

IF2211 Strategi Algoritma

Pemanfaatan Algoritma *Greedy* dalam Aplikasi Permainan “Worms”

Tugas Besar 1



Oleh:

Steven Nataniel - 13519002

Jason Stanley Yoman - 13519019

Kevin Ryan - 13519191

**PROGRAM STUDI INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
SEMESTER II TAHUN 2020/2021**

BAB I

DESKRIPSI TUGAS

Worms adalah sebuah turn-based game yang memerlukan strategi untuk memenangkannya. Setiap pemain akan memiliki 3 worms dengan perannya masing-masing. Pemain dinyatakan menang jika ia berhasil bertahan hingga akhir permainan dengan cara mengeliminasi pasukan worms lawan menggunakan strategi tertentu. Pada tugas besar pertama Strategi Algoritma ini, digunakan sebuah game engine untuk mengimplementasikan permainan Worms. Game engine dapat diperoleh pada [laman berikut](#).

Pada tugas ini, telah diimplementasikan seorang “pemain” worms dengan menggunakan **strategi greedy** untuk memenangkan permainan. Adapun beberapa peraturan umum, yaitu sebagai berikut.

1. Peta permainan berukuran 33x33 cells. Terdapat 4 tipe cell, yaitu air, dirt, deep space, dan lava yang masing-masing memiliki karakteristik berbeda. Cell dapat memuat powerups yang bisa diambil oleh worms yang berada pada cell tersebut.
2. Di awal permainan, setiap pemain akan memiliki 3 pasukan worms dengan peran dan nilai health points yang berbeda, yaitu:
 - a. Commando
Role worm yang memiliki darah lebih dibanding worm lain.
 - b. Agent
Role worm yang memiliki senjata tambahan berupa banana bomb yang dapat memberikan damage besar pada jangkauan area tertentu.
 - c. Technologist
Role worm yang memiliki senjata tambahan berupa snowball yang dapat memberikan efek freeze pada jangkauan area tertentu.
3. Pada setiap round, masing-masing pemain dapat memberikan satu buah command untuk pasukan worm mereka yang masih aktif (belum tereliminasi). Berikut jenis-jenis command yang ada pada permainan:
 - a. Move

- b. Dig
 - c. Shot
 - d. Do Nothing
 - e. Banana Bomb
 - f. Snowball
 - g. Select
4. Command dari kedua pemain akan dieksekusi secara bersamaan (bukan sekuensial) dan akan divalidasi terlebih dahulu. Command juga akan dieksekusi sesuai urutan prioritas tertentu.
 5. Beberapa command, seperti shot dan banana bomb dapat memberikan damage pada worms target yang terkena serangan, sehingga mengurangi health pointsnya. Jika health points suatu worm sudah habis, maka worm tersebut dinyatakan tereliminasi dari permainan.
 6. Permainan akan berakhir ketika salah satu pemain berhasil mengeliminasi seluruh pasukan worms lawan atau permainan sudah mencapai jumlah round maksimum (400 rounds).

BAB II

LANDASAN TEORI

A. Algoritma Greedy

Algoritma Greedy atau algoritma serakah merupakan algoritma yang memecahkan persoalan secara langkah per langkah (step by step) sedemikian sehingga pada setiap langkah algoritma akan mengambil pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepannya (dikenal juga dengan prinsip *take what you can get now*) dan berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

B. Pemanfaatan Game Engine

a. Cara Kerja Game Engine dan Bot

Game Engine menyiapkan state-state game serta semua data yang diperlukan oleh bot. Bot akan menerima data dengan menelusuri suatu folder bernama *rounds*. Lalu, data tersebut akan dipakai untuk *instantiate* object-object pada bot yang akan dimanfaatkan untuk mengaplikasikan algoritma greedy. Setelah, algoritma greedy dijalankan, bot diharuskan mengeluarkan command balik kepada game engine. Game Engine akan merespon balik apakah command valid atau tidak serta membuat log (data dan state pada suatu ronde) pada suatu folder *match log*.

b. Bot

Pada *starter kit*, telah disediakan suatu class Bot yang akan di instantiate setiap ronde. Class ini bertanggung jawab untuk mengembalikan command yang akan dipakai. Class ini telah kami ubah sehingga sangat berbeda dari *starter kit* agar code lebih *clean*, karena sebelumnya class ini juga mengandung fungsi dan prosedur utilitas yang berhubungan dengan worm, map, dan sebagainya. Fungsi dan prosedur utilitas yang berhubungan dengan worm, map, strategy dan sebagainya kami pindah ke class khusus.

c. Entities Package

Class-class pada package entities berfungsi sebagai penyimpan data state game. Seperti yang dijelaskan pada poin cara kerja game engine dan bot. Bot akan membaca suatu file pada setiap ronde yang telah dibuat oleh game engine. File tersebut akan dibaca menjadi string lalu di parse menjadi *JSON* khusus menggunakan library *Google GSON*. Dengan metode *decorator*, setiap field pada class-class entitas memiliki *annotation* serialized name. Semua field yang diberi *annotation* tersebut akan secara otomatis mengambil data dari json dan hal ini dilakukan oleh library *Google GSON* tersebut. Package entitas tersebut memuat class-class seperti worm, my worm, cell, position, powerup, weapon, banana bomb, snowball, game state, dan my player.

d. Enums Package

Berbeda seperti entitas, enums merupakan suatu variasi dari sebuah state. Contohnya enum direction, enum ini akan membuat variasi arah seperti utara, selatan, timur, barat, dan 4 arah mata angin lainnya. Class ini juga sebagian menggunakan *annotation* karena akan digunakan untuk membaca data dari JSON. Package enums ini memuat beberapa class diantaranya, cell type, direction, power up type, dan profession.

e. Command

Class ini bertanggung jawab untuk me-render command dengan constructor yang dimilikinya. Class ini hanya di instantiate saat bot sudah tahu ingin melakukan apa dan mengembalikan ke program utama.

f. Visualiser

Visualiser merupakan program *external* yang dibuat menggunakan *platform Unity*. Visualiser akan membaca match log yang telah di generate oleh Game Engine lalu memvisualisasikannya.

g. Cara Menjalankan Bot

Ubah *bot.json* sesuai nama jar yang telah di compile dan build. Masukkan nama bot dan tim ke dalam file *json* tersebut dan letakkan pada suatu folder. Lalu, siapkan *engine.jar* yang bisa didapatkan dari *starter pack* beserta

game-config.json dan *game-runner-config.json*. Ubah *game-runner-config.json* sesuai directory folder yang mengandung jar dan *bot.json*. Defaultnya, lawannya adalah reference bot, namun bisa diganti dengan bot lain atau bahkan bot kelompok lain. Setelah itu, jalankan *run.bat* untuk Windows User atau *make run* untuk Linux User.

h. Cara Menjalankan Visualiser

Copy folder match logs ke folder Matches pada folder visualizer. Setelah itu, jalankan *entelect-visualiser.exe*, atau *run.bat*, atau *start-visualiser.bat*. Setelah itu akan muncul menu beserta list match yang dapat divisualisasikan.

C. Penambahan Strategy Greedy pada Game Worms

Pada setiap *round* permainan, *game runner* akan meminta *command* untuk dijalankan. Cara kita untuk mensuplai *command* tersebut adalah dengan menjalankan perintah *run* pada bot. Perintah *run* akan mengembalikan *command* yang dapat di-render di *terminal*. Dibaliknya, perintah *run* memanggil beberapa *class* utilitas pada bot seperti utilitas plane (*PlaneUtils*), utilitas game (*GameUtils*), utilitas strategy (*StrategyUtils*), dan utilitas worm (*WormUtils*). Perintah *run* juga akan menggunakan strategi greedy untuk memilih *command* yang sesuai untuk dijalankan.

BAB III

PEMANFAATAN STRATEGY GREEDY

3.1 Proses Mapping Persoalan Worms Menjadi Elemen-Elemen Algoritma Greedy

3.1.1 Himpunan Kandidat

Seluruh himpunan worms dan himpunan command-command pada game (move, dig, shoot, do nothing, snowball, dan select).

3.1.2 Himpunan Solusi

Himpunan commands yang telah digunakan untuk membunuh seluruh worms lawan.

3.1.3 Fungsi Seleksi

Menyatakan strategi greedy yang digunakan melalui pertimbangan-pertimbangan dan uji coba. Pada kasus ini, strategi yang digunakan adalah greedy by profession.

3.1.4 Fungsi Solusi

Penentuan apakah himpunan commands yang dijalankan dapat memenangkan permainan.

3.1.5 Fungsi Kelayakan

Memeriksa apakah worms player melakukan rangkaian command-command yang valid.

3.1.6 Fungsi Objektif

Worms player berhasil membunuh semua worms musuh dengan memanfaatkan strategi greedy.

3.2 Eksplorasi Alternatif Solusi Greedy

3.2.1 Greedy by Nearest Distance

Strategi ini mengutamakan worms musuh yang berada di posisi paling dekat dengan worms player. Strategi ini tidak memandang profession musuh dan program akan mengkalkulasikan jarak current player worm ke seluruh worms musuh dan akan men-set target ke worms yang berjarak paling dekat dengan current player worm.

3.2.2 Greedy by Lowest Health

Strategi ini mengutamakan worms musuh yang memiliki darah atau health yang lebih rendah. Seluruh worms player akan bergerak dan menyerang worms musuh yang memiliki health points paling rendah. Pada kasus early phase game, di mana health points technologist sama dengan healthpoint agent worms player akan menyerang worms musuh yang paling dekat dengan mereka. Ketika darah semua worms tidak ada yang sama, semua worms player akan menyerbu worms musuh dengan darah paling sedikit.

3.2.3 Greedy by Health Pack

Strategi ini mengutamakan health pack yang terletak pada posisi tengah. Strategi ini memaksa worms player untuk jalan ke tengah dan mengambil health pack, terlepas dari apakah health worms player sedang penuh atau tidak. Hal ini dilakukan untuk mencegah worms lawan untuk mengambil health pack. Setelah health pack terambil, worms akan menembak worms musuh yang paling dekat dengan mereka.

3.2.4 Greedy by Profession

Strategi ini menset target ke worms dengan profesi tertentu yang dianggap berbahaya dalam game. Profesi yang diutamakan dalam strategi ini adalah profesi agent. Dengan menyerang dan membunuh agent musuh terlebih dahulu, player dapat memastikan bahwa banana bomb yang dimiliki musuh tidak akan mengancam worms player lagi. Setelah agent musuh mati, target akan di set ke commando musuh dengan pertimbangan bahwa technologist musuh tidak dapat menyerang sebanyak commando dikarenakan ability snowball yang dimilikinya.

3.3 Analisis Efisiensi Dari Alternatif Solusi

Jika ditinjau dari efisiensi program atau game, dapat dikatakan bahwa alternatif solusi greedy by nearest distance memiliki efisiensi paling tinggi. Hal ini dikarenakan worms tidak akan mempertimbangkan banyak hal dan akan langsung bertindak menuju objektif permainan, yaitu mengeliminasi seluruh worms musuh. Namun sayangnya, strategi ini memiliki efektivitas yang kurang dibandingkan strategi-strategi alternatif lainnya.

Pada strategi greedy by lowest health, tingkat efisiensi strategi ini tidak terlalu tinggi dikarenakan ada beberapa worms player yang akan berjalan sangat jauh pada early

game ketika darah worms musuh sudah lagi tidak sama. Efisiensi akan berkurang banyak pada saat perjalanan worms player menuju worms yang memiliki health paling sedikit.

Pada strategi greedy by healthpack, tingkat efisiensi strategi akan menurun pada saat early phase game, di mana worms akan berjalan menuju healthpack yang berada / terletak di tengah map. Ketika worms player melakukan perjalanan, worms player sedang tidak bertindak dan berlaku untuk mencapai tujuan game, yaitu mengeliminasi seluruh musuh. Maka dari itu, tingkah laku worms player inilah yang akan menurunkan efisiensi dari strategi ini

Pada strategi greedy by profession, tingkat efisiensi tidak terlalu tinggi, namun bukan yang paling rendah dibandingkan alternatif solusi lainnya. Hal ini dikarenakan pada early phase game, worm player yang terletak paling jauh dari agent musuh harus berjalan lebih jauh untuk menyerang agent musuh. Tindakan worm player yang paling jauh ini akan menurunkan efisiensi strategi, di mana worm tersebut tidak bertingkah laku untuk menuju objektif permainan pada saat ia melakukan perjalanan tersebut.

3.4 Analisis Efektivitas Dari Alternatif Solusi

3.4.1 Greedy By Nearest Distance

Efektivitas strategi ini kurang baik dikarenakan akan ada banyak flaw yang terjadi ketika worms player bertemu dengan beberapa worms lainnya. Worm milik player akan berkemungkinan untuk mengubah target pada saat peperangan sehingga memungkinkan worm musuh untuk kabur atau menggunakan special ability yang dapat mengalahkan worms player.

3.4.2 Greedy By Lowest Health

Efektivitas strategi ini kurang baik dikarenakan ada kemungkinan bahwa worms player akan diserang di tengah-tengah perjalanan menuju worms yang memiliki health paling sedikit. Dikarenakan strategi greedy, worm yang diserang tidak akan dapat melindungi diri dan menyerang balik worms yang sedang menyerang dirinya.

3.4.3 Greedy By Health Pack

Efektivitas strategi ini kurang baik dikarenakan seluruh worms player akan berjalan ke bagian tengah map terlebih dahulu sehingga menyebabkan worms berkumpul di tengah map dan menjadi sangat berbahaya oleh banana bomb agent musuh

3.4.4 Greedy By Profession

Strategi ini memiliki efektivitas yang paling tinggi dibandingkan alternatif solusi lainnya karena worms musuh yang memiliki special ability yang dapat mengancam worms player akan dimatikan terlebih dahulu sehingga worms player dapat dengan lebih mudah mengeliminasi atau membunuh worms musuh yang lainnya.

3.5 Strategy Greedy yang Dipilih Beserta Alasannya

Berdasarkan hasil analisa kami yang didapatkan dengan melakukan uji coba antara bot kami dengan reference bot, didapat bahwa strategy greedy by profession menghasilkan tingkat kemenangan tertinggi dibandingkan greedy lainnya. Strategi ini juga bersifat fleksibel karena akan beradaptasi dengan kondisi lapangan. Dalam beberapa testing, didapatkan bahwa agent musuh (worm yang bersenjata banana) merupakan ancaman paling berbahaya untuk kemenangan tim. Commando merupakan jenis worm yang paling berbahaya selanjutnya karena kemampuannya untuk menahan lebih banyak serangan. Yang terakhir adalah technologist yang dapat membekukan musuh selama 5 round. Beku selama 5 round menurut kami tidak terlalu berbahaya karena jika sedang bertatap muka, sebuah worm yang dibekukan hanya dapat diserang paling banyak sebanyak 2 kali kecuali jika musuh menggunakan command select untuk meluncurkan serangan beruntun.

Dengan paparan alasan di atas, kami akhirnya memutuskan untuk menggunakan greedy by profession sebagai strategi pilihan kami.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Desain Kelas

4.1.1 Class Worm Utils

Nama Method	Parameter & Return Type	Penjelasan
getCurrentWorm	Param : - Return : MyWorm	Mendapatkan worm kita yang sedang di select.
getEnemy	Param : Profession Return : Worm	Mendapatkan worm musuh berdasarkan profesinya.
getEnemyTechnologist	Param : - Return Worm	Mendapatkan technologist musuh.
getEnemyAgent	Param : - Return Worm	Mendapatkan agent musuh.
getEnemyCommando	Param : - Return Worm	Mendapatkan commando musuh.
getOurTechnologist	Param : - Return Technologist	Mendapatkan technologist kita.
getOurAgent	Param : - Return Agent	Mendapatkan agent kita.
getOurCommando	Param : - Return Commando	Mendapatkan commando kita.
isAlive	Param : Worm Return boolean	Mengecek apakah worm sudah mati.

4.1.2 Game Utils

Nama Method	Parameter & Return Type	Penjelasan
isValidCoordinate	Param : <ul style="list-style-type: none">- X int- Y int Return : boolean	Mengecek apakah koordinat (x,y) di game merupakan koordinat yang valid.
lookup	Param : <ul style="list-style-type: none">- X int- Y int Return : Cell	Mengembalikan Entitas Cell pada koordinat (x,y).

4.1.3 Plane Utils

Nama Method	Parameter & Return Type	Penjelasan
euclideanDistance	Param : <ul style="list-style-type: none">- A Cell- B Cell Return : int	Mengembalikan jarak antara 2 cell dengan metode euclidean distance dan dibulatkan kebawah.
resolveDirection	Param : <ul style="list-style-type: none">- A Cell- B Cell Return : Direction	Diberikan 2 buah titik, kembalikan arah mata angin yang berasal dari titik A dan berakhir di B.
getSurroundingCell	Param : <ul style="list-style-type: none">- X int- Y int Return : List<Cell>	Diberikan sebuah koordinat cell, fungsi ini akan mengembalikan cell-cell tetangganya.
constructFireDirectionLines	Param : <ul style="list-style-type: none">- cell Cell- range int Return : List<Cell>	Mengembalikan list area yang bisa ditembak jika berpusat di cell dan memiliki jarak penembakan sebesar range
generateLine	Param: <ul style="list-style-type: none">- Starting Cell- End Cell Return : List<Cell>	Mengembalikan path cell dalam bentuk list bermula dari starting dan berakhir di end
nextLine	Param : <ul style="list-style-type: none">- Starting Cell	Mengembalikan cell yang berikutnya harus dituju ketika

	- End Cell Return Cell	worm ingin bergerak dari starting ke end
getBananaBombRange	Param : - Return List<Cell>	Mengembalikan list cell yang dapat ditembak banana oleh agent
getFreezerRange	Param : - Return List<Cell>	Mengembalikan list cell yang dapat ditembak snowball oleh technologist
getShootingArea	Param : - Center Cell - Radius int Return List<Cell>	Mengembalikan list cell yang akan dikenai efek oleh banana atau freezer jika dilemparkan di center
getBananaArea	Param : Center Cell Return List<Cell>	Mengembalikan list cell yang akan dikenai efek oleh banana jika dilemparkan di center

4.1.4 Strategy Utils

Nama Method	Parameter & Return Type	Penjelasan
setTargetWorm	Param : - Return Worm	Mengembalikan musuh yang dijadikan target
agentCanShootTwoWorms	Param : - Return : Cell	Mengembalikan sebuah cell jika agent dapat melempar banana dan mengenai 2 musuh sekaligus
technologistCanShootTwoWorms	Param : - Return : Cell	Mengembalikan sebuah cell jika technologist dapat melempar snowball dan mengenai 2 musuh sekaligus
getAvailableShoot	Param : - Worm MyWorm Return : Cell	Mengembalikan cell yang terdapat musuh di range tembakannya
desperateAgent	Param : - Return Cell	Mengembalikan cell ketika agent dapat menembak banana dan sisa health ≤ 40

desparateTechnologist	Param : - Return Cell	Mengembalikan cell ketika technologist dapat menembak snowball dan sisa health \leq 40
bananaCanKill	Param : - Return Cell	Mengembalikan cell ketika agent dapat menembak banana dan musuh yang ditembaknya mati
bananaHitTarget	Param : - Target Worm Return Cell	Mengembalikan cell yang jika ditembakkan banana, akan mengenai target
freezeHitTarget	Param : - Target Worm Return Cell	Mengembalikan cell yang jika ditembakkan snowball, akan mengenai target
nearHealthPack	Param : - Center Cell Return Cell	Mengembalikan cell jika sekelilingnya terdapat health pack

4.2 Implementasi

Berikut adalah *pseudocode* untuk strategi *greedy*

If selection_token > 0 then

If currentWorm bukanlah agent && agent bisa shoot banana ke 2 worm sekaligus then

Agent shoot banana ke cell terpilih

If currentWorm bukanlah technologist && technologist bisa shoot snowball ke 2 worm sekaligus then

Technologist shoot snowball ke cell terpilih

If profesi currentWorm == Agent then

If agent bisa shoot banana ke 2 worm sekaligus then

Agent shoot banana ke cell terpilih

If agent bisa membunuh worm musuh dengan banana then

Agent shoot banana ke cell terpilih

If health \leq 30 && ada cell yang jika ditembakkan banana dapat kena musuh then

```

        Agent shoot banana ke cell terpilih
    Target = StrategyUtils.setTargetWorm()
    If agent bisa menembak banana ke target then
        Agent shoot banana ke cell terpilih
    If agent bisa menembak ke salah satu musuh then
        Tembak musuh
    Bergerak ke tempat target

Else if profesi currentWorm == Technologist then
    If technologist bisa shoot snowball ke 2 worm sekaligus then
        Technologist shoot snowball ke cell terpilih
    If health <= 30 && ada cell yang jika ditembakkan snowball dapat kena
        musuh then
        Technologist shoot snowball ke cell terpilih
    Target = StrategyUtils.setTargetWorm()
    If technologist bisa menembak snowball ke target then
        Technologist shoot snowball ke cell terpilih
    If technologist bisa menembak ke salah satu musuh then
        Tembak musuh
    Bergerak ke tempat target

Else if profesi currentWorm == Commando then
    If commando bisa menembak ke salah satu musuh then
        Tembak musuh
    Bergerak ke tempat target

```

4.3 Struktur Data

4.3.1 Standar

Seperti yang sudah dijelaskan pada bab 2, data-data dari game engine disimpan pada object-object dari class entitas. Sebagaimana akan dijelaskan pada tabel berikut.

Class	Tanggung Jawab
Position	Menyimpan posisi worm.
Cell	Turunan dari position. Memiliki data tambahan berupa power up dan tipe cell.
GameState	Berisi data tentang konfigurasi game serta data player, map, dan musuh.
MyPlayer	Berisi data health, score, jumlah seleksi worm, worm milik player.
MyWorm	Turunan dari worm. Memiliki informasi lebih tentang senjata.
Opponent	Mirip seperti MyPlayer hanya saja datanya hanya score dan list worm.
PowerUp	Struktur data untuk power up.
Weapon	Berisi data tentang senjata dasar para worm berupa range dan damage.
Worm	Berisi data tentang suatu worm, seperti health, position, diggingRange, movementRange, roundsUntilUnfrozen, dan profession.
CellType	Enum untuk jenis-jenis cell (DEEP_SPACE, DIRT, AIR, LAVA)
Direction	Enum untuk arah mata angin (N, W, E, S, NE, NW, SW, SE)
PowerUpType	Enum untuk jenis-jenis power up (HEALTH_PACK)
Profession	Enum untuk profession (AGENT,

	TECHNOLOGIST, COMMANDO)
--	-------------------------

4.3.2 Tambahan

Adapun class-class tambahan yang dibuat untuk membantu dan memudahkan mengimplementasikan strategy greedy diatas, yaitu sebagai berikut. Ada juga class dan enum standar diatas yang ditambah untuk melengkapi kebutuhan, contohnya seperti cell type, sebelumnya tidak ada lava, lalu ditambahkan. Ditambahkan enum profession.

Class	Tanggung Jawab
Agent	Turunan dari class MyWorm. Bedanya class ini memiliki data mengenai Banana Bomb yang dimiliki.
Commando	Turunan dari class MyWorm.
Technologist	Turunan dari class MyWorm. Bedanya class ini memiliki data mengenai Snowball yang dimiliki.
BananaBomb	Memuat info mengenai senjata banana bomb.
Snowball	Memuat info mengenai senjata snowball.

4.4 Analisis dari Desain Solusi Algoritma Greedy

4.4.1 Solusi Optimal

Solusi optimal akan tercapai pada saat strategi lawan yang digunakan adalah strategi greedy by healthpack, greedy by nearest distance. Pada strategi greedy by healthpack, worms player dapat dengan mudah membunuh agent musuh dikarenakan commando player dan technologist player akan bertemu dengan agent musuh di tengah-tengah. Kondisi ini tercapai di posisi yang menepi dari tengah map sehingga ketika commando musuh dan technologist musuh akan mengambil healthpack, posisi mereka akan menjadi berdekatan. Hal ini dapat di

counter oleh strategi greedy by profession dimana agent player akan melempar banana bomb dan mengenai beberapa worms sekaligus. Pada strategi greedy by nearest distance, dikarenakan commando dan technologist player berjalan sesaat setelah game dimulai, maka jarak antara commando musuh dan technologist musuh ke agent player menjadi yang paling dekat sehingga agent player akan berperang dengan commando dan technologist musuh terlebih dahulu dan menggunakan banana bomb untuk memenangkan pertarungan. Setelah itu, commando player dan technologist player akan dapat membunuh commando dan technologist musuh dengan mudah.

4.4.2 Solusi Tidak Optimal

Solusi tidak optimal akan tercapai ketika melawan strategi greedy by lowest health. Hal ini dikarenakan technologist player akan menjadi worm pertama yang terserang sehingga worm technologist player akan mati terlebih dahulu. Karena hal ini, snowball player akan terkuras habis untuk melakukan freeze pada worms musuh. Setelah technologist mati, maka commando akan menjadi next target dari musuh. Namun, pada kasus ini, technologist musuh dapat melakukan freeze pada commando player sehingga commando player tidak akan dapat menyerang worms player. Pada saat agent player berjalan sampai ke tujuannya, agent akan berakhir di kondisi 1 vs 3 sehingga agent akan kalah pada perang.

Solusi tidak optimal akan tercapai pada strategi lain yang menyebabkan commando musuh dan technologist musuh bergerak menjauhi agent. Hal ini dikarenakan agent hanya akan mulai bergerak pada saat jaraknya ke titik tengah di antara commando musuh dan technologist musuh lebih besar daripada jarak antara salah satu worm musuh ke agent player. Masalah ini hanya dapat diatasi dengan mengeset sebuah nilai round sehingga ketika game mencapai round tersebut, agent akan mulai bergerak terlepas dari jarak antara commando musuh, technologist musuh, dan agent player.

BAB V

KESIMPULAN DAN SARAN

Kelompok 2 tugas besar Strategi Algoritma IF2211 berhasil menyelesaikan tugas besar 1 dengan efektif dan efisien. Dengan memanfaatkan strategi algoritma greedy, kelompok dapat menyusun strategi yang memiliki tingkat kemenangan yang cukup besar (75-90%) melawan *reference bot*. Adapun strategi yang diterapkan adalah greedy by profession; di mana worms pada game ini akan membunuh agent musuh terlebih dahulu, lalu mengganti target ke commando, dan diakhiri dengan target ke technologist musuh. Tetapi ada kemungkinan,

Saran untuk tugas besar ini adalah dengan melakukan lebih banyak melakukan uji coba bersama kelompok-kelompok lain untuk mengevaluasi dan memperbaiki strategi yang digunakan.

DAFTAR PUSTAKA

Munir, Rinaldi. Tugas-Besar-1-IF2211-Strategi-Algoritma-2021.pdf. 5 Februari 2021.

<https://www.tutorialspoint.com/what-to-use-serializedname-annotation-using-gson-in-java>.

Diakses 12 Februari 2021.

<https://github.com/EntelectChallenge/2019-Worms/blob/develop/game-engine/game-rules.md>.

Diakses 12 Februari 2021.