

Optimasi Pembangkitan Labirin Menggunakan Algoritma Genetika dengan Evaluasi Berbasis Simulasi DFS dan Delapan Kombinasi Evolusi

Bagus Cipta Pratama (23/561539/PA/22097), David Neilleen Irvinne (23/517639/PA/22199),
Syafran Abdillah Erdin (23/521752/PA/22444), Muhammad Dhiwaul Akbar (23/523237/PA/22513)
Departemen Ilmu Komputer dan Elektronika
Universitas Gadjah Mada, Yogyakarta, Indonesia

Abstract—Desain labirin merupakan permasalahan optimasi kombinatorial yang menuntut struktur yang valid, solvable, dan memiliki tingkat kesulitan yang terkontrol. Penelitian ini mengembangkan sistem pembangkit labirin berbasis *Genetic Algorithm* (GA) menggunakan representasi empat-bit dinding per sel serta mekanisme *normalize* dan *repair* untuk menjaga konsistensi struktur. Evaluasi dilakukan melalui satu fungsi fitness berbasis simulasi *Depth-First Search* (DFS) yang meniru pola eksplorasi manusia untuk mengukur tingkat kesulitan navigasi. Delapan konfigurasi GA diuji sebagai hasil kombinasi dari dua metode seleksi, dua operator crossover, dan dua kondisi evolusi (dengan atau tanpa elitisme). Seluruh konfigurasi mencapai nilai fitness terbaik sebesar 7.08717×10^{-4} , dengan elitisme terbukti meningkatkan stabilitas dan konsistensi evolusi. Hasil menunjukkan bahwa GA mampu menghasilkan labirin yang valid, solvable, serta memiliki karakteristik struktural yang dapat dikendalikan.

Index Terms—DFS, Genetic Algorithm, Maze Generation, Optimization

I. PENDAHULUAN

Labirin merupakan struktur graf bergrid yang membutuhkan kemampuan navigasi dalam ruang dua dimensi. Tantangan utama dalam mendesain labirin meliputi pemastian bahwa struktur yang dihasilkan valid, memiliki jalur solusi dari titik awal ke titik akhir, serta menawarkan tingkat kesulitan yang sesuai kebutuhan. Mendesain labirin secara manual dapat memakan waktu dan menghasilkan variasi yang terbatas. Algoritma Genetika (GA) menawarkan pendekatan otomatis yang mampu mengeksplorasi ruang solusi yang sangat besar dan menghasilkan konfigurasi labirin yang lebih beragam.

Pada penelitian ini, kami mengembangkan generator labirin otomatis berbasis GA dengan populasi berjumlah 64 individu yang berevolusi selama 32 generasi. Setiap individu merupakan satu labirin lengkap berukuran 12×12 . Evaluasi kualitas individu dilakukan menggunakan satu fungsi *fitness* berbasis simulasi *Depth-First Search* (DFS), yang meniru perilaku eksplorasi manusia dalam menavigasi labirin dan mengukur tingkat kesulitan berdasarkan jumlah langkah yang ditempuh. Sementara itu, algoritma *Breadth-First Search* (BFS) digunakan hanya untuk memastikan solvabilitas labirin, bukan sebagai komponen penilaian fitness.

Penelitian ini mengevaluasi delapan kombinasi mekanisme GA—dua metode seleksi (roulette dan tournament), dua op-

erator crossover (uniform biased dan simple arithmetic), serta dua konfigurasi evolusi (dengan dan tanpa elitisme)—untuk menganalisis pengaruh masing-masing komponen terhadap stabilitas evolusi dan kualitas labirin yang dihasilkan.

II. METODOLOGI

Metodologi penelitian ini dirancang untuk mengembangkan labirin melalui pendekatan evolusioner yang stabil, konsisten, dan dapat menghasilkan struktur yang memiliki tingkat kesulitan terkontrol. Bagian ini menjelaskan secara ilmiah representasi labirin, proses validasi dan koreksi struktur, strategi evaluasi, serta operator genetika yang digunakan dalam proses evolusi.

A. Representasi Labirin

Labirin dimodelkan sebagai kisi dua dimensi berukuran $N \times N$. Setiap sel memiliki empat sisi (kiri, atas, kanan, bawah) yang masing-masing dapat berupa dinding atau jalur. Representasi sel dilakukan menggunakan empat variabel biner:

$$c_{ij} = (w_L, w_T, w_R, w_B), \quad w_k \in \{0, 1\}.$$

Dengan demikian, labirin secara keseluruhan dapat dipandang sebagai vektor besar berdimensi:

$$X = (c_{11}, c_{12}, \dots, c_{NN}),$$

yang menghasilkan ruang solusi eksponensial. Representasi ini memastikan setiap modifikasi lokal tetap konsisten dengan struktur spasial labirin.

B. Pembentukan Populasi Awal

Populasi awal dibentuk melalui beberapa tahapan terkontrol untuk memastikan setiap individu merupakan labirin valid yang dapat diselesaikan. Tahapannya:

- 1) Sampling acak: seluruh sel diberi konfigurasi dinding acak.
- 2) Penyelarasan struktural: antar sel bertetangga diberi koreksi sehingga dinding bersifat konsisten secara *encoding* yang disepakati.
- 3) Validasi rute: dilakukan pencarian jalur untuk memastikan adanya jalur dari titik awal menuju titik akhir.

Tahapan ini menjamin bahwa evolusi dimulai dari populasi yang valid dan tidak mengandung kandidat yang mustahil diselesaikan.

C. Normalisasi Struktur

Untuk menjaga integritas struktur, diterapkan proses normalisasi yang menyelaraskan dinding antar sel. Jika dua sel saling bertetangga, kondisi sisi yang sama harus bersifat simetris. Untuk setiap pasang sel horizontal (i, j) dan $(i, j+1)$, konsistensi dinding dinyatakan sebagai:

$$w_R(i, j) = w_L(i, j+1),$$

dan untuk pasang sel vertikal (i, j) dan $(i+1, j)$ berlaku:

$$w_B(i, j) = w_T(i+1, j).$$

Di mana $w_R, w_L, w_T, w_B \in \{0, 1\}$ merepresentasikan ada atau tidaknya dinding kanan, kiri, atas, dan bawah. Proses ini memastikan representasi grid konsisten dan bebas konflik geometris, sehingga setiap koneksi antar sel dibentuk secara valid.

D. Perbaikan Sel Tidak Valid

Beberapa konfigurasi lokal, khususnya sel yang seluruh sisinya tertutup, dapat membentuk ruang isolasi. Kondisi sel tertutup penuh direpresentasikan sebagai:

$$w_L(i, j) + w_T(i, j) + w_R(i, j) + w_B(i, j) = 4.$$

Untuk mencegah isolasi, mekanisme perbaikan memilih salah satu sisi untuk dibuka sehingga:

$$\sum_{k \in \{L, T, R, B\}} w_k(i, j) \leq 3.$$

Dengan membuka minimal satu sisi, seluruh area dalam labirin tetap terhubung dan dapat diakses pada proses pencarian jalur.

E. Analisis Keterselesaian

Sebelum suatu labirin dievaluasi dengan fungsi fitness berbasis DFS, sistem terlebih dahulu memastikan bahwa labirin tersebut memiliki jalur solusi yang valid. Proses verifikasi ini dilakukan menggunakan *Breadth-First Search* (BFS), yang berfungsi sebagai uji keterselesaian (*solvability check*), bukan sebagai komponen penilaian fitness.

Labirin dimodelkan sebagai graf tidak berarah (V, E) , di mana setiap sel merupakan simpul dan sebuah sisi $(u, v) \in E$ ada jika dan hanya jika tidak terdapat dinding pemisah antara kedua sel tersebut. BFS kemudian digunakan untuk menentukan apakah terdapat jalur dari titik awal $s = (0, 0)$ ke titik tujuan $t = (N-1, N-1)$.

Secara formal, BFS mengembalikan jarak minimum:

$$d(s, t) = \text{BFS}(s, t),$$

dan jika tidak ditemukan jalur, maka:

$$d(s, t) = \infty.$$

Apabila $d(s, t) = \infty$, labirin dikategorikan *unsolvable* dan dikeluarkan dari proses evolusi. Dengan demikian, BFS hanya berfungsi memastikan bahwa setiap individu yang dievaluasi oleh fungsi fitness DFS merupakan labirin yang layak dan dapat diselesaikan, tanpa memberikan kontribusi langsung pada nilai fitness itu sendiri.

F. Evaluasi Fitness: Kesulitan Mirip Manusia (Simulasi DFS)

Evaluasi fitness pada penelitian ini berbasis simulasi *Depth-First Search* (DFS) acak yang meniru perilaku manusia ketika menjelajahi labirin. DFS tidak digunakan untuk mencari jalur optimal, melainkan untuk memodelkan eksplorasi yang melibatkan banyak backtracking, sehingga mencerminkan tingkat kebingungan navigasi yang lebih realistis.

Setiap individu dievaluasi melalui sejumlah simulasi DFS. Jika s_1, s_2, \dots, s_R merupakan jumlah langkah yang ditempuh dari masing-masing simulasi, maka estimasi langkah rata-rata dihitung sebagai:

$$\bar{s} = \frac{1}{R} \sum_{r=1}^R s_r.$$

Nilai \bar{s} ini dianggap sebagai *cost value*, yaitu indikasi tingkat kesulitan labirin dari perspektif eksplorasi manusia. Untuk mengonversi *cost value* menjadi fitness, penelitian ini mendefinisikan rentang tingkat kesulitan target.

Batas bawah menggambarkan jalur lurus tanpa percabangan:

$$\text{lower bound} = (N + N - 1),$$

sedangkan batas atas adalah estimasi kondisi dengan banyak backtracking:

$$\text{upper bound} \approx 1.8 N^2.$$

Target tingkat kesulitan (kesulitan menengah) didefinisikan sebagai titik tengah rentang tersebut:

$$\text{mid} = \frac{\text{lower bound} + \text{upper bound}}{2}.$$

Fitness dihitung berdasarkan kedekatan *cost value* dengan nilai target:

$$F = \frac{|\bar{s} - \text{mid}|}{\text{mid}}.$$

Semakin kecil nilai F , semakin baik kualitas labirin, karena jumlah langkah yang dihasilkan DFS mendekati tingkat kesulitan menengah yang diinginkan.

Dalam penelitian ini, hanya satu fungsi fitness digunakan pada setiap eksperimen, yaitu DFS-based fitness. Tidak terdapat penggabungan fungsi fitness ataupun pendekatan multi-objektif agar dinamika evolusi dapat diamati secara jelas dan independen.

G. Mekanisme Seleksi Induk

Penelitian ini menggunakan dua metode seleksi:

- 1) Roulette Selection : peluang setiap individu untuk dipilih sebanding dengan kebalikan nilai fitness-nya. Peluang seleksi individu ke- i dirumuskan sebagai:

$$P(i) = \frac{1/F_i}{\sum_j 1/F_j}.$$

- 2) Tournament Selection : sejumlah individu dipilih secara acak sebagai kandidat, kemudian individu dengan fitness terbaik di antara kelompok tersebut terpilih sebagai induk.

Kedua mekanisme ini memberikan keseimbangan antara eksplorasi dan eksploitasi.

H. Operator Crossover

Operator crossover bertanggung jawab mengombinasikan informasi genetik dari dua induk. Dua pendekatan digunakan:

- Uniform Biased Crossover: setiap komponen anak dipilih dari salah satu induk berdasarkan probabilitas tertentu, memungkinkan dominasi salah satu induk.
- Arithmetic Crossover: nilai genetik anak merupakan kombinasi linier dari dua induk, memberikan transisi yang lebih halus di ruang solusi.

Pendekatan ini memungkinkan pembentukan struktur labirin baru dengan karakteristik campuran.

I. Mekanisme Mutasi

Mutasi mempertahankan keragaman genetik populasi. Terdapat dua tingkat mutasi:

- Mutasi kecil: perubahan lokal pada satu sisi dinding sel.
- Mutasi besar: penggantian konfigurasi dinding sel secara acak.

Mutasi kecil menjaga stabilitas struktur, sedangkan mutasi besar memungkinkan eksplorasi ruang solusi yang lebih luas.

J. Strategi Elitisme

Elitisme memastikan bahwa sejumlah individu dengan kualitas terbaik pada suatu generasi dipertahankan tanpa perubahan pada generasi berikutnya. Hal ini dilakukan dengan:

- 1) mengidentifikasi individu terbaik berdasarkan nilai evaluasi,
- 2) menyisipkannya langsung ke populasi generasi berikutnya,
- 3) memberi ruang bagi keturunan baru untuk mengisi posisi lainnya.

Strategi ini mengurangi risiko hilangnya solusi berkualitas akibat proses evolusi acak.

K. Siklus Evolusi Keseluruhan

Siklus evolusi keseluruhan mencakup tahapan berikut:

- 1) membentuk populasi awal yang valid,
- 2) mengevaluasi seluruh individu,
- 3) memilih induk menggunakan mekanisme seleksi,
- 4) menghasilkan keturunan melalui crossover,
- 5) menerapkan mutasi untuk menjaga keragaman,
- 6) menyusun populasi baru berisi keturunan,
- 7) menerapkan elitisme,
- 8) mengulang proses hingga mencapai jumlah generasi yang ditentukan.

Rangkaian tahap ini memastikan perkembangan kualitas labirin secara bertahap dan konsisten melalui proses evolusioner.

III. EKSPERIMEN

Delapan konfigurasi percobaan dirancang untuk mengevaluasi pengaruh kombinasi mekanisme seleksi, operator crossover, dan penerapan elitisme terhadap kualitas labirin yang dihasilkan. Seluruh percobaan menggunakan ukuran populasi 64 individu dan dijalankan selama 32 generasi, dengan parameter implementasi sebagai berikut:

- Probabilitas mutasi: 0.02 untuk mutasi kecil (bit-flip) dan 0.10 untuk mutasi besar.
- Ukuran turnamen: 32 individu ketika menggunakan *tournament selection*.
- Simulasi DFS: setiap evaluasi DFS dijalankan sebanyak 8 kali untuk menghitung nilai *average steps taken*.
- Nilai Elitism: apabila *elitism* digunakan, dari 32 individu di populasi, 2 individu terbaik akan dipilih untuk lanjut ke generasi selanjutnya menggantikan 2 individu terburuk.
- Ukuran grid: seluruh eksperimen menggunakan labirin berukuran 12×12 .

Setiap konfigurasi dievaluasi menggunakan tiga metrik utama:

- 1) Best Fitness Value: nilai fitness terbaik pada generasi terakhir.
- 2) Number of Different Shortest Paths: jumlah jalur terpendek berbeda pada solusi terbaik.
- 3) Average Steps Taken: rerata langkah DFS yang menggambarkan tingkat kebingungan navigasi manusia.

A. Hasil Eksperimen

Tabel berikut merangkum hasil akhir dari delapan konfigurasi percobaan dengan menampilkan satu nilai *best fitness* yang berlaku untuk seluruh konfigurasi.

TABLE I
HASIL AKHIR DELAPAN KONFIGURASI EKSPERIMEN

No.	Selection	Crossover	Elitism	Best Fitness
1	Roulette	Uniform	Yes	0.000708717
2	Roulette	Uniform	No	
3	Roulette	Arithmetic	Yes	
4	Roulette	Arithmetic	No	
5	Tournament	Uniform	Yes	
6	Tournament	Uniform	No	
7	Tournament	Arithmetic	Yes	
8	Tournament	Arithmetic	No	

TABLE II
PARAMETER STRUKTURAL LABIRIN TERBAIK BERDASARKAN HASIL PERCOBAAN

No.	Different Shortest Paths	Avg Steps Taken
1	2-4	141
2	1-6	140.25-143
3	1	141
4	2-709	139.5-142.5
5	6-12	141
6	4-64	138.75-142
7	2	141
8	11-273	140-142.75

B. Analisis

Semua konfigurasi berhasil mencapai nilai fitness terbaik yang sama, yaitu 7.08717×10^{-4} , namun parameter struktural yang dihasilkan menunjukkan perbedaan yang signifikan. Konfigurasi tanpa elitisme memiliki variansi yang jauh lebih besar, terutama pada kombinasi “Roulette + Arithmetic + No Elitism” yang dapat menghasilkan hingga ratusan jalur terpendek. Hal ini terjadi karena ketiadaan elitisme memungkinkan solusi terbaik hilang akibat crossover dan mutasi, sehingga struktur labirin menjadi tidak stabil.

Sebaliknya, konfigurasi dengan elitisme jauh lebih konsisten, dengan jumlah jalur terpendek dan rata-rata langkah DFS yang relatif stabil di seluruh generasi. Ini menunjukkan bahwa elitisme efektif menjaga solusi terbaik dan mencegah populasi bergeser ke bentuk labirin yang terlalu bercabang.

Dari sisi metode seleksi, *tournament selection* menunjukkan perilaku yang lebih terkontrol dibandingkan *roulette selection*, terutama ketika elitisme tidak digunakan. Sensitivitas roulette terhadap distribusi fitness populasi menyebabkan fluktuasi yang lebih besar. Secara keseluruhan, elitisme muncul sebagai faktor stabilisasi utama, sementara tournament selection memberikan tekanan seleksi yang lebih konsisten.

IV. HASIL DAN PEMBAHASAN

Eksperimen dilakukan pada delapan konfigurasi Algoritma Genetika (GA) yang mengombinasikan dua metode seleksi (roulette dan tournament), dua jenis crossover (uniform dan arithmetic), serta opsi elitisme. Seluruh nilai fitness dihitung melalui simulasi DFS sehingga mencerminkan tingkat kesulitan labirin berdasarkan panjang rute eksplorasi.

Secara umum, seluruh konfigurasi menunjukkan penurunan fitness dari generasi ke generasi, sehingga dapat disimpulkan

bahwa GA berhasil mengarahkan populasi menuju labirin dengan tingkat kesulitan yang diinginkan.

A. Best Fitness: Stabilitas Solusi Terbaik

Gambar 1 memperlihatkan bahwa pada konfigurasi tanpa elitisme, nilai *best fitness* mengalami fluktuasi signifikan. Meskipun sesekali muncul solusi sangat baik, solusi tersebut sering hilang di generasi berikutnya akibat crossover atau mutasi. Fenomena ini mencerminkan karakteristik bahwa tanpa elitisme, individu terbaik tidak memiliki perlindungan.

Sebaliknya, pada konfigurasi dengan elitisme (Gambar 2), nilai *best fitness* turun lebih cepat dan kemudian bertahan stabil di sekitar nilai 0.0007. Elitisme memastikan bahwa solusi terbaik tidak hilang sehingga evolusi bergerak lebih terarah dan tidak mengalami regresi kualitas.

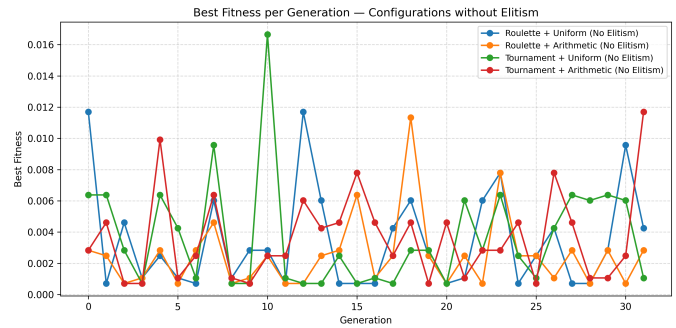


Fig. 1. Best fitness per generasi tanpa elitisme.

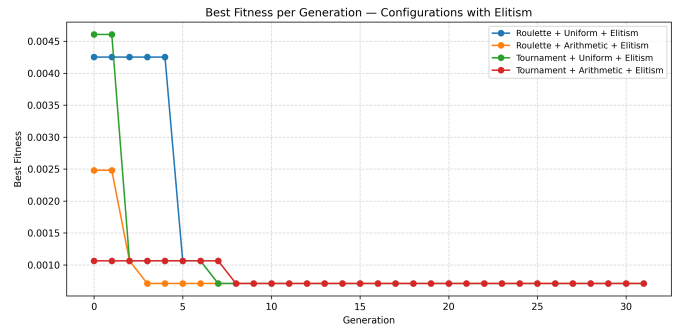


Fig. 2. Best fitness per generasi dengan elitisme.

B. Rata-rata Fitness: Mengapa Elitisme Lebih Fluktuatif?

Grafik rata-rata fitness menunjukkan tren yang menarik. Pada konfigurasi tanpa elitisme (Gambar 3), kurva terlihat lebih halus dan fluktuasinya relatif kecil. Hal ini terjadi karena seluruh individu dapat tergantikan setiap generasi, sehingga distribusi fitness populasi menjadi lebih merata dan perubahan antar generasi lebih cair.

Sebaliknya, konfigurasi dengan elitisme (Gambar 4) justru menunjukkan osilasi yang lebih besar. Meskipun nilai terbaik stabil, elitisme mempercepat homogenisasi populasi sehingga sedikit variasi genetik menghasilkan perubahan besar pada sejumlah individu non-elit. Ketika mutasi atau kombinasi genetik menghasilkan individu yang jauh lebih buruk, nilai

rata-rata fitness langsung terdorong naik, menghasilkan pola fluktuasi yang lebih tajam.

Dengan demikian, elitisme membuat *best fitness* stabil, tetapi memunculkan fluktuasi pada *average fitness* karena adanya kesenjangan kualitas antara elit dan non-elit.

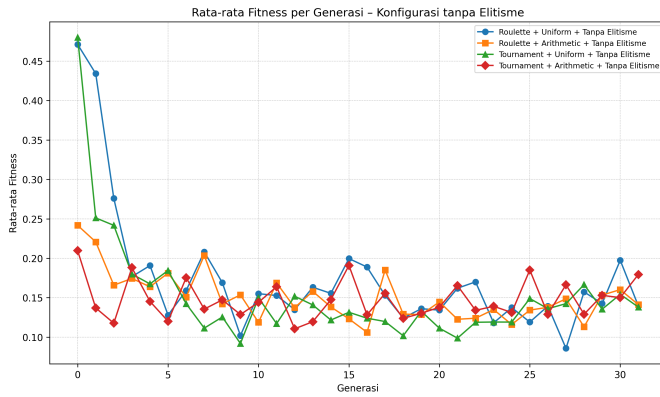


Fig. 3. Rata-rata fitness per generasi tanpa elitisme.

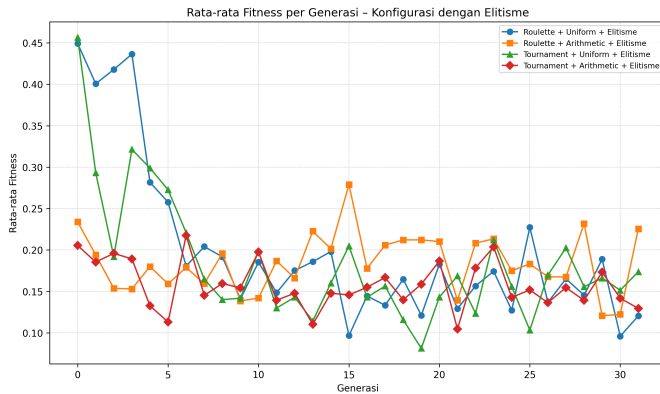


Fig. 4. Rata-rata fitness per generasi dengan elitisme.

C. Analisis Seleksi dan Crossover

Tournament selection terlihat lebih stabil dibanding roulette pada seluruh grafik. Tournament selalu memilih kandidat terbaik dari subset kecil, sehingga tekanan seleksi konsisten dan tidak terlalu dipengaruhi distribusi fitness global. Sementara itu, roulette selection lebih peka terhadap perubahan kecil dalam proporsi fitness, sehingga menghasilkan fluktuasi lebih besar.

Pada sisi crossover, uniform crossover menghasilkan perubahan struktur labirin yang lebih terkontrol dibanding arithmetic crossover. Arithmetic crossover lebih agresif dan dapat menghasilkan topologi labirin yang sangat berbeda dari induknya, yang menjelaskan fluktuasi fitness yang lebih besar terutama pada konfigurasi tanpa elitisme.

D. Interpretasi Fitness terhadap Kesulitan Labirin

Fitness yang rendah menunjukkan jarak yang kecil antara rata-rata langkah DFS dan target kesulitan menengah. Dengan demikian, labirin yang dihasilkan:

- tidak terlalu mudah (jalan lurus dan pendek),
- tidak terlalu membingungkan (percabangan yang berlebihan),
- memberikan tingkat kesulitan yang stabil dan terkontrol.

Dari seluruh grafik, dapat disimpulkan bahwa GA yang dirancang bekerja dengan efektif dalam menghasilkan labirin yang solvable dan memiliki tingkat kesulitan konsisten dengan target evaluasi DFS.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa Algoritma Genetika dapat digunakan secara efektif untuk menghasilkan labirin berukuran 12×12 dengan tingkat kesulitan yang terkontrol menggunakan fungsi fitness berbasis simulasi DFS. Delapan konfigurasi GA diuji sebagai kombinasi dari dua metode seleksi dan dua operator crossover, masing-masing dengan dan tanpa elitisme.

Hasil eksperimen memperlihatkan bahwa meskipun seluruh konfigurasi mencapai nilai fitness akhir yang sama, dinamika evolusinya sangat bergantung pada mekanisme GA yang digunakan. Elitisme terbukti memiliki pengaruh paling besar dalam menjaga stabilitas evolusi dan mencegah hilangnya solusi terbaik. Tournament selection menghasilkan proses evolusi yang lebih konsisten dibanding roulette, sementara uniform crossover memberikan perubahan struktur yang lebih stabil dibanding arithmetic crossover.

Secara keseluruhan, kombinasi tournament selection, uniform crossover, dan elitisme merupakan konfigurasi yang paling stabil dan efisien dalam menghasilkan labirin yang solvable dan memiliki tingkat kesulitan sesuai target. Penelitian ini menunjukkan bahwa pendekatan GA dengan evaluasi DFS dapat digunakan sebagai kerangka yang fleksibel untuk merancang labirin yang menantang sekaligus terkontrol tingkat kompleksitasnya.

REFERENCES

- [1] E. K. Susanto, R. Fachruddin, M. I. Diputra, D. Herumurti, and A. A. Yudianto, "Maze Generation Based on Difficulty using Genetic Algorithm with Gene Pool," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 554–559, 2020.
- [2] H. Zhang and X. Huang, "Design and Implement of the Complex Maze Shortest Path Simulation System Based on Improved Ant Colony Optimization Algorithm," in *2020 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1319–1323, 2020.
- [3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [4] J. Buckley, "Maze Generation Algorithms," *Journal of Game Development*, 2017.