence* yang akan membawa sebuah karakter dari *starting state* hingga *goal state*. Jika *planner* tersebut sukses, *planner* akan menghasilkan sebuah rencana untuk diikuti oleh karakter tersebut untuk pengarahan perilakunya. Jika tidak, maka karakter akan meninggalkan rencana sekarang dan akan membuat yang baru[12].

Magnusson pernah melakukan penelitian serupa pada sebuah *Real-time Strategy Game*. Magnusson membuat sebuah AI bernama AI Ice, yang bertujuan untuk mengalahkan musuh di dalam game tersebut. Untuk melakukan hal tersebut, AI Ice akan mencari informasi di dalam *gameplay* untuk menentukan, membuat, dan menjalankan sebuah tugas sesuai prioritas yang paling tinggi. Hasil dari penelitian tersebut mengatakan bahwa AI Adaptive dengan metode GOAP yang memiliki fitur yang sama dengan pemain manusia (*disabled fog of war, no extra resource, etc*) mampu mengalahkan AI statis yang memiliki fitur yang lebih dari fitur pemain manusia[2].

Studiawan pernah membuat penelitian mengenai GOAP pada *space tactical game*. Metode yang digunakan adalah metode GOAP dan *blackboard system*. Tujuan dari penelitian tersebut adalah untuk melihat apakah metode GOAP mampu diimplementasikan di dalam *space tactical game*. Hasil penelitian menunjukkan bahwa GOAP membantu mengurangi kompleksitas ketika mendesain AI[19].

3. METODOLOGI PENELITIAN

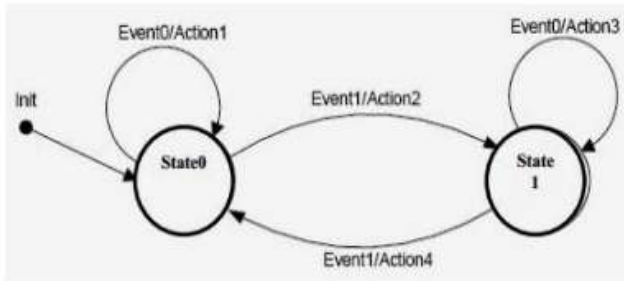
3.1 Goal Oriented Action Planning

Goal-Oriented Action Planning adalah suatu metode *decision-making* yang dapat membuat suatu karakter tidak hanya melakukan apa yang akan dia lakukan, tetapi juga menentukan bagaimana cara ia melakukannya. Dengan struktur GOAP, GOAP mampu memfasilitasi suatu karakter dengan cara mempertahankan dan menggunakan ulang *behavior* tersebut disesuaikan dengan situasi dimana karakter tersebut berada. Sistem GOAP tidak akan mengganti kebutuhan akan Finite-state machine, tetapi lebih menyederhanakan FSM yang dibutuhkan [12]. GOAP merupakan teknik untuk *decision making* yang akan menghasilkan rantai aksi yang bernama *plan* untuk mencapai suatu *goal state* yang sudah ditentukan sebelumnya [19]. GOAP biasanya memiliki dua komponen utama dalam melakukan tugasnya, yaitu *goals* dan *actions*. Agar *goals* tercapai, maka harus ada *action* yang dijalankan dan *world condition* yang harus dicapai. Untuk menjalankan *action*, ada *precondition* yang harus dipenuhi. Setelah *action* dijalankan, akan ada *effect* yang dapat mengubah *world condition* untuk mencapai *goals* yang diinginkan. *Precondition* adalah kondisi dimana sebuah *state* harus terpenuhi untuk menjalankan sebuah *action*. Input dari GOAP adalah sebuah *goals* sesuai dengan *world condition* sekarang, dan *action* sesuai dengan *goals* yang ingin dicapai.

3.2 Finite State Machine

Finite state machine mengenalkan sebuah konsep tentang *state* yang sudah mesin pernah lakukan sebelumnya. Semua *states* mewakili semua kemungkinan situasi dimana sebuah mesin bisa berada. Ketika aplikasi berjalan, *state* dapat berubah dari waktu ke waktu, dan *output* yang dihasilkan juga bergantung pada *state* sekarang, dan juga *input* yang dibutuhkan [21]. Finite State Machine sudah digunakan untuk menjelaskan secara matematis berbagai *states* dan *events* yang dapat mengarah ke *transition*[16]. Penerapan *finite state machine* pada game berguna untuk menentukan berbagai macam respon NPC terhadap pemain berdasarkan interaksi yang dilakukan[4]. *Finite State Machine*

membagi sebuah respon objek game ke dalam *states* sehingga objek tersebut memiliki bagian untuk setiap respon objek *game*[14]. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu[17]. Di dalam *state* tersebut, biasanya mesin melakukan sebuah *action* yang sesuai dengan *state* tersebut. Contohnya adalah sebuah *warrior* di dalam sebuah *game*. *Warrior* tersebut memiliki 2 keadaan awal (*state*). Keadaan pertama yaitu mencari musuh. Ketika *warrior* berada di *state* mencari musuh, maka ia akan menjalankan *action* bernama *move to X*, dimana *warrior* akan berjalan ke tempat-tempat tertentu untuk menemukan musuh. Apabila *warrior* berhasil menemukan musuh, maka *warrior* akan bertransisi *state* menjadi menyerang musuh. Apabila *warrior* berada di *state* menyerang musuh, maka *warrior* akan menjalankan *action* memukul musuh tersebut.



Gambar 1. Struktur *Finite State Machine*

Pada Gambar 1 terdapat 2 buah *state* dan 2 buah *input*. Ketika sistem mulai dihidupkan, sistem bertransisi ke *state0*, dan menghasilkan *action1* jika terjadi masukan *event0*, sedangkan jika terjadi *event1*, maka *action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *state1* dan seterusnya[14].

3.3 Turn-based Strategy Games (TBS)

Di dalam *turn-based strategy*, *player* hanya dapat menjalankan aksi di putarannya. Sebagai contoh, *player* hanya dapat melakukan 2 aksi di tiap putarannya. *Turn-based strategy* memberikan kebebasan kepada *player* untuk berpikir tanpa waktu yang terbatas [20]. *Turn-based Strategy Games* bersifat putaran, dimana tiap pemain memiliki waktunya sendiri untuk menggerakkan *unit-unit* yang dimiliki oleh pemain tersebut. Tiap pemain memiliki objektif, yaitu mengalahkan musuh dengan *unit-unit* yang mereka miliki.

3.4 A* search

Algoritma A* tradisional dapat menghasilkan *path* yang optimal dengan cara meminimalkan biaya *path*[10]. Metode yang digunakan menunjukkan bahwa cara ini akan mencari suatu *path* dengan *exploring minimum number of nodes* dengan *minimum cost solution*. Karena *simplicity* yang dimilikinya, A* hampir selalu menjadi pilihan *search method*. Ini karena A* menjamin untuk mencari rute terpendek di graf [11]. A* menggunakan *Best First Search* (BFS) dan menemukan jalur dengan biaya terkecil (*least-cost path*) dari *node* awal ke *node* tujuan [1]. A* digunakan untuk mencari *path* terpendek [9]. Algoritma ini dapat membatasi ruang pencarian untuk mendapatkan solusi yang baik dalam waktu yang wajar [13]. Algoritma A* menerapkan teknik heuristic dalam membantu penyelesaian persoalan. Heuristik adalah penilai yang memberi harga pada tiap simpul yang memandu A* mendapatkan solusi yang diinginkan[3]. Apabila terdapat dua *path* atau lebih yang memiliki *goals* yang sama, tetapi dengan *cost* yang berbeda, maka *path* dengan *cost* yang paling kecil yang akan diambil. A* mengevaluasi *nodes* dengan mengombinasikan $g(n)$,

biaya untuk mencapai *node*, dan $h(n)$, biaya yang dibutuhkan untuk mencapai ke *goal* dari *nodes* [15].

$$f(n) = g(n) + h(n) \quad (1)$$

Dimana :

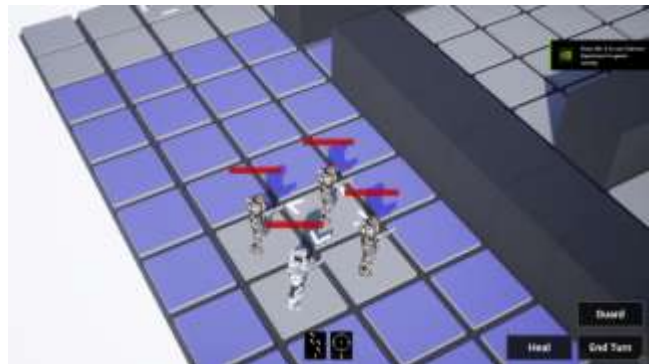
- $g(n)$ adalah biaya *path* dari *start node* ke *node n*
- $h(n)$ adalah perkiraan biaya *path* terendah dari *n* ke *goal*
- $f(n)$ adalah biaya solusi termurah melalui *n* ke *goal*

4. PENGUJIAN DAN DISKUSI

Pengujian pertama dilakukan untuk mendapatkan hasil kepuasan *human player* yang sudah selesai mencoba aplikasi ini dan mengambil survei setelahnya sebanyak 20 orang. Tiap orang memiliki *resources* yang berbeda untuk melawan AI *Goal-Oriented Action Planning*. Survei yang diisi berisikan skala 1-5 dalam 3 hal, yaitu tingkat kepuasan pemain ketika melawan AI yang dilawan, tingkat kesulitan AI yang dilawan, dan tingkat kerealistisan gerakan yang dilakukan oleh AI yang dilawan. Dari hasil survei ini, didapatkan kepuasan pemain ketika melawan AI GOAP tanpa *extra resources* dan AI FSM yang memiliki *resources* yang lebih.

Pengujian kedua dilakukan untuk mendapatkan performa AI *Goal-Oriented Action Planning* tanpa *extra resources* berdasarkan *winrate* ketika melawan AI *Finite State Machine* yang memiliki *resources* yang lebih dari AI *Goal-Oriented Action Planning*. *Resources* yang dimaksud adalah jumlah *unit*, *power unit*, dan *personality* yang dimiliki oleh AI *Finite State Machine*. Pengujian dilakukan sebanyak 12 kali. Pengujian dibagi menjadi 6 kelompok, dimana tiap kelompok terdapat 2 kali percobaan. Perbedaan pengujian yang terdapat di tiap kelompok terdapat pada siapa yang mendapatkan putaran pertama. Perbedaan satu kelompok dengan kelompok yang lain adalah *unit* yang digunakan tiap AI.

4.1 Hasil Gameplay



Gambar 2. *Interface Game*

Pada gambar 2, dapat dilihat *interface game* yang dapat dilihat oleh *player*. Di bawah, terdapat *skill window* yang menyediakan informasi tentang *skill* dari karakter yang dikontrol. Kemudian, di kanan terdapat aksi *heal*, *guard*, dan *end turn* untuk mengakhiri putaran dari karakter tersebut, dimana *heal* berfungsi untuk menyembuhkan karakter tersebut sebanyak 10 poin, *guard* untuk mengurangi *damage* yang masuk sebesar 50%, dan *end turn* untuk mengakhiri putaran tanpa melakukan apa-apa.

4.2 Pengujian Kelompok A

Pada kelompok ini, *unit* yang digunakan oleh AI *Finite State Machine* ada 5, yaitu, 2 *Warrior*, 2 *Archer*, dan 1 *Healer*, serta

personality aggressive. Unit yang digunakan oleh AI *Goal-Oriented Action Planning* ada 4, yaitu 1 *Warrior*, 2 *Archer*, dan 1 *Healer*. Pada pengujian pertama, putaran pertama diberikan kepada AI GOAP. Pada pengujian kedua, putaran pertama diberikan kepada AI FSM.

Pada 2 kali pengujian, didapatkan *winrate* sebesar 50%. Pada pengujian pertama, ketika AI GOAP mendapatkan putaran pertama, kemenangan diraih oleh AI GOAP. Di pengujian ini, AI FSM mampu mengeliminasi unit yang dimiliki oleh AI GOAP sebesar 25%. Pada pengujian kedua, ketika AI FSM memiliki putaran pertama, kemenangan berhasil diraih oleh AI FSM. Di pengujian ini, AI GOAP mampu mengeliminasi unit FSM sebanyak 40%.

4.3 Pengujian Kelompok B

Pada kelompok ini, unit yang digunakan oleh AI *Finite State Machine* ada 5, yaitu, 2 *Warrior*, 2 *Archer*, dan 1 *Healer*, *personality aggressive*, dengan catatan *power* yang dimiliki oleh beberapa unit dikurangi. *Power* yang direndahkan berupa HP *Warrior* dari 100 menjadi 80, HP *Archer* dari 60 menjadi 80, dan *attack Archer* dari 30 menjadi 20. Unit yang digunakan oleh AI *Goal-Oriented Action Planning* ada 4, yaitu 1 *Warrior*, 2 *Archer*, dan 1 *Healer*, tanpa ada pengurangan *power unit*. Pada pengujian ketiga, putaran pertama diberikan kepada AI GOAP. Pada pengujian keempat, putaran pertama diberikan kepada AI FSM.

Pada 2 kali pengujian, didapatkan *winrate* sebesar 100%. Pada pengujian ketiga, ketika AI GOAP mendapatkan putaran pertama, kemenangan diraih oleh AI GOAP. Di pengujian ini, AI FSM mampu mengeliminasi unit yang dimiliki oleh AI GOAP sebesar 25%. Pada pengujian keempat, meski AI GOAP tidak mendapatkan putaran pertama, AI GOAP mampu meraih kemenangan terhadap AI FSM. Di pengujian ini, AI FSM memiliki hasil yang sama seperti pengujian sebelumnya, dimana AI FSM mampu mengeliminasi unit yang dimiliki oleh AI GOAP sebesar 25%.

4.4 Pengujian Kelompok C

Pada kelompok ini, unit yang digunakan oleh AI *Finite State Machine* ada 5, yaitu, 2 *Warrior*, 2 *Archer*, dan 1 *Mage*, serta *personality neutral*. Unit yang digunakan oleh AI *Goal-Oriented Action Planning* ada 4, yaitu 2 *Warrior*, 1 *Archer*, dan 1 *Mage*. Pada pengujian kelima, putaran pertama diberikan kepada AI GOAP. Pada pengujian keenam, putaran pertama diberikan kepada AI FSM.

Pada 2 kali pengujian, didapatkan *winrate* sebesar 0%. Pada pengujian kelima, meski AI GOAP mendapatkan putaran pertama, kemenangan diraih oleh AI FSM. Di pengujian ini, AI GOAP mampu mengeliminasi unit AI FSM sebesar 20%. Pada pengujian keenam, kemenangan berhasil diraih oleh AI FSM. Tetapi di pengujian ini, AI GOAP mampu mengeliminasi unit yang dimiliki oleh AI FSM sebanyak 80%.

4.5 Pengujian Kelompok D

Pada kelompok ini, unit yang digunakan oleh AI *Finite State Machine* ada 5, yaitu, 2 *Warrior*, 2 *Archer*, dan 1 *Mage*, *personality neutral*, dengan catatan *power* yang dimiliki dari beberapa unit dikurangi. *Power* yang dikurangi berupa HP *Warrior* dari 100 menjadi 80, HP *Archer* dari 60 menjadi 80, dan *attack Archer* dari 30 menjadi 20. Unit yang digunakan oleh AI *Goal-Oriented Action Planning* ada 4, yaitu 2 *Warrior*, 1 *Archer*, dan 1 *Mage*, tanpa ada pengurangan *power unit*. Pada pengujian

ketujuh, putaran pertama diberikan kepada AI GOAP. Pada pengujian kedelapan, putaran pertama diberikan kepada AI FSM.

Pada 2 kali pengujian, didapatkan *winrate* sebesar 50%. Pada pengujian ketujuh, ketika AI GOAP mendapatkan putaran pertama, kemenangan diraih oleh AI GOAP. Di pengujian ini, AI FSM mampu mengeliminasi unit yang dimiliki oleh AI GOAP sebesar 25%. Pada pengujian kedelapan, kemenangan diraih oleh AI FSM. Di pengujian ini, AI GOAP mampu mengeliminasi unit yang dimiliki oleh AI FSM sebanyak 40%.

4.6 Pengujian Kelompok E

Pada kelompok ini, unit yang digunakan oleh AI *Finite State Machine* ada 5, yaitu, 2 *Warrior*, 2 *Archer*, dan 1 *Mage*, serta *personality non-aggressive*. Unit yang digunakan oleh AI *Goal-Oriented Action Planning* ada 4, yaitu 1 *Warrior*, 2 *Archer*, dan 1 *Mage*. Pada pengujian kesembilan, putaran pertama diberikan kepada AI GOAP. Pada pengujian kesepuluh, putaran pertama diberikan kepada AI FSM.

Pada 2 kali pengujian, didapatkan *winrate* sebesar 50%. Pada pengujian sembilan, meski AI GOAP memiliki putaran pertama, kemenangan diraih oleh AI FSM. Di pengujian ini, AI GOAP mampu mengeliminasi unit AI FSM sebesar 20%. Pada pengujian kesepuluh, meski AI FSM memiliki putaran pertama, kemenangan diraih oleh AI GOAP. Di pengujian ini, AI FSM tidak mampu mengeliminasi unit yang dimiliki oleh AI GOAP, sehingga AI GOAP memiliki kemenangan telak.

4.7 Pengujian Kelompok F

Pada kelompok ini, unit yang digunakan oleh AI *Finite State Machine* ada 5, yaitu, 2 *Warrior*, 2 *Archer*, dan 1 *Mage*, *personality non-aggressive*, dengan catatan *power* yang dimiliki oleh beberapa unit dikurangi. *Power* yang dikurangi berupa HP *Warrior* dari 100 menjadi 80, HP *Archer* dari 60 menjadi 80, dan *attack Archer* dari 30 menjadi 20. Unit yang digunakan oleh AI *Goal-Oriented Action Planning* ada 4, yaitu 1 *Warrior*, 2 *Archer*, dan 1 *Mage*, tanpa ada pengurangan *power unit*. Pada pengujian kesebelas, putaran pertama diberikan kepada AI GOAP. Pada pengujian keduabelas, putaran pertama diberikan kepada AI FSM.

Pada 2 kali pengujian, didapatkan *winrate* sebesar 100%. Pada pengujian kesebelas, kemenangan diraih oleh GOAP. Di pengujian ini, AI FSM mampu mengeliminasi unit AI GOAP sebesar 25%. Pada pengujian keduabelas, meski AI FSM memiliki putaran pertama, kemenangan diraih oleh AI GOAP. Di pengujian ini, AI FSM tidak mampu mengeliminasi unit yang dimiliki oleh AI GOAP, sehingga AI GOAP memiliki kemenangan telak.

4.8 Pengujian Human Player terhadap AI GOAP tanpa *extra resources* dan AI FSM dengan *extra resources*

Pengujian Human Player vs AI GOAP dilakukan dimana *human player* dapat memilih unit yang diinginkan maksimal 4 unit, melawan AI GOAP yang memiliki 4 unit juga. Pengujian dilakukan kepada 20 *human player* yang berbeda. Survei dilakukan setelah *human player* selesai melakukan uji coba aplikasi melawan AI GOAP. Pertanyaan yang diajukan berupa pertanyaan berbentuk skala dari 1 sampai 5 dengan tiga buah pertanyaan, yaitu tingkat kepuasan *human player* melawan AI GOAP, seberapa realistis gerakan AI GOAP, dan seberapa susah AI GOAP di dalam aplikasi tersebut. Setelah itu, *human player* juga diarahkan untuk melawan AI FSM yang memiliki 5 unit yang memiliki *personality* yang berbeda. Setelah itu, akan diadakan survei yang sama untuk AI FSM. Hasil kuesioner tentang *rating*

AI GOAP yang tidak memiliki *extra resources* berdasarkan jawaban *player* dapat dilihat pada Tabel 1, dan hasil kuesioner tentang *rating* AI FSM yang memiliki *extra resources* dapat dilihat pada Tabel 2.

Tabel 1. Tabel Hasil Jawaban *Human Player* tentang AI GOAP yang tidak memiliki *extra resources*

Pengalaman	Jumlah Responden	Tingkat Kesulitan AI	Tingkat Kepuasan saat melawan AI	Tingkat Kerealistisan AI
Sering Bermain	8	3.13	4.13	3.63
Kadang-Kadang bermain	10	3.7	3.7	3.6
Tidak Pernah Bermain	2	3.5	3.5	4

Dari hasil jawaban semua *player* tentang AI GOAP yang tidak memiliki *resources* tambahan di Tabel 1, AI GOAP memiliki tingkat kesulitan yang paling tinggi jika dihadapkan dengan *player* yang kadang-kadang bermain *game genre turn-based tactics*. Selain itu, *player* yang sering bermain *game genre turn-based tactics* memiliki rata-rata tingkat kepuasan tertinggi ketika melawan AI GOAP. Tingkat kerealistisan AI GOAP memiliki rata-rata paling tinggi ketika dihadapkan dengan *player* yang tidak pernah bermain *game genre turn-based tactics*. AI GOAP memiliki 12.5% *winrate* terhadap *player* yang sering bermain *game genre turn-based tactics*, 20% *winrate* terhadap *player* yang kadang-kadang bermain *game genre turn-based tactics*, dan 50% *winrate* terhadap *player* yang tidak pernah bermain *game genre turn-based tactics*.

Tabel 2. Tabel Hasil Jawaban *Human Player* tentang AI FSM yang memiliki *extra resources*

Pengalaman	Jumlah Responden	Tingkat Kesulitan AI	Tingkat Kepuasan saat melawan AI	Tingkat Kerealistisan AI
Sering Bermain	8	3.75	3.13	2.88
Kadang-Kadang bermain	10	4.2	3	2.8
Tidak Pernah Bermain	2	4.5	2.5	3

Dari hasil jawaban semua *player* tentang AI FSM yang memiliki *resources* tambahan di Tabel 2, AI FSM memiliki tingkat

kesulitan yang paling tinggi jika dihadapkan dengan *player* yang tidak pernah bermain *game genre turn-based tactics*. Selain itu, *player* yang sering bermain *game genre turn-based tactics* memiliki rata-rata tingkat kepuasan tertinggi ketika melawan AI FSM yang memiliki *extra resources*. Tingkat kerealistisan AI FSM memiliki rata-rata paling tinggi ketika dihadapkan dengan *player* yang tidak pernah bermain *game genre turn-based tactics*. AI FSM yang memiliki *extra resources* memiliki 37.5% *winrate* terhadap *player* yang sering bermain *game genre turn-based tactics*, 70% *winrate* terhadap *player* yang kadang-kadang bermain *game genre turn-based tactics*, dan 100% *winrate* terhadap *player* yang tidak pernah bermain *game genre turn-based tactics*.

5. KESIMPULAN

Berdasarkan hasil pengujian AI *Goal-Oriented Action Planning* melawan AI *Finite State Machine*, *Human Player* melawan AI *Goal-Oriented Action Planning* dan *Finite State Machine*, dapat disimpulkan beberapa hal diantaranya :

- AI *Goal-Oriented Action Planning* yang terdapat di penelitian ini memiliki aksi yang disesuaikan, yaitu apabila terdapat musuh di *range* yang memiliki nyawa lebih kecil, maka AI GOAP akan memprioritaskan musuh tersebut. Apabila AI *Goal-Oriented Action Planning* dan musuh AI dalam kondisi *low health*, maka akan dilakukan pengecekan berdasarkan perbedaan HP. Apabila HP AI GOAP lebih kecil dan perbedaan HP AI dengan HP musuh > 10, maka AI akan memprioritaskan untuk melakukan *run away* kemudian melakukan *guard*. Tetapi, HP AI GOAP lebih kecil dan perbedaan HP AI dengan HP musuh <= 10, maka AI *Goal-Oriented Action Planning* akan melakukan *run away* dan *heal*. Apabila HP AI GOAP lebih besar dari musuh ketika di *low health condition*, maka AI akan memprioritaskan untuk menyerang musuh tersebut.
- Dari perbandingan hasil survei yang dilakukan *human player* setelah melawan AI GOAP tanpa *extra resources* dan AI FSM dengan *extra resources*, bisa disimpulkan bahwa AI FSM dengan *extra resources* memiliki tingkat kesulitan yang lebih tinggi daripada AI GOAP tanpa *extra resources*, baik untuk *player* yang tidak pernah bermain *game genre turn-based tactics*, kadang-kadang, maupun sering. Selain itu, tingkat kepuasan pemain juga lebih tinggi ketika melawan AI GOAP tanpa *extra resources* daripada melawan AI FSM dengan *extra resources* baik untuk *player* yang sering bermain, kadang-kadang, maupun tidak pernah. Kerealistisan AI GOAP tanpa *extra resources* juga lebih tinggi dibandingkan dengan AI FSM dengan *extra resources*.
- Dari 12 kali pengujian AI GOAP tanpa *extra resources* melawan AI FSM dengan *extra resources*, didapatkan *winrate* AI GOAP sebesar 33.33% ketika AI FSM memiliki *power unit* yang seimbang dengan *unit* AI GOAP, sedangkan AI FSM memiliki *winrate* sebesar 66.66% dalam 6 kali pengujian dengan *resources* yang berbeda-beda. Selain itu, AI GOAP mampu meraih *winrate* sebesar 83.33% ketika melawan AI FSM dengan *power unit* yang dikurangi sebanyak 6 kali pengujian. Dari pengujian-pengujian tersebut, meski AI GOAP memiliki kesempatan mendapatkan putaran pertama, AI GOAP tetap bisa kalah melawan AI FSM, begitu juga sebaliknya. Selain itu, kemenangan AI GOAP juga lebih sering didapatkan ketika AI FSM memiliki jumlah *unit* yang lebih banyak, tetapi

dengan *power* yang dikurangi. Karena *winrate* yang dimiliki AI GOAP tidak terlalu besar, maka AI GOAP masih belum mampu untuk menggantikan AI FSM di dalam *Turn-based Tactics*.

Saran yang dapat diberikan untuk penelitian lebih lanjut antara lain:

- *Power* yang ada di beberapa *unit* harus dipertimbangkan lagi untuk mencegah ketidaksetaraan *power* antar *unit*.
- Menambahkan lebih banyak skenario ke dalam *game* untuk AI GOAP dengan tujuan meningkatkan *winrate* ketika menghadapi AI FSM dengan *resources* tambahan.

6. DAFTAR PUSTAKA

- [1] Dalem, I. B. 2018. Penerapan Algoritma A* (Star) Menggunakan Graph Untuk Menghitung Jarak Terpendek. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 1, 1 (April, 2018), 41-47. DOI=doi:10.31598/jurnalresistor.v1i1.253.
- [2] Hall, T., & Magnusson, M. 2010. Adaptive Goal Oriented Action Planning for RTS Games. URI=https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A831698&dsid=5330.
- [3] Hermawan, H., & Setiyani, H. 2019. Implementasi Algoritma A-Star Pada Permainan Komputer Roguelike Berbasis Unity. *Jurnal Algoritma, Logika dan Komputasi*, 2, 1. 111-120. DOI=http://dx.doi.org/10.30813/j-alu.v2i1.1571
- [4] Hidayat, E. W., Rachman, A. N., & Azim, M. F. 2019. Penerapan Finite State Machine pada Battle Game Berbasis Augmented Reality. *Jurnal Edukasi dan Penelitian Informatika*, 5,1,54. DOI=http://dx.doi.org/10.26418/jp.v5i1.29848
- [5] Hormansyah, D. S., Ririd, A. R. T. H., & Pribadi, D. T. 2018. Implementasi FSM (Finite State Machine) pada Game Perjuangan Pangeran Diponegoro. *Jurnal Informatika Polinema*, 4, 4, 290. DOI= https://doi.org/10.33795/jip.v4i4.222
- [6] Huang, Y., Yaple, Z. A., & Yu, R. 2020. Goal-oriented and habitual decisions: Neural signatures of model-based and model-free learning. *NeuroImage*, 215, 116834. DOI= https://doi.org/10.1016/j.neuroimage.2020.116834
- [7] Karlsten, F. 2011. Entrapment and Near Miss: A Comparative Analysis of Psycho-Structural Elements in Gambling Games and Massively Multiplayer Online Role-Playing Games. *International Journal of Mental Health and Addiction*, vol 9, no 2, 193-207. DOI=doi:10.1007/s11469-010-9275-4.
- [8] Korhonen, H., & Koivisto, E. M. 2006. Playability Heuristics for Mobile Games. *ACM International Conference Proceeding Series*, vol 159, 9-16. DOI=doi:10.1145/1152215.1152218.
- [9] Kusuma, M., & Machbub, C. 2019. Humanoid Robot and Rerouting using A-Star Search Algorithm. *IEEE International Conference on Signals and Systems*. 110-115. DOI= 10.1109/ICSSIGSYS.2019.8811093
- [10] Liu, C., Mao, Q., Chu, X., & Xie, S. 2019. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. *Applied Sciences*, 9, 6, 1057. DOI=https://doi.org/10.3390/app9061057
- [11] Mathew, G. E. 2015. Direction based heuristic for pathfinding in video games. *Procedia Computer Science*, vol 47, 262-271. DOI=doi:10.1016/j.procs.2015.03.206.
- [12] Orkin, J. 2003. Applying Goal-Oriented Action Planning to Games. *AI Game Programming Wisdom 2*, 217-227.
- [13] Peng, D., Wolff, A., & Haunert, J. H. 2017. Using the A* Algorithm to Find Optimal Sequences for Area Aggregation. *International Cartographic Conference*. 389-404. DOI=10.1007/978-3-319-57336-6_27
- [14] Ramadhan, H. F., Sitorus, S. H., & Rahmayuda, S. 2019. Game Edukasi Pengenalan Budaya dan Wisata Kalimantan Barat Menggunakan Metode Finite State Machine Berbasis Android. *Coding Jurnal Komputer dan Aplikasi*, 7, 1. 108-119. DOI=http://dx.doi.org/10.26418/coding.v7i01.32691
- [15] Russell, S., & Norvig, P. 2009. *Artificial Intelligence: A Modern Approach 3rd edition*. Pearson.
- [16] Saleh, M. Esa, Y & Mohammed, A. 2017. Centralized Control for DC Microgrid using Finite State Machine. *IEEE Power & Energy Society Innovative Smart Grid Technologies*, 1-5. DOI= 10.1109/ISGT.2017.8086062
- [17] Sifaulloh, H., Fadila, J. N., & Nugroho, F. 2021. Penerapan Metode Finite State Machine pada Game Santri on the Road. *Walisongo Journal of Information Technology*, 3, 1, 11-18. DOI= 10.21580/wjit.2021.3.1.7135
- [18] Sofyan, I. A., Akbar, M. A., & Afrianto, T. 2019. Implementasi Dynamic Difficulty Adjustment Pada Racing Game Menggunakan Metode Behaviour Tree. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, vol 3, no 1, 643-650.
- [19] Studiawan, R., Hariadi, M., & Sumpeno, S. 2018. Tactical Planning in Space Game using Goal-Oriented Action Planning. *JAREE (Journal on Advanced Research in Electrical Engineering)*, vol 2, no 1, 5-11. DOI=doi:10.12962/j25796216.v2.i1.32.
- [20] Sulaeman, F. S., & Aji, D. P. 2021. Turn Based Strategy Games to Hone Your Knowledge of Indonesian Culture Based on Android. *1st Paris Van Java International Seminar on Health, Economics, Social Science and Humanities (PVJ-ISHESSH 2020)*, 535, 47-50. DOI= 10.2991/assehr.k.210304.011
- [21] Wagner, F., Schmuki, R., Wagner, T., & Wolstenholme, P. 2006. *Modeling Software with Finite State Machines*. New York: Auerbach Publications. DOI= 10.1201/9781420013641.