

1. PENDAHULUAN

1.1 Latar Belakang

Permainan *Real Time Strategy* atau RTS merupakan bagian dari kategori permainan strategi yang saat ini sedang populer di pasaran (Barambones et al., 2023). Kategori ini sering kali tersaji dalam bentuk peperangan militer yang berlangsung secara *real time* sehingga pemain dan NPC (*Non Player Character*) diharuskan untuk membuat keputusan maupun taktik yang efisien untuk menghadapi situasi dan lingkungan permainan yang berubah-ubah (Barambones et al., 2023; Bavelier & Green, 2019; Buro, 2003; Huat & Teo, 2012; Kang et al., 2022; Kozik et al., 202; Liu et al., 2016; Nicolau et al., 2017; Ponsen et al., 2006). Sebagai salah satu unsur penting dari suatu permainan *video*, menghasilkan *Artificial Intelligence* (AI) NPC yang bervariasi menjadi tantangan dalam mengembangkan permainan RTS bagi pengembang karena perancangan yang dilakukan secara manual (Buro, 2003; Kang et al., 2022; Riedl & Zook, 2013; Robertson & Watson, 2014).

Dalam mengembangkan *behavior* NPC yang bervariasi untuk NPC permainan RTS, pada penelitian ini akan digunakan teknik *Procedural Content Generation* (PCG) dan algoritma *evolutionary* (EA). PCG merupakan teknik perancangan konten secara otomatis menggunakan algoritma yang telah ditentukan (de Pontes & Gomes, 2020; Lara-Cabrera et al., 2014; Melotti & de Moraes, 2019; Soares De Lima et al., 2019). Meskipun teknik ini telah digunakan sejak tahun 1980 (de Pontes & Gomes, 2020; Gravina et al., 2019; Liapis et al., 2013), hingga saat ini PCG telah banyak diterapkan untuk mengembangkan berbagai konten seperti *level*, *maps*, *items*, senjata, *quest*, karakter, *rules* (de Pontes & Gomes, 2020; Liapis et al., 2013; Liu et al., 2021; Summerville et al., 2018;), misi (Lara-Cabrera et al., 2014), *rewards*, *resource distribution*, posisi, kombinasi dialog, hingga pola *behavior* (Melotti & de Moraes, 2019). Disisi lain, EA merupakan algoritma optimasi yang terinspirasi dari proses evolusi alam sehingga mampu memperoleh solusi optimal dari suatu permasalahan kompleks (Masek et al., 2023; Slowik, A. & Kwasnicka, H., 2020; Weber & Notargiacomo, 2020). Pada algoritma ini, solusi terbaik merupakan kromosom dengan skor *fitness* tertinggi diantara sekumpulan kromosom berskala besar (de Pontes & Gomes, 2020; Thaker et al., 2015; Weber & Notargiacomo, 2020). Hingga saat ini, kombinasi PCG dan algoritma EA terbukti dapat memberikan hasil yang efektif dan telah pernah diaplikasikan dalam menghasilkan *behavior agent* untuk berbagai macam permainan seperti Checkers, Chess, Pac-Man, Super Mario Bros, permainan FPS (*First Person*

Shooter), balapan mobil (Dockhorn et al., 2023; Schrum et al., 2023), DEFCON (Lim et al., 2010) dan permainan *Capture the Flag* (Fronek et al., 2020).

Sebelumnya, metode evolusi pernah digunakan dalam menghasilkan *behavior tree* komandan pada penelitian berjudul *Evolving Behavior Trees for the Commercial Game DEFCON* (Lim et al., 2010). Dimana *behavior tree* komandan dari penelitian ini berhasil mengungguli AI asli dari permainan DEFCON yang berkategori RTS lebih dari 50% ketika pengujian. Kemudian, pada tahun 2022 penelitian serupa berjudul *Procedural Creation of Behavior Trees for NPCs* dilakukan oleh Fronek untuk mengembangkan *behavior agent* permainan *Capture The Flag*. Dimana *behavior* yang dihasilkan dinilai memiliki perilaku cerdas dan mampu untuk belajar dari pengalaman tempur sebelumnya karena menggunakan algoritma *evolutionary*. Disisi lain, Penggunaan *behavior tree* pada metode evolusi juga dikarenakan *behavior tree* merupakan model AI yang mudah untuk dibaca dan populer sebagai model AI NPC dalam industri *video game*, *behavior tree* juga menggunakan struktur hierarki yang fleksibel dan modular sehingga sesuai dengan karakteristik kromosom yang dibutuhkan dalam algoritma *evolutionary* (Colledanchise et al., 2019; Colledanchise & Natale, 2022; Hong et al., 2023; Masek et al., 2023; Zhang et al., 2018).

Namun berbeda dengan penelitian sebelumnya yang menggunakan lingkungan permainan *turn-based* dengan sebagian aspek *Real Time Strategy* (RTS), dalam penelitian ini akan mengembangkan *behavior agent* berkategori RTS *non-turn-based* yang bervariasi menggunakan permainan *War Simulator* sebagai lingkungannya. *Behavior* yang akan dikembangkan terbagi atas 3 jenis yaitu *shooter*, *Swordsman*, dan *staff fighter* dengan fokus berbeda menggunakan teknik PCG dan algoritma evolusi. Hasil akhir dari skripsi ini akan berupa sebuah permainan *War Simulator* yang melakukan simulasi peperangan antar 2 tim dengan strategi serta kombinasi *agent* yang dapat ditentukan pemain melawan formasi tim musuh. Selain itu, untuk memberikan pengalaman dan *behavior agent* yang lebih beragam, setiap permainan dimulai *behavior tree* tiap *agent* akan dirancang ulang menggunakan model *behavior* hasil *training*. Kemudian sebagai pengujian keberhasilan skripsi ini, akan dilakukan pengujian variasi melalui beberapa sampel *behavior* hasil *training*. Selain itu, juga akan dilakukan pengambilan kuesioner kepada sejumlah volunteer yang telah memainkan permainan *War Simulator* sebanyak 3 kali dengan respon berupa skala likert 1-5 sebagai tingkat persetujuan volunteer terhadap pernyataan yang diberikan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ada, dapat dirumuskan permasalahan sebagai berikut:

1. Apakah penerapan algoritma *evolutionary* dan teknik *procedural content generation* dapat menghasilkan *behavior tree* yang bervariasi untuk *agent* permainan *War Simulator*?

1.3 Tujuan Penelitian

Tujuan dari skripsi ini adalah untuk mengetahui apakah penerapan *evolutionary algorithm* dapat menghasilkan *behavior tree* yang bervariasi untuk *agent* permainan *War Simulator*.

1.4 Ruang Lingkup

Berikut merupakan ruang lingkup pengerjaan skripsi ini:

1. Menggunakan Unreal Engine 5.1 sebagai *game engine*.
2. Permainan yang dirancang merupakan permainan *War Simulator* bergenre RTS.
3. Tiap *agent* dalam permainan ini kemudian akan menggunakan *behavior tree* yang diperoleh dari hasil evolusi dari 2 *behavior tree* acak yang dipilih dari 32 *behavior* dengan skor terbaik dari proses *training* untuk menghasilkan variasi yang beragam setiap permainan dimainkan.
4. Selama proses *training* setiap generasi akan terdiri dari 32 tim yang mewakili setiap kromosom *behavior tree*.
5. Proses *training* akan dibagi menjadi 2 macam yaitu *training* terpisah per *agent* dan *training cluster* yang langsung menggunakan ketiga jenis *agent* dalam 1 tim.
6. Pada *training* terpisah setiap tim kemudian akan beranggotakan 5 *agent* dengan jenis yang sama dan menggunakan *behavior tree* hasil terjemahan dari kromosom yang sama sedangkan pada *training cluster* akan terdiri dari 6 *agent* dengan perbandingan 2 *swordsman*, 2 *shooter*, dan 2 *staff-fighter*.
7. Khusus generasi pertama kromosom *behavior tree* milik setiap tim akan dihasilkan dengan ukuran serta *node* yang acak kemudian diberi skor sebesar 1 sebagai permulaan.
8. Alur seleksi:
 - a. Menghadapkan setiap tim secara acak satu sama lain dalam arena yang terpisah untuk saling bertarung.

- b. Terdapat 3 kondisi yang dapat dihasilkan dari pertandingan antar tim:
 - i. Menang: ketika terdapat 1 atau lebih *agent* yang tersisa dalam tim.
 - ii. Kalah: ketika tidak terdapat *agent* yang tersisa dalam tim.
 - iii. Seri: ketika permainan tetap berlangsung hingga 2 menit.
 - c. Tim yang menang akan memperoleh peningkatan skor sebesar $1 + \text{roundDown}(\text{skor tim yang kalah}/2)$ dan lolos ke generasi selanjutnya dengan kromosom *behavior tree* yang sama. Sedangkan tim yang kalah tidak akan memperoleh perubahan.
 - d. Ketika pertandingan menghasilkan kondisi “seri” maka, setiap tim akan dicek terlebih dahulu, yaitu:
 - i. Jika belum pernah menang sebelumnya: Kromosom akan dirancang baru seperti perancangan kromosom generasi pertama dan diberi skor sebesar 1.
 - ii. Jika telah pernah menang sebelumnya: kedua kromosom akan tidak akan dihapus namun akan dipastikan untuk tidak saling berhadapan di iterasi berikutnya.
9. Alur *crossover* dan mutasi:
- a. Untuk menentukan ukuran *behavior tree* dan *node* yang diwarisi, maka digunakan *rule* sebagai berikut, sebagai konsep *crossover* dan mutasi menggunakan perhitungan skor kedua tim:
 - i. $X < Sv*3 \rightarrow$ gen dari kromosom tim yang menang.
 - ii. $Sv*3 \leq X < (Sv*3+Sd) \rightarrow$ gen dari kromosom tim yang kalah.
 - iii. $X \geq (Sv*3+Sd) \rightarrow$ tidak mewarisi dari generasi sebelumnya namun *generate* baru secara acak.
 - iv. Keterangan:
 - $X \rightarrow$ angka acak dari 0 sampai t
 - $t = Sv*3 + Sd + 5$
 - $Sv \rightarrow$ skor tim yang menang.
 - $Sd \rightarrow$ skor tim yang kalah.
 - b. Kromosom hasil perancangan baru kemudian akan memiliki skor sebesar $1 + (Sv+Sd)/10$.
 - c. Jika *node* diambil dari *tree* dengan ukuran yang lebih kecil atau dengan simpul berbeda, maka pewarisan akan dialihkan ke *tree* lainnya, atau jika tidak maka akan dihasilkan secara acak.

- d. Kromosom hasil perancangan kemudian akan disimpan hingga seluruh proses evolusi pada iterasi tersebut telah selesai. Setelahnya akan diambil 32 kromosom dengan skor tertinggi untuk digunakan pada iterasi sebelumnya, dan kromosom sisanya akan dieliminasi.
10. Terdapat 3 jenis *behavior* yang akan di-*generate* dengan penambahan bonus skor yang dihitung dari jumlah *damage* yang diterima dan diberikan selama pertandingan, yaitu:
 - a. *Shooter*: Sebagai penyerangan jarak jauh, *behavior* ini berfokus untuk memberikan *damage* dan menghindari serangan musuh.
 - b. *Swordsman*: Sebagai perisai dan penyerang jarak dekat, *behavior* ini berfokus untuk menerima *damage* selagi memberikan *damage* pada musuh.
 - c. *Staff Fighter*: Sebagai penyerangan jarak dekat yang berfokus pada memberikan *damage*, *behavior* ini memiliki keseimbangan antara memberikan *damage* dan menerima *damage*.
 11. Berikut *node* atau gen kondisi yang digunakan dalam penyusunan kromosom *behavior tree*:
 - a. Jarak dengan *agent* musuh terdekat.
 - b. Jarak dengan *agent* tim terdekat.
 - c. Jumlah *Health Point*.
 - d. Jumlah amunisi tersisah.
 - e. Jumlah musuh.
 - f. Jumlah anggota tim.
 - g. Jarak dengan *item buff* terdekat.
 - h. Cooldown *skill heals*.
 - i. Cooldown *attack*.
 - j. Total HP *agent* tim.
 - k. Total HP *agent* musuh.
 12. Berikut *node* atau gen aksi yang digunakan oleh *agent* dalam kromosom *behavior tree*:
 - a. Berjalan ke *agent* tim terjauh
 - b. Berjalan ke *agent* tim terdekat
 - c. Berjalan ke *agent* musuh terdekat
 - d. Berjalan ke *agent* musuh terjauh
 - e. Mengambil item terdekat
 - f. *Reload* (memulihkan jumlah amunisi *shooter*).

- g. *Skill heal* (memulihkan 20% HP secara bertahap dalam 5 detik).
 - h. *Attack*.
 - i. Lari (bergerak ke arah berlawanan dari *agent* musuh terdekat).
13. Berikut status yang dimiliki *agent*:
- a. HP (*Health Point*).
 - b. *Attack*.
 - c. Amunisi.
 - d. Cooldown dari *skill heal*.
 - e. Cooldown *attack*.
14. Berikut *rule* yang terdapat dalam permainan ini:
- a. Dalam proses perancangan kromosom generasi pertama atau crossover, jika kromosom baru tidak memiliki gen *attack* maka proses perancangan akan diulang kembali.
 - b. Ketika *shooter* menjalankan aksi menyerang dengan amunisi = 0, maka *agent* akan langsung melakukan aksi *reload*.
 - c. *Behavior Tree* akan bekerja secara *in game real time*.
 - d. Operator pada *node* kondisi terbagi menjadi 2 macam, yaitu: \leq (*less or equal*) dan \geq (*greater or equal*) dengan parameter yang ditetapkan secara acak sesuai dengan *range* parameternya.
 - e. *Behavior tree* yang dihasilkan menggunakan model *binary tree* dengan ukuran 7-23 *node* yang terdiri dari: 4-12 *node* aksi sebagai *leaf node* dan 3-11 *node* kondisi *parent node*.
 - f. Setiap pertarungan akan menempati arena tertutup berukuran 4500 x 3000 unit, dimana arena akan dibagi menjadi 3 wilayah sebesar 1500 unit yaitu: wilayah tim A (tim pemain), wilayah netral, dan wilayah tim B.
 - g. Setiap 20 detik sejak detik ke-2 permainan dimulai, terdapat *item buff* yang di-*spawn* di arena dengan lokasi yang acak.
 - h. Setiap *item buff* memiliki efek meningkatkan *attack speed* dan *movement speed* *agent* sebesar 25% yang dapat bertahan selama 15 detik dan tidak dapat ditumpuk.
 - i. Khusus pada *training cluster*, penambahan bonus skor tidak diterapkan karena *training* bertujuan untuk menghasilkan tim yang berfokus pada kemenangan lepas dari peran *agent*.

15. Aset pendukung seperti model *agent*, BGM (*Background music*), SFX (*Sound Effect*), VFX (*Visual Effect*), animasi, dan aset lainnya yang digunakan dalam skripsi ini akan menggunakan aset gratis yang diperoleh melalui platform aset yang ada seperti: *unreal marketplace*, *mixamo*, dan sebagainya yang kemudian disesuaikan sendiri.
16. Hasil akhir skripsi ini akan berupa sebuah project permainan *War Simulator* dimana pemain dapat melakukan simulasi pertarungan dengan strategi serta kombinasi *agent* yang ditentukan melawan formasi musuh.
17. Pengujian variasi *behavior* hasil *training* akan dilakukan dengan analisis 3 sampel *behavior* dari tiap *agent* dan tipe *training*, serta analisis *behavior* dengan skor tertinggi pada iterasi 0, 5, 10, 20, 50, dan 100.
18. Pengujian *behavior* yang dikembangkan juga akan dilakukan dengan pengambilan kuesioner pada sejumlah volunteer yang telah memainkan permainan yang dibuat sebanyak 3 kali dengan respon berupa skala likert 1-5 sebagai tingkat persetujuan volunteer terhadap pernyataan yang diberikan. Pernyataan yang digunakan adalah sebagai berikut:
 - a. Behavior agent terkesan cukup pintar.
 - b. Behavior agent Shooter sesuai dengan karakteristik penyerang jarak jauh.
 - c. Behavior agent Swordsman sesuai dengan karakteristik prajurit pelindung.
 - d. Behavior agent Staff Fighter sesuai dengan karakteristik prajurit tongkat.
 - e. Behavior agent Shooter sesuai dengan yang diharapkan.
 - f. Behavior agent Swordsman sesuai dengan yang diharapkan.
 - g. Behavior agent Staff fighter sesuai dengan yang diharapkan.
 - h. Behavior setiap agent (training tim) dapat bekerja sama dengan baik.

1.5 Metodologi Penelitian

Berikut merupakan garis besar pengerjaan skripsi ini:

1. Studi literatur
 - a. *Unreal Engine 5.1*
 - b. *Evolutionary Algorithm*.
 - c. *Procedural Content Generator*.
 - d. Analisis proyek dan penelitian sebelumnya.
2. Perancangan dan implementasi program
 - a. Perancangan *environment*.

- b. Perancangan *task & logic agent*.
 - c. Perancangan *rule* permainan.
 - d. *Training agent*.
 - e. Perancangan proyek permainan.
3. Pengujian dan analisis model
 - a. Pengujian hasil permainan kepada sejumlah volunteer menggunakan survei dengan respon berupa skala likert 1-5.
 4. Pengambilan kesimpulan
 - a. *Pre-Processing* data hasil pengujian.
 - b. Membuat *insight* dari hasil pengujian.
 5. Pembuatan laporan
 - a. Perancangan laporan berdasarkan hasil yang diperoleh.

1.6 Manfaat Penelitian

Manfaat yang dapat diperoleh dari skripsi ini adalah peningkatan pengetahuan mengenai pengembangan AI NPC berupa *behavior tree* yang bervariasi menggunakan teknik *procedural content generation* dengan algoritma *evolutionary* serta dapat menjadi inspirasi dalam pengembangan AI dalam industri *video game* yang menggunakan Unreal Engine 5.1 kedepannya.

1.7 Sistematika Penelitian

Berikut ini adalah sistematika penulisan yang digunakan dalam penyusunan skripsi ini:

BAB I : PENDAHULUAN

Bab ini berisikan latar belakang permasalahan, perumusan masalah, tujuan skripsi, ruang lingkup, metode penelitian, dan sistematika penulisan skripsi ini.

BAB II : LANDASAN TEORI

Bab ini berisikan teori-teori serta studi yang menjadi menjadi landasan perancangan skripsi ini.

BAB III : ANALISA DAN DESAIN SISTEM

Bab ini berisikan analisis dan desain sistem yang dibuat dalam skripsi ini.

BAB IV : IMPLEMENTASI SISTEM

Bab ini berisikan implementasi sistem berdasarkan desain dan analisis dari Bab III

BAB V : PENGUJIAN SISTEM

Bab ini berisikan pengujian sistem yang telah dibuat pada Bab IV.

BAB VI : KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang diperoleh dari hasil pengujian sistem dan saran yang diberikan untuk membantu pengembangan selanjutnya.