

## Algoritme Genetika untuk Desain Level Dinamis pada *Game* Edukasi Kebakaran Hutan

Danny Wijaya, Hanny Haryanto\*, Erna Zuni Astuti, Wijanarto

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro  
Jl. Imam Bonjol 207, Semarang, Indonesia 50131

\*email: hanny.haryanto@dsn.dinus.ac.id

(Naskah masuk: 2 September 2020; diterima untuk diterbitkan: 2 April 2021)

**ABSTRAK** – Kebakaran hutan dianggap sebagai ancaman yang memiliki potensi besar dalam menghambat pembangunan yang berkelanjutan karena efeknya langsung bagi ekosistem. Sosialisasi kebakaran hutan terhadap masyarakat dapat dilakukan melalui berbagai cara, salah satunya menggunakan game. Game dapat berfungsi untuk menumbuhkan pengetahuan, kreativitas, emosi, kecerdasan dan nilai-nilai sikap. Dalam pembuatan game, desain level merupakan salah satu faktor utama yang menentukan kualitas gameplay dari suatu game. Penelitian ini menggunakan algoritme genetika (GA) sebagai metode Procedural Content Generation dalam desain level dari game edukasi kebakaran hutan. Algoritma genetika digunakan sebagai metode desain level berupa maze hutan dalam game edukasi kebakaran hutan. Implementasi metode GA menggunakan model Steady State Population, yang berarti dalam setiap generasi baru akan dihasilkan dua kromosom baru dari hasil crossing kromosom generasi sebelumnya yang telah diseleksi berdasarkan nilai fitness. Model ini digunakan agar kromosom dengan nilai fitness terendah digantikan oleh kromosom baru hasil crossing dengan tujuan mendorong populasi menuju nilai fitness yang diinginkan. Level yang terbentuk memiliki kriteria-kriteria seperti harus dapat diselesaikan, pohon-pohon dalam map tersebar secara merata, dan titik api sebagai awal dimulainya kebakaran berjumlah sedikit sehingga dapat menghasilkan bentuk level yang optimal dari hasil iterasi dengan memilih kandidat terbaik.

**Kata Kunci** – Algoritma Genetika; desain level; game; kebakaran hutan; Procedural Content Generation.

## Genetic Algorithm for Dynamic Level Design on Education *Game* of Forest Fire

**ABSTRACT** – Forest fires could hinder sustainable development because of its direct effect on ecosystem. Public socialization of forest fires can be done in various ways, one of which is using games. Games can function to foster knowledge, creativity, emotion, intelligence and attitude values. In game development, level design is one of the main factors that determine the quality of the gameplay. This study uses a genetic algorithm (GA) as a Procedural Content Generation in the level design of a forest fire education game. GA is used as a level design method in the form of forest maze level. It uses Steady State Population model. Each new generation produce two new chromosomes from the crossing of the previous selected generation based on the fitness value. Chromosomes with the lowest fitness value are replaced by new chromosomes from the crossing to push the population towards the desired fitness value. The formed level has criteria such as must be able to be completed, the trees in the map are evenly distributed, and the number of hotspots as the start of the fire is small so that it can produce the optimal level shape from the iteration results by selecting the best candidate.

**Keywords** - forest fire; game; Genetic Algorithm; level design; Procedural Content Generation.

### 1. PENDAHULUAN

Dalam dekade terakhir ini masalah kebakaran hutan semakin menarik perhatian dari pihak lokal maupun internasional karena memiliki dampak terhadap lingkungan dan ekonomi, khususnya

setelah terjadinya bencana El Nino pada 1997 sampai 1998 yang menghanguskan hutan seluas 25 juta hektar di seluruh dunia [1]. Kebakaran dianggap sebagai ancaman yang memiliki potensi besar dalam menghambat pembangunan yang berkelanjutan

karena efeknya langsung bagi ekosistem. Sebagai contoh di Asia Tenggara, dampak kebakaran hutan cukup signifikan sehingga terjadi Perjanjian Lintas Batas Pencemaran Kabut oleh negara - negara Association of Southeast Asian Nations (ASEAN) yang terjadi pada bulan Juni 2002 bertempat di Kuala Lumpur [1].

Kebakaran hutan memiliki beberapa dampak langsung, yang pertama adalah menimbulkan penyakit infeksi saluran pernafasan akut bagi masyarakat yang tinggal disekitar area kebakaran. Kedua, secara sosial dan ekonomi masyarakat dirugikan karena berkurangnya efisiensi kerja dan transportasi, seperti contoh sekolah dan kantor terpaksa diliburkan serta transportasi penghubung terganggu. Ketiga, kebakaran hutan yang cukup besar dapat menyebabkan terjadinya transboundary haze pollution (pencemaran asap lintas batas) ke wilayah negara-negara tetangga seperti Singapura, Malaysia dan Brunei Darussalam [2]. Walaupun perhatian terhadap masalah kebakaran hutan terus meningkat dan sudah ada banyak upaya yang dilakukan untuk menghindari, mengurangi dan menekan dampak kebakaran hutan, bencana yang sama seperti kabut asap masih terjadi pada tingkatan yang berbeda setiap tahunnya di seluruh dunia. Dari permasalahan di atas ada beberapa solusi untuk memberikan edukasi tentang kebakaran hutan kepada masyarakat seperti melalui sosialisasi kepada masyarakat, himbuan melalui video tentang upaya pencegahan kebakaran hutan, serta juga dapat dilakukan melalui *Video Game*, itulah mengapa penelitian ini menggunakan *Video Game*.

*Game* biasanya dibuat untuk hiburan penghilang stres dan juga dibuat sebagai sarana pendidikan untuk anak yang ingin belajar tanpa merasakan rasa bosan pada suatu hal yang ingin dipelajari. *Game* dapat berfungsi untuk menumbuhkan pengetahuan, kreativitas, emosi, kecerdasan dan nilai-nilai sikap, oleh karena itu *game* dapat digunakan sebagai sarana edukasi kebakaran hutan.

*Video Game* adalah permainan interaktif dengan antarmuka pengguna melalui gambar yang dihasilkan oleh piranti video. *Video game* pada umumnya menyediakan sistem penghargaan, seperti skor dan medali, yang didapat dari tingkat keberhasilan dalam menyelesaikan tahap - tahap yang terdapat dalam permainan. *Game* dirancang agar murid menjadi sadar dan termotivasi untuk belajar dan sebagai penghibur dari *game* yang dimaksud [3]. Penelitian dari Massachusetts Institute of Technology (MIT) berhasil membuktikan bahwa *game* dapat digunakan untuk meningkatkan logika dan pemahaman pemain terhadap suatu masalah melalui *game* yang dinamai *Scratch* [4]. Jika dibandingkan dengan metode pembelajaran konvensional, *game* edukasi memiliki keunggulan

dalam beberapa aspek, salah satu keunggulan yang signifikan adalah animasi yang dapat meningkatkan daya ingat sehingga materi pelajaran dapat diingat dalam waktu yang lebih lama dibandingkan dengan metode konvensional [4].

Dalam pembuatan *game*, desain level merupakan salah satu faktor utama yang menentukan kualitas *gameplay* dari suatu *game*, tetapi dikarenakan setiap plot cerita dalam *game* harus didampingi oleh desain level yang sesuai maka dapat menyebabkan ketidakseimbangan antara plot cerita dengan desain level [5]. Jadi seringkali untuk mengimbangi plot yang ada, pembuatan desain level memerlukan sumber daya yang terlalu boros, namun jika desain level dapat dikembangkan secara algoritmik akan menghemat sumber daya waktu dan uang [6]. Maka dari itu dalam penelitian ini akan membuat desain level secara dinamis menggunakan *Procedural Content Generation* karena terbukti dapat memberikan pengalaman *game* yang dinamis [7]. Metode - metode dalam *Procedural Content Generation* yang dapat digunakan dalam pembuatan desain level seperti, *Search Based Procedural Content Generation* [8], *Occupancy Regulated Extension* [9], dan algoritme genetika [7].

Penelitian yang dilakukan oleh [7] menyimpulkan bahwa, penerapan *Procedural Content Generation* untuk perancangan level pada 2D *Endless Runner Game* menggunakan algoritme genetika dapat dipakai, tapi perancangan penilaian *fitness* yang digunakan sebagai parameter nilai dari level cukup kompleks dan butuh untuk diperbaiki. Dari penelitian [10] disimpulkan bahwa metode *quality-diversity* dapat digunakan sebagai strategi pencarian dalam *search-base procedural content generation* yang menghasilkan konten berkualitas dan beragam dalam *game*. Kemudian, penelitian tentang implementasi PCG [11] menggunakan metode *Occupancy Regulated Extension* dengan hasil pengujian *expressiveness range* diperoleh nilai 93% dari seluruh data. [12] menyimpulkan bahwa algoritme genetika dapat digunakan untuk menyusun *market layout* secara otomatis, dan dalam penggunaannya algoritme genetika dapat menghasilkan solusi optimal yang beragam, serta dapat bekerja dengan banyak jenis parameter. Penelitian oleh [13] dapat disimpulkan bahwa algoritme genetika dapat digunakan untuk desain level dalam *game* bergenre *platformer* tapi memiliki kelemahan dalam segi waktu karena *search space* yang besar. Penelitian oleh [14] menyimpulkan bahwa *search-based* PCG dapat digunakan untuk menciptakan lapangan balap dalam *game*, tanpa memiliki persimpangan dan lapangan tidak terbuka. Dari hasil kuesioner didapat bahwa lapangan balap hasil PCG dapat memberikan tingkat kesenangan yang cukup bagi pemain bahkan jika dibandingkan

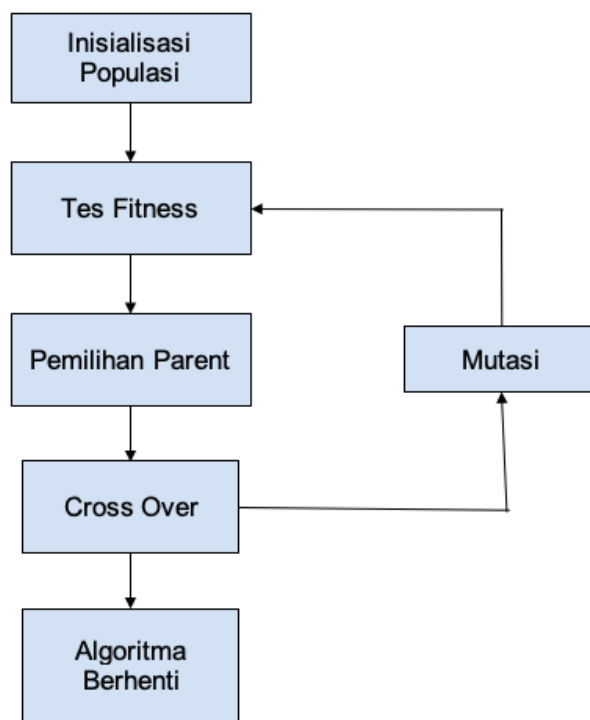
dengan lapangan balap yang dibuat secara manual. Penelitian oleh [15] menyimpulkan bahwa variasi konten yang dapat dibuat mencapai tak terhingga, dan menggunakan kombinasi dari metode generasi simpel dapat menciptakan hasil konstruksi level yang kompleks, menarik, bervariasi, dan mengejutkan.

Penelitian ini menggunakan algoritme genetika sebagai metode *Procedural Content Generation* dalam desain level dari *game* edukasi kebakaran hutan. Metode algoritme genetika digunakan karena dapat menghasilkan bentuk level yang optimal dari hasil iterasi dengan memilih kandidat terbaik [15]. Algoritma genetika merupakan algoritme yang meniru proses evolusi yang terjadi di alam, yakni dengan melakukan proses seleksi terhadap populasi yang kemudian disilangkan serta dilakukan mutasi terhadap kromosom untuk menghasilkan kromosom-kromosom baru yang lebih baik daripada sebelumnya [15]. Dari rumusan latar belakang diatas, penelitian ini akan merancang desain level dalam *game* edukasi kebakaran hutan secara dinamis menggunakan algoritme genetika.

## 2. METODE DAN BAHAN

Algoritma genetika digunakan sebagai metode desain level berupa maze hutan dalam *game* edukasi kebakaran hutan. Level yang terbentuk harus memiliki kriteria-kriteria seperti harus dapat diselesaikan, pohon-pohon dalam map tersebar secara merata, dan titik api sebagai awal dimulainya kebakaran berjumlah sedikit. Berikut penjelasan tahap-tahap yang dilakukan dalam penelitian ini.

Implementasi metode GA menggunakan model *Steady State Population*, yang berarti dalam setiap generasi baru akan dihasilkan dua kromosom baru dari hasil crossing kromosom generasi sebelumnya yang telah diseleksi berdasarkan nilai *fitness*. Model ini digunakan agar kromosom dengan nilai *fitness* terendah digantikan oleh kromosom baru hasil crossing dengan tujuan mendorong populasi menuju nilai *fitness* yang diinginkan. Tahapan umum metode ditunjukkan pada gambar 1 berikut.



Gambar 1. Tahapan algoritme genetika

Inisialisasi populasi menggunakan Random Initialization, yang berguna untuk menciptakan populasi awal berisi kromosom-kromosom yang bervariasi sehingga dapat mendapatkan hasil yang optimal dan tidak berhenti pada solusi dini yang kurang optimal. Populasi awal yang diciptakan dengan *Random Initialization* berjumlah 60 kromosom. Dalam penelitian ini gen direpresentasikan dengan bentuk integer. Gen berisi koordinat, panjang *block* pohon, apakah *block* pohon memiliki titik api atau tidak (1 jika berisi titik api, dan 0 jika tidak), apakah *block* pohon vertikal atau horizontal (1 jika vertikal, 0 jika horizontal).

Dalam penelitian ini gen direpresentasikan dengan bentuk integer. Gen berisi koordinat, panjang *block* pohon, apakah *block* pohon memiliki titik api atau tidak (1 jika berisi titik api, dan 0 jika tidak), apakah *block* pohon vertikal atau horizontal (1 jika vertikal, 0 jika horizontal). Dalam penelitian ini gen direpresentasikan dengan bentuk integer. Gen berisi koordinat, panjang *block* pohon, apakah *block* pohon memiliki titik api atau tidak (1 jika berisi titik api, dan 0 jika tidak), apakah *block* pohon vertikal atau horizontal (1 jika vertikal, 0 jika horizontal).

Tahap selanjutnya adalah *Tes Fitness*. Dalam tahap ini setiap kromosom akan diukur nilai *fitness*nya dengan empat fungsi *fitness* sebagai berikut.

### 1. Apakah level dapat diselesaikan?

Dalam fungsi ini dilakukan eksplorasi ke seluruh tile dalam map untuk mengetahui apakah terdapat tile yang seharusnya dapat dilewati tetapi tertutup

oleh pohon sehingga tidak bisa dijelajahi oleh pemain. Jika ada floor yang tidak bisa dilalui pemain maka diberi nilai 0,2, sebaliknya akan mendapat nilai 1. Fungsi ini ditunjukkan pada rumus 1 berikut.

$$f(n) = \begin{cases} n = 1, & trav < fCo \\ n = 0,2, & trav > fCo \end{cases} \quad (1)$$

Dengan n adalah nilai *fitness*, trav adalah jumlah floor yang sudah dilalui dan fCo adalah jumlah total floor.

2. Rasio jumlah vertikal dan horizontal dari *block* pohon.

Fungsi yang ditunjukkan pada rumus 2 digunakan untuk menjaga rasio jumlah vertikal dan horizontal tetap seimbang. Sehingga semakin kecil perbedaan jumlah pohon vertical dengan horizontal, semakin besar nilai *fitness*nya.

$$f(n) = \begin{cases} \frac{hCo}{vCo}, & vCo \geq hCo \\ \frac{vCo}{hCo}, & hCo < vCo \end{cases} \quad (2)$$

Nilai n adalah nilai *fitness*, hCo adalah jumlah *block* pohon horizontal dan vCo adalah jumlah *block* pohon vertical.

3. Persebaran pohon dalam map.

Dalam penghitungan yang ditunjukkan oleh rumus 3 dilakukan *looping* untuk setiap tile impassable, lalu dilakukan perhitungan lanjut terhadap tile impassable disekitarnya. Fungsi ini bertujuan agar *block* pohon tersebar secara merata dalam map.

$$n = 1 - \left(\frac{x}{imCo}\right) \quad (3)$$

Dimana n adalah nilai *fitness*, x menunjukkan jumlah pohon yang tergolong padat, dan imCo adalah jumlah pohon dalam map.

4. Jumlah titik api dalam map.

Fungsi dalam rumus 4 ini bertujuan untuk menjaga jumlah titik api dalam map agar tidak terlalu banyak karena akan menyebabkan level yang tidak dapat diselesaikan.

$$f(n) = \begin{cases} n = 1, & z \leq 5 \\ n = \frac{5}{z}, & z > 5 \end{cases} \quad (4)$$

Pada rumus 4 di atas, n adalah nilai *fitness* dan z adalah jumlah titik api. Setelah dilakukan empat fungsi evaluasi di atas, maka hasilnya adalah sebagai berikut.

Tabel 1. Bobot nilai evaluasi

Evaluasi	Bobot
1	0,4
2	0,1
3	0,1
4	0,4

Setelah Tes *Fitness* selanjutnya dilakukan pemilihan *parent*. Dalam penelitian ini digunakan Roulette Wheel Selection sebagai metode pemilihan *parent*. Dimana setiap *parent* memiliki kemungkinan terpilih yang berbeda - beda berdasarkan nilai *fitness*. Cara ini dipilih agar *parent* dengan *fitness* yang tinggi memiliki kemungkinan yang lebih untuk dipilih dalam proses *crossover* sehingga dapat menghasilkan kromosom yang lebih berkualitas. Untuk tahap *crossover*, Digunakan uniform *crossover* yang bekerja dengan memilih secara merata gen yang akan ditukar agar kromosom hasil *crossover* memiliki gabungan kromosom yang merata dari *parent*. Kemudian Mutasi yang digunakan dalam penelitian ini adalah random resetting agar tidak mendapatkan solusi dini yang kurang optimal atau *premature convergence* karena dalam mutasi ini kromosom dipilih secara random untuk dirubah. Algoritma berhenti jika populasi sudah tidak menghasilkan kromosom yang jauh berbeda dari sebelumnya.

### 3. HASIL DAN PEMBAHASAN

Pada tahap ini algoritme genetika akan diimplementasikan untuk menghasilkan blueprint berupa array yang berisi informasi genetik yang akan digunakan untuk membuat lingkungan *game* edukasi kebakaran hutan dalam *engine* unity. Contoh informasi genetik dapat dilihat pada gambar 2 dibawah ini.

0	2	0	0	0	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
2	2	1	1	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	0	1	1	0	0	2	0	0	0	0	2	1	1	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0	1	1	2	1	0	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	2	1	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	2	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0
0	0	2	0	1	0	0	0	0	0	2	1	0	2	1	0	1	0	1	0
0	0	1	0	2	0	0	0	1	0	1	1	0	0	0	0	1	0	1	0

Gambar 2. Informasi genetik

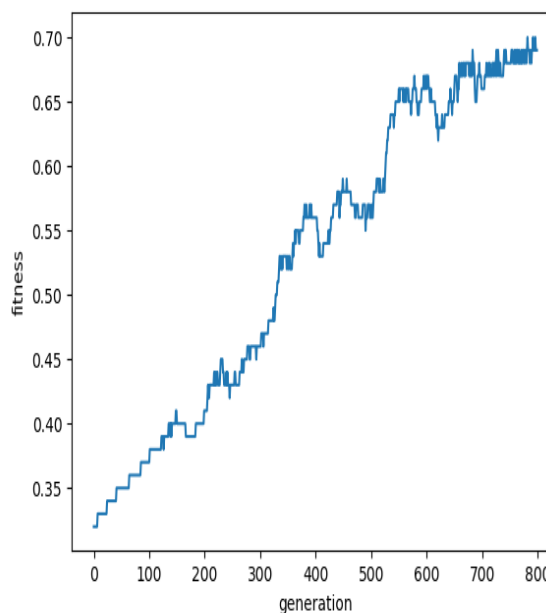
Pada Gambar 1 di atas, [0] merepresentasikan floor, [1] merupakan representasi pohon dan [2] adalah representasi titik api. Dari pengaturan populasi dan parameter *fitness* evaluation yang sudah ditentukan sebelumnya akan digunakan untuk menghasilkan individu dengan nilai *fitness* yang tinggi.

Inisialisasi populasi dalam penelitian ini menggunakan Random Initialization, yang digunakan untuk membuat populasi awal yang berisi dengan kromosom-kromosom yang bermacam-macam variasi sehingga menghasilkan produk yang optimal dan tidak berhenti pada solusi awal yang kurang optimal. Populasi awal yang dihasilkan dengan Random Initialization bertotal 60. Angka ini dipilih dari hasil eksperimen dengan mencoba berbagai angka populasi dan didapat bahwa 60 dapat menghasilkan kandidat yang optimal. Kemudian populasi awal yang berjumlah 60 individu dinilai dengan *fitness* evaluation yang sudah ditentukan sebelumnya. Dari hasil penilaian tersebut akan dipilih 2 individu dengan metode Roulette Wheel Selection agar individu dengan nilai *fitness* tinggi memiliki kemungkinan lebih untuk dipilih sehingga dapat menghasilkan individu baru yang lebih optimal. Lalu 2 individu dari populasi yang dipilih sebagai *parent* akan disilangkan informasi genetiknya. Persilangan menggunakan uniform *crossover* yang bekerja dengan memilih secara merata gen yang akan ditukar agar kromosom hasil *crossover* memiliki gabungan kromosom yang merata dari *parent*. Setiap individu dalam populasi memiliki kemungkinan untuk bermutasi agar menjaga variasi informasi genetik dalam populasi, sehingga mengurangi kemungkinan tiba pada solusi dini yang kurang optimal. Mutasi yang digunakan yaitu random resetting yang mengubah dna individu secara acak.

Tahap evaluasi, *crossover*, pemilihan *parent*, dan mutasi akan terus dilakukan hingga generasi ke-800 karena dalam percobaan yang sudah dilakukan, nilai *fitness* rata-rata dari populasi generasi diatas 800 tidak berubah secara signifikan karena setiap individu memiliki informasi genetik yang serupa. Maka dari itu setelah mencapai generasi ke-800 algoritme ini dihentikan dan diambil individu dengan nilai *fitness* tertinggi sebagai solusi optimal.

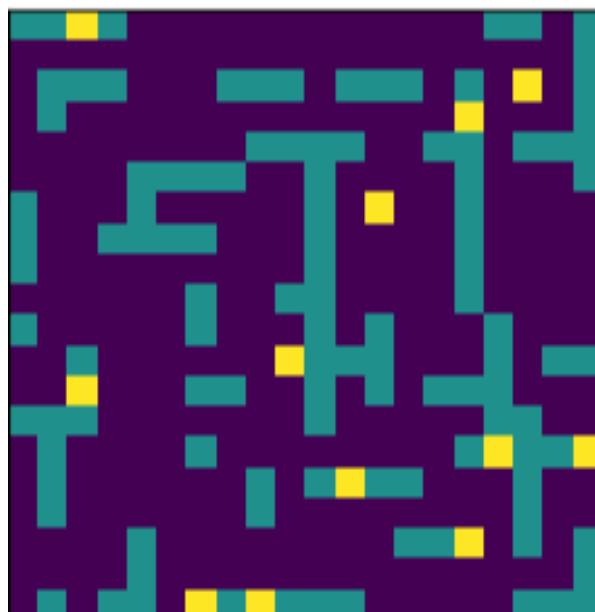
Dari penelitian [10], *fitness* function didesain untuk menciptakan lingkungan *game* yang bervariasi serta dapat diselesaikan pemain sehingga pemain tidak bosan dalam memainkannya, dalam penelitian ini ditambahkan 1 *fitness* function yang berfungsi untuk mengontrol jumlah titik api agar tidak terlalu banyak sehingga dapat diselesaikan oleh pemain. Karena jika terdapat terlalu banyak titik api pada

awal *game player* tidak memiliki waktu yang cukup untuk memadamkan api. Jadi dengan adanya tambahan *fitness* function ini rata-rata jumlah titik api dari generasi ke generasi menurun sehingga solusi yang didapat lebih optimal untuk *game* ini. Perkembangan rata-rata nilai evaluasi sampai 800 generasi ditunjukkan pada gambar 3 berikut.



Gambar 3. Perkembangan rata-rata nilai evaluasi

Pada gambar 4 berikut menampilkan contoh bentuk sederhana lingkungan *game* dari individu dengan nilai *fitness* tertinggi, yang merupakan hasil algoritme genetika generasi ke-800.

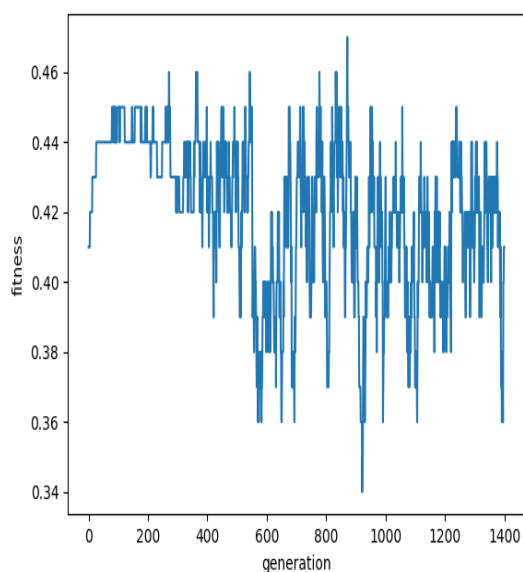


Gambar 4. Hasil lingkungan *game* yang terbentuk setelah 800 generasi

Pada hasil lingkungan *game* yang terbentuk pada

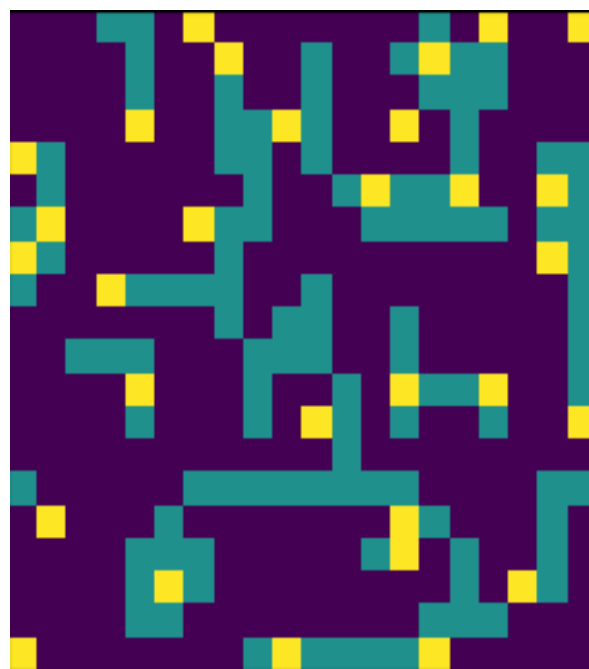
gambar 4, warna biru melambangkan pohon, ungu melambangkan floor, dan titik api direpresentasikan dengan warna kuning.

Dalam proses eksperimen awal, penelitian ini menggunakan susunan gen dalam kromosom yang berbeda. Susunan gen pada awalnya berbentuk tabel 25x25 yang setiap kotaknya mengandung informasi tentang posisi berupa koordinat (x, y) serta isi dari kotak tersebut (pohon atau floor). Dengan susunan seperti ini kemudian dilakukan *fitness* evaluation sehingga menghasilkan perkembangan nilai *fitness* rata-rata setelah 1400 generasi yang ditunjukkan pada gambar 5 dibawah ini.



Gambar 5. Perkembangan nilai *fitness* rata-rata setelah 1400 generasi

Pada generasi ke-1400 individu dengan nilai *fitness* tertinggi memiliki bentuk lingkungan *game* sederhana seperti ini. Pada gambar 6, dapat dilihat bahwa masih terdapat jalan-jalan buntu yang tidak dapat diakses pemain sehingga *game* tidak dapat diselesaikan.



Gambar 6. Hasil desain level *game* setelah 1400 generasi

Hasil yang dicapai tidak mampu menemukan global optima dan malah terjadi penurunan nilai *fitness* karena setiap kotak dalam tabel kromosom 25x25 berdiri sendiri sehingga tidak ada korelasi antar kotak yang dapat merepresentasikan solusi. Maka dari itu susunan gen diubah menjadi tabel berukuran 68x4 yang setiap baris melambangkan 1 deretan pohon dan setiap kolom berisi informasi koordinat, panjang *block* pohon, apakah *block* pohon memiliki titik api atau tidak (1 jika berisi titik api, dan 0 jika tidak), apakah *block* pohon vertikal atau horizontal (1 jika vertikal, 0 jika horizontal).

Dengan susunan seperti ini dihasilkan perkembangan nilai *fitness* rata-rata populasi seperti pada gambar 4.2, tetapi untuk mengoptimalkan hasil yang didapat ukuran lingkungan *game* diperkecil dari 25x25 menjadi 20x20 agar meminimalisir search space.

Setelah proses algoritme genetika selesai, maka hasil yang diperoleh yaitu lingkungan *game* optimal dalam bentuk array seperti pada gambar 1 dan diimplementasikan menggunakan *game engine* Unity. Proses algoritme ini membutuhkan waktu yang terlalu lama bagi pemain yakni sekitar 5 menit, maka proses algoritme dipisah dari *engine* unity dengan cara menyimpan terlebih dahulu 10 blueprint hasil algoritme kedalam variabel Unity, sehingga setiap kali pemain memulai *game* tidak perlu menjalankan algoritme genetika terus menerus. Jadi ketika pemain memulai *game* akan dipilih 1 dari 10 lingkungan *game* yang ada secara random. Setelah semua proses di atas dijalankan maka lingkungan *game* yang dihasilkan ditunjukkan pada gambar 7 sebagai berikut.



Gambar 7. Implementasi desain level *game* pada *game engine*

Setiap pohon, tanah, dan titik api dalam gambar 6 dimunculkan berdasarkan array dna hasil operasi algoritme genetika. Sebagai contoh:

1. Jika dalam array dna pada koordinat (4,5) terdapat nilai 1 maka dalam lingkungan *game* akan muncul pohon pada koordinat yang sama.
2. Jika dalam array dna pada koordinat (6,4) terdapat nilai 0 maka dalam lingkungan *game* akan muncul tanah pada koordinat yang sama.
3. Jika dalam array dna pada koordinat (3,11) terdapat nilai 2 maka dalam lingkungan *game* akan muncul titik api pada koordinat yang sama.

#### 4. KESIMPULAN

Dari penelitian ini dapat ditarik kesimpulan sebagai berikut. Berdasarkan hasil yang didapat, algoritme genetika berhasil untuk menciptakan desain level secara dinamis dengan memilih kandidat optimal dari hasil iterasi generasi ke generasi. Proses pembuatan lingkungan *game* menggunakan Algoritma Genetika memiliki faktor -faktor penting, salah satunya yaitu representasi genetika, karena dari penelitian yang telah dilakukan, dna dari individu harus dapat merepresentasikan solusi dari masalah yang ada agar meminimalisir search space. Faktor yang kedua yaitu desain *Fitness Evaluation* harus dapat mendorong populasi menuju solusi yang optimal karena jika desain yang diberikan kurang komprehensif maka dapat menyebabkan nilai *fitness* rata-rata stagnan atau bahkan menurun secara drastis.

#### DAFTAR PUSTAKA

- [1] Tacconi, L., "Kebakaran hutan di Indonesia: penyebab, biaya dan implikasi kebijakan," *Kebakaran hutan di Indones. penyebab, biaya dan implikasi Kebijak.*, CIFOR, no. 38(i), 2003, doi: 10.17528/cifor/001200.
- [2] D. Hunawan, "Menyelesaikan Kebakaran Hutan dan Lahan (KARHUTLA) di Indonesia melalui ' Jalan Pantas ' atau ' Jalan Pintas '?," *Law Research Review Quarterly*, vol. 2, no. 2, pp. 277-292, 2016, doi: 10.15294/snh.v2i01.21312.
- [3] C. B. S. Bjørner, Thomas; Hansen, "Designing an Educational Game: Design Principles from a Holistic Perspective," *The International Journal of Learning Annual Review*, vol. 17, no. 10 pp. 279-290. 2011, doi: 10.18848/1447-9494/CGP/v17i10/47275
- [4] A. V. Vitianingsih, "Game Edukasi Sebagai Media Pembelajaran Pendidikan Anak Usia Dini," *Jurnal Inform Vol. 1, No. 1, ISSN 2502-3470*, 2016.
- [5] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, "Toward supporting stories with procedurally generated game worlds," *2011 IEEE Conf. Comput. Intell. Games, CIG 2011*, pp. 297-304, 2011, doi: 10.1109/CIG.2011.6032020.
- [6] N. Shaker, J. Togelius, and M. J. Nelson, "Why use procedural content generation" in *Procedural Content generation in Games: A Textbook and an Overview of Current Research*. 2016.
- [7] M. A. Muslim, E. Muhammad, A. Jonemaro, and M. A. Akbar, "Penerapan Procedural Content Generation untuk Perancangan Level pada 2D Endless Runner Game menggunakan Genetic Algorithm," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 5, pp. 4406-4414, 2019.
- [8] H. A. Prasetya and N. U. Maulidevi, "Search-based procedural content generation for race tracks in video games," *International Journal on Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 774-786, 2016, doi: 10.15676/ijeei.2016.8.4.6.
- [9] A. Setyamurti, W. S. Wardhono, and T. Afirianto, "Implementasi Procedural Generation untuk Membangun Level Tactical RPG dengan menggunakan Metode *Occupancy Regulated Extension*," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 8, pp. 2416-2420, 2018.
- [10] Gravina, D., Khalifa, A., Liapis, A., Togelius, J., & Yannakakis, G. N. (2019). Procedural content generation through quality diversity. *IEEE Conference on Computational Intelligence and Games, CIG*, 2019-Augus. <https://doi.org/10.1109/CIG.2019.8848053>
- [11] Mourato, Fausto., P. Manuel, Birra, Fernando., Nur U. Automatic level generation for platform

- videogames using genetic algorithms. *ACE '11: Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, November 2011.
- [12] Dalgic, H. O., Bostanci, E., & Guzel, M. S. (2017). Genetic Algorithm Based Floor Planning System. Retrieved from <http://arxiv.org/abs/1704.06016>
- [13] Bjørner, Classon, J. "Procedural Generation of Levels with Controllable Difficulty for a Platform Game Using a Genetic Algorithm". Master Thesis. Linköping University. Linköping. 2016.
- [14] Teinemaa, T. A Procedural Approach for Infinite Deterministic 2D Grid-Based World Generation. *Foundations of Digital Games Conference 2015*, June 2015.
- [15] A. B. Safak, E. Bostanci, and A. E. Soylucicek, "Automated Maze Generation for Ms. Pac-Man Using Genetic Algorithms," *International Journal of Machine Learning and Computing*, vol. 6, no. 4, pp. 226–230, 2016, doi: 10.18178/ijmlc.2016.6.4.602.