



UNIVERSITAS
GADJAH MADA

Penerapan Algoritma Genetika untuk Desain Permainan Labirin

Bagus Cipta Pratama (23/561539/PA/22097)

David Neilleen Irvinne (23/517639/PA/22199)

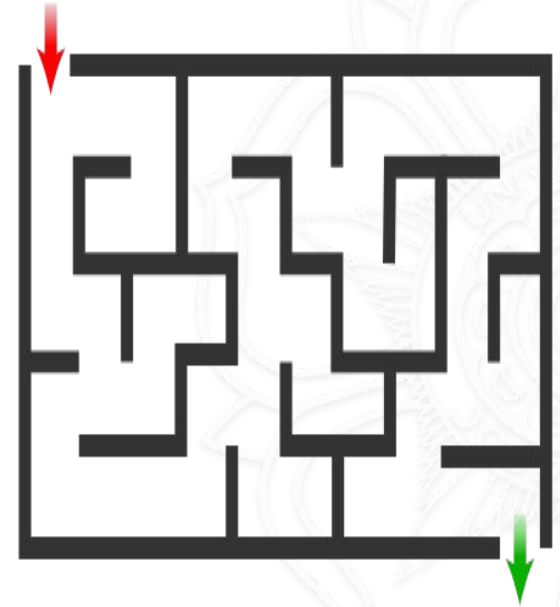
Syafran Abdillah Erdin (23/521752/PA/22444)

Muhammad Dhiwaul Akbar (23/523237/PA/22513)



Latar Belakang

- 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.



Proposed method

- 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
- dua metode seleksi (roulette dan tournament), dua operator crossover (uniform biased dan simple arithmetic), serta dua konfigurasi evolusi (dengan dan tanpa elitisme)

Encoding

Sebuah gen dapat direpresentasikan sebagai satu bilangan desimal 4-bit dengan setiap bit merepresentasikan :

- Bit 0 -> dinding kiri
- Bit 1 -> dinding atas
- Bit 2 -> dinding kanan
- Bit 3 -> dinding bawah

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

Jadi, setiap gen berada di rentang $[0, 15]$.

Encoding

Sebuah konfigurasi labirin sama dengan satu kromosom. Kromosom adalah sequence of gen (konfigurasi sebuah sel) atau lebih formal :

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

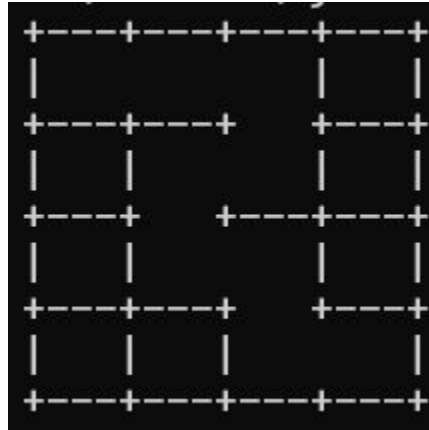
Contoh kromosom : (11, 10, 6, 15, 15, 3, 12, 15, 15, 9, 6, 15, 15, 15, 9, 14)

(3, 6, 11, 6, 5, 9, 6, 5, 5, 3, 8, 12, 13, 9, 10, 14)

Decoding

Gen : 11 \rightarrow 1011 \rightarrow Sel tersebut memiliki dinding kiri, kanan, dan bawah, tetapi tidak memiliki dinding atas.

Kromosom : (11, 10, 6, 15, 15, 3, 12, 15, 15, 9, 6, 15, 15, 15, 9, 14)



Populasi Awal

Tahapan pembentukan populasi awal:

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.

Normalisasi

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:

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

Repairing

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 dan agar seluruh area terhubung, mekanisme perbaikan memilih salah satu sisi untuk dibuka sehingga:

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

Analisis Keterselesaian

Labirin memiliki beberapa aturan:

- Labirin dimodelkan sebagai *undirected graph* $G(V, E)$
- Setiap cell adalah node pada graph, $c_{ij} \in V$
- Edge atau jalan (u, v) dikatakan ada jika cell u dan cell v memiliki shared wall yang terbuka, $(u, v) \in E$.

Untuk menentukan suatu labirin solvable atau tidak :

- Inisialisasi titik start pada $(0, 0)$ dan finish di $(N - 1, N - 1)$
- Lakukan BFS
- Apabila BFS me-return not inf, maka labirin solvable. Sebaliknya, apabila labirin me-return nilai inf, maka labirin unsolvable

Fitness Function

Setiap individu menjalani 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:

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

Fitness dihitung berdasarkan kedekatan cost value dengan nilai target:

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

Di mana, $\text{mid} = \frac{\text{lower bound} + \text{upper bound}}{2}$, $\text{lower bound} = (N + N - 1)$, $\text{upper bound} \approx 1.8 N^2$.

Selection

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

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

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

Crossover

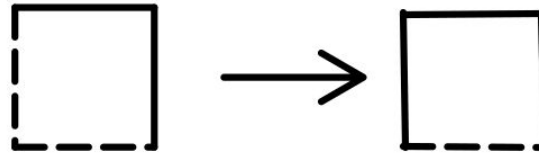
- Uniform Biased Crossover: setiap komponen anak dipilih dari salah satu induk berdasarkan probabilitas tertentu, memungkinkan dominasi salah satu induk.
- Arithmetic Crossover: nilai genetik anak merupakan kombinasi linier dari dua induk, memberikan transisi yang lebih halus di ruang solusi.

Pendekatan ini memungkinkan pembentukan struktur labirinbaru dengan karakteristik campuran.

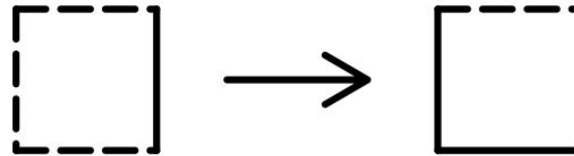
Mutation

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.

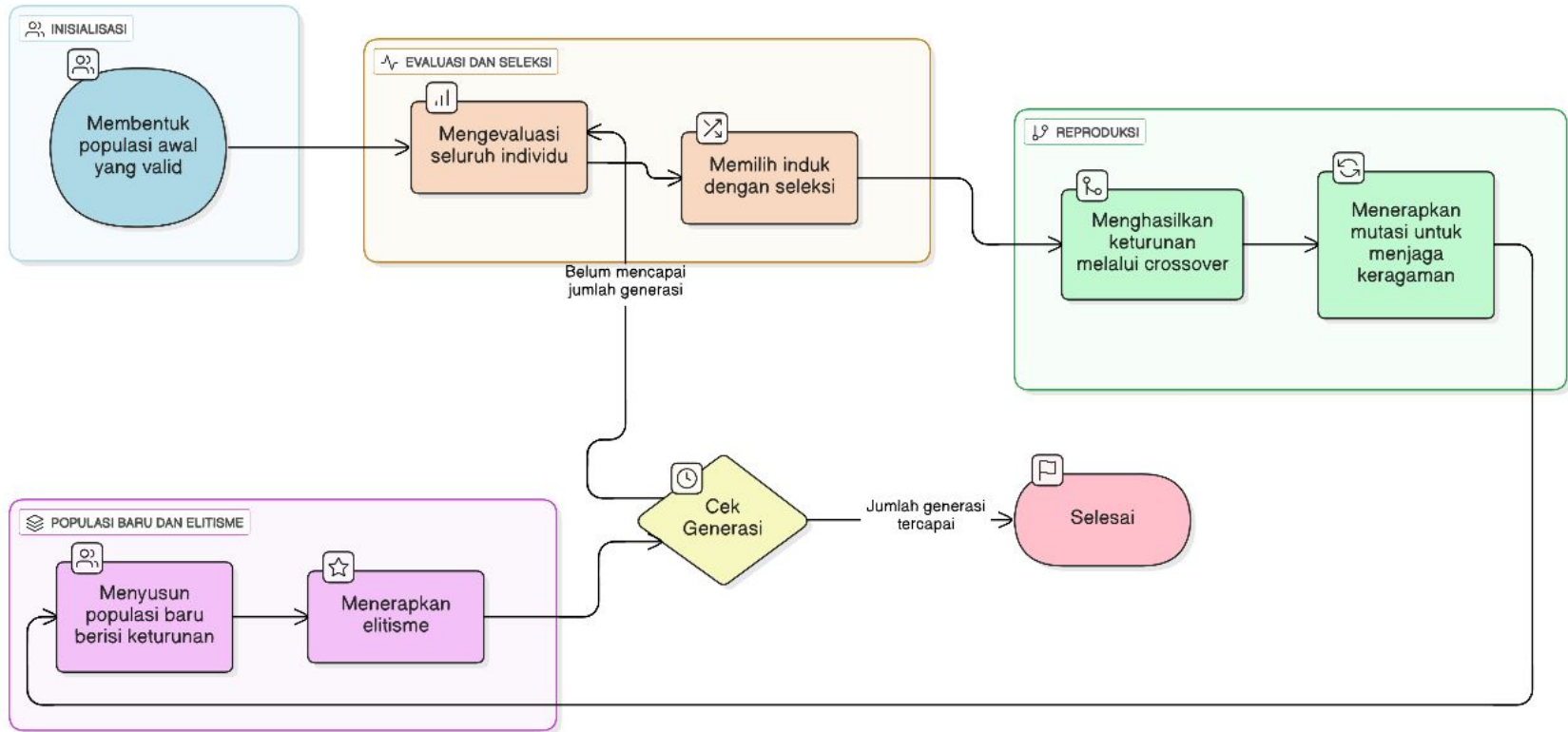


Elitism

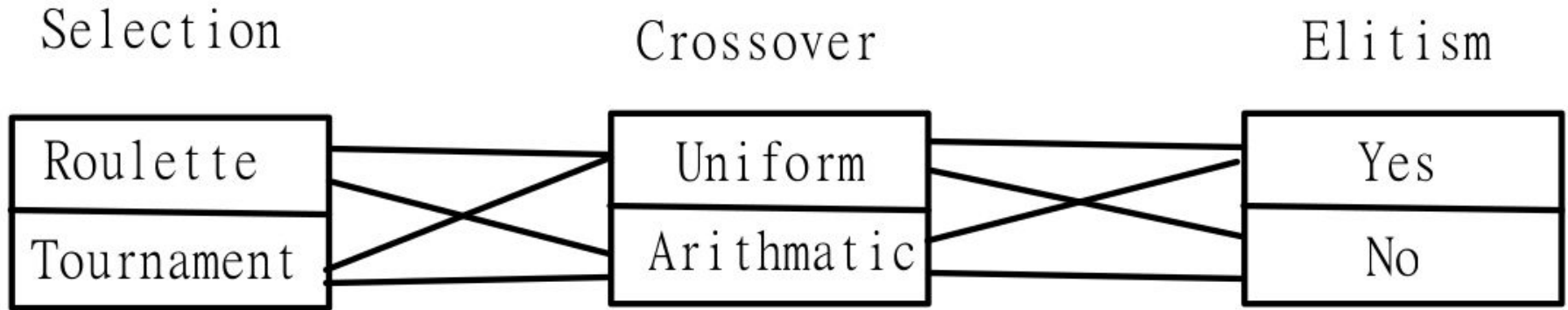
Elitisme memastikan bahwa sejumlah individu dengan kualitas terbaik pada suatu generasi dipertahankan tanpa perubahan pada generasi berikutnya. Hal ini dilakukan dengan:

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

The Bigger Picture



Experiment



Experiment - Parameter

Ukuran labirin (N)	12
Ukuran populasi	64
Jumlah generasi	32
Probabilitas mutasi kecil	0.02
Probabilitas mutasi besar	0.1
Tour-size	32
#DFS run / evaluasi	8

Experiment - Metric

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

Result

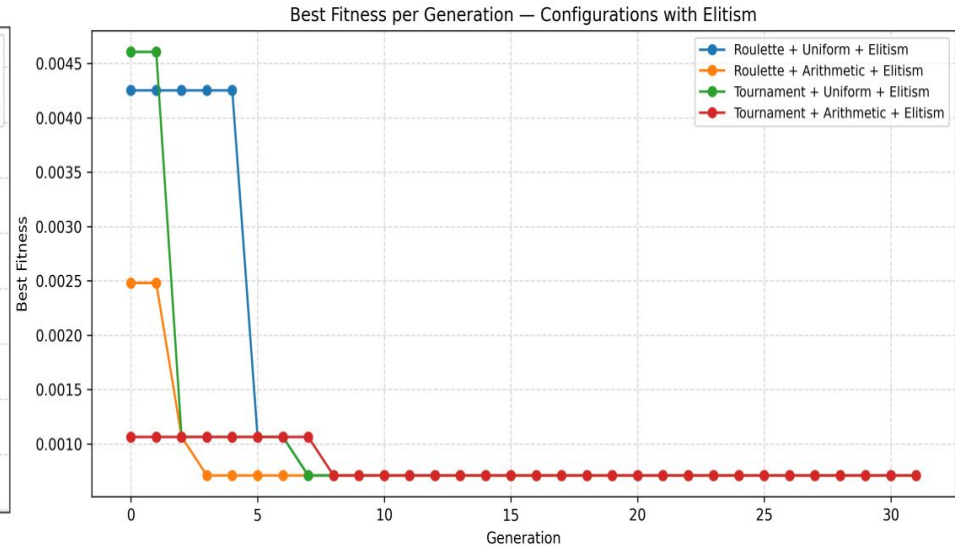
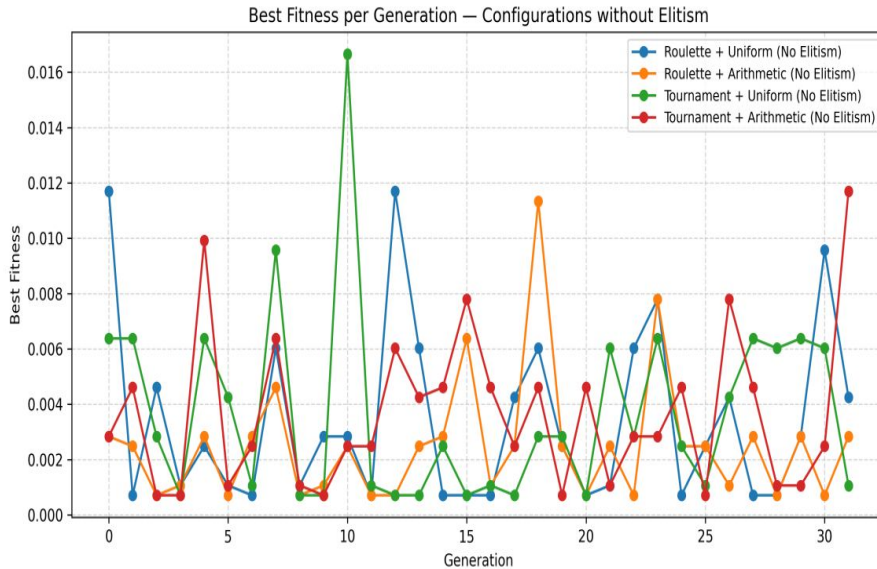
TABLE I
HASIL AKHIR DELAPAN KONFIGURASI EKSPERIMEN

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

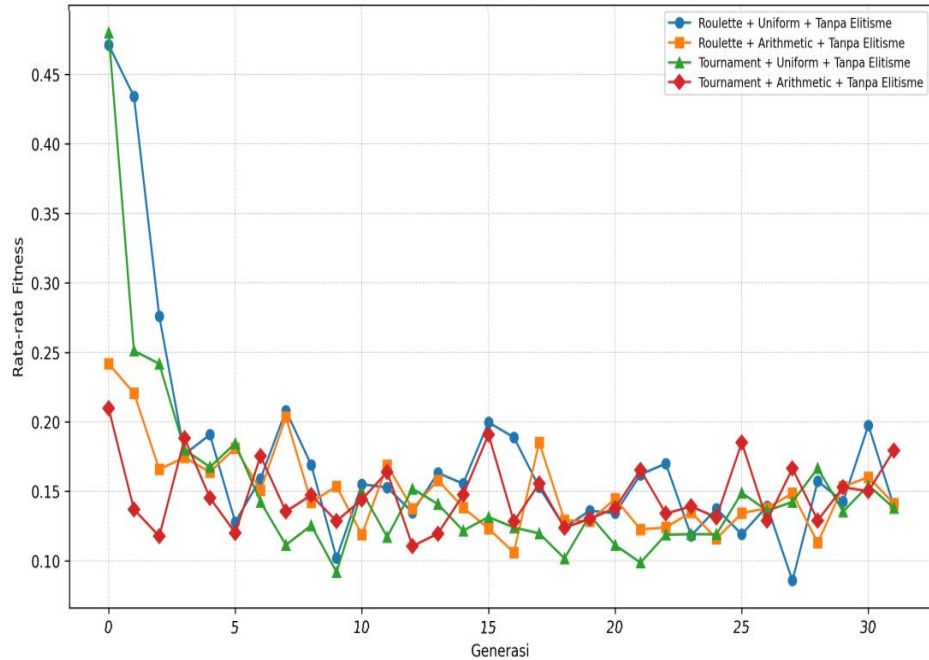
No.	Different Shortest Paths	Avg Steps Taken
1	2-4	141
2	1-6	140.25-143
3	1	141
4	2-709	139.5-142.5
5	6-12	141
6	4-64	138.75-142
7	2	141
8	11-273	140-142.75

Result - Best Fitness

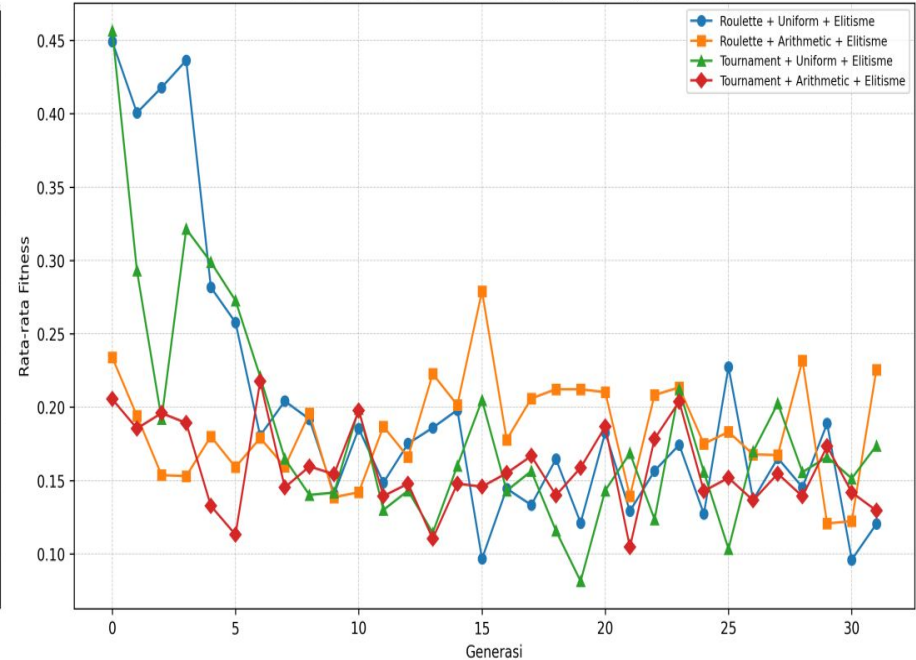


Result - Mean

Rata-rata Fitness per Generasi - Konfigurasi tanpa Elitisme



Rata-rata Fitness per Generasi - Konfigurasi dengan Elitisme



Kesimpulan

- Algoritma Genetika dapat digunakan secara efektif untuk menghasilkan labirin berukuran 12×12 dengan tingkat kesulitan yang terkontrol menggunakan fungsi fitness berbasis simulasi DFS.
- 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.



UNIVERSITAS
GADJAH MADA

Terima Kasih

