
Visualisasi Algoritma *Backtracking* Untuk Solusi *Shortest Path Problem* Pada *Grid 2D*

Fadhilah Nur Hamal*¹, Sugiarto Cokrowibowo², Arnita Iriani³

^{1,2,3}Program Studi Teknik Informatika, Universitas Sulawesi Barat
E-mail: *¹fadhilahnurhamal@gmail.com, ²sugiarto.cokrowibowo@unsulbar.ac.id,
³arnitairianti@unsulbar.ac.id

Abstrak

Masalah jalur terpendek pada grid 2d sering terjadi di berbagai konteks, seperti perencanaan rute dalam navigasi, pengiriman barang dalam logistik, atau perencanaan penempatan dalam desain ruangan. algoritma *backtracking* merupakan metode pencarian sistematis yang dapat digunakan untuk menemukan jalur terpendek dari satu titik ke titik lain dalam grid 2d. Namun, pemahaman konsep dan proses algoritma *backtracking* dalam menyelesaikan masalah ini bisa menjadi sulit karena sifatnya yang abstrak dan kompleks. karena itu, penting untuk mengembangkan pendekatan tentang algoritma *backtracking* dan solusi jalur terpendek pada grid 2d. Pendekatan yang dapat digunakan adalah melalui visualisasi interaktif. dengan menggunakan visualisasi, pengguna dapat melihat langkah-langkah algoritma secara visual dalam konteks grid 2d yang lebih intuitif.

Kata kunci: visualisasi, algoritma *backtracking*, *shortest path problem*, grid 2 dimensi, interaktif.

Abstract

The shortest path problem on a 2d grid often occurs in various contexts, such as route planning in navigation, freight forwarding in logistics, or placement planning in room design. The *Backtracking* algorithm is a systematic search method that can be used to find the shortest path from one point to another in a 2D grid. However, understanding the concepts and processes of *backtracking* algorithms in solving these problems can be difficult due to their abstract and complex nature. Therefore, it is important to develop an approach about *backtracking* algorithms and shortest path solutions on 2D grids. The approach that can be used is through interactive visualization. Using visualization, users can visually view the steps of the algorithm in the context of a more intuitive 2D grid.

Keywords: visualization, *backtracking* algorithm, *shortest path problem*, 2-dimensional grid, interactive.

1. PENDAHULUAN

Perkembangan bidang komputasi dan pemrosesan grafis dalam beberapa tahun terakhir, telah memberikan kontribusi besar dalam pemahaman dan visualisasi algoritma. Salah satu algoritma yang sering digunakan dalam pemecahan masalah jalur terpendek adalah algoritma *backtracking*. Algoritma *backtracking* merupakan salah satu bentuk algoritma yang banyak digunakan oleh para programmer ataupun pengguna komputer ahli untuk menyelesaikan suatu permasalahan komputasional pada perangkat komputer yang mereka gunakan. Dalam programming algoritma *backtracking*, rekursi adalah kunci dari *programming backtracking*. (dosenit.com).

Pada aplikasi dunia nyata, masalah jalur terpendek pada *grid 2 dimensi* sering terjadi dalam berbagai konteks, seperti perencanaan rute dalam navigasi, pengiriman barang dalam

logistik, atau perencanaan penempatan dalam desain ruangan. Namun, pemahaman konsep dan proses algoritma *backtracking* dalam menyelesaikan masalah ini bisa menjadi sulit bagi banyak orang karena sifatnya yang abstrak dan kompleks.

Oleh karena itu, penting untuk mengembangkan pendekatan yang memungkinkan pemahaman yang lebih baik tentang algoritma *backtracking* dan solusi jalur terpendek pada *grid 2 dimensi*. Salah satu pendekatan yang dapat digunakan adalah melalui visualisasi interaktif. Dengan menggunakan visualisasi, pengguna dapat melihat langkah-langkah algoritma secara visual dalam konteks *grid 2 dimensi* yang lebih intuitif.

Visualisasi algoritma *backtracking* untuk solusi jalur terpendek pada *grid 2 dimensi* dapat memberikan manfaat yang signifikan. Ini tidak hanya memfasilitasi pemahaman yang lebih baik tentang konsep dan mekanisme algoritma *backtracking*, tetapi juga memungkinkan pengguna untuk berinteraksi dengan visualisasi tersebut. Pengguna dapat melakukan perubahan pada *grid*, memulai ulang pencarian, atau mengamati animasi langkah demi langkah, yang semuanya dapat meningkatkan pemahaman mereka tentang algoritma.

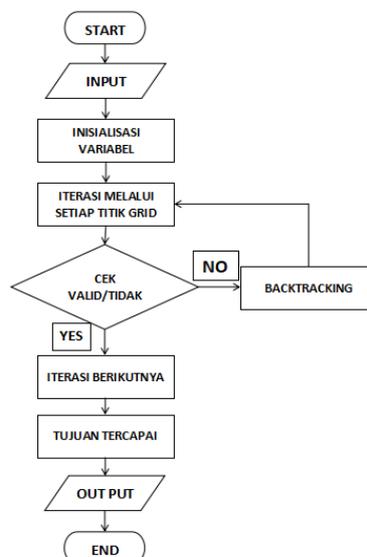
Namun, meskipun terdapat berbagai pendekatan dan perangkat lunak yang ada, masih ada kebutuhan untuk melakukan penelitian lebih lanjut dalam pengembangan visualisasi algoritma *backtracking* untuk solusi jalur terpendek pada *grid 2 dimensi*. Penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan merancang dan mengimplementasikan sebuah perangkat lunak yang menggabungkan algoritma *backtracking* dengan visualisasi interaktif untuk memberikan pemahaman yang lebih baik tentang solusi jalur terpendek pada *grid 2 dimensi*.

2. METODE

Dalam penelitian ini menggunakan penelitian eksperimental, peneliti dapat merancang dan mengimplementasikan perangkat lunak yang menggabungkan algoritma *backtracking* dengan visualisasi interaktif. Peneliti menggunakan *grid 2D* dengan rintangan, titik awal, dan titik tujuan yang dapat dimasukkan. Selanjutnya, dapat dilakukan serangkaian eksperimen untuk memvalidasi dan menguji efektivitas visualisasi dalam membantu pemahaman dan interaksi pengguna dengan algoritma *backtracking*.

2.1 Gambaran proses

Gambaran proses visualisasi algoritma *backtracking* untuk solusi *shortest path problem* pada *grid 2 dimensi* sebagai berikut (Gambar 3.1):



Gambar 1 Flowchart

1. Mulai: Simbol awal dari *flowchart*, menandakan dimulainya aliran program.
2. *Input Grid* dan Tujuan: Simbol yang menunjukkan pengambilan *input grid* 2 dimensi dari pengguna, serta koordinat tujuan dalam *grid*.
3. Inisialisasi Variabel: Simbol untuk menginisialisasi variabel-variabel yang akan digunakan dalam algoritma *backtracking*, seperti posisi awal, jalur sementara, jalur terpendek, dan lainnya.
4. Iterasi Melalui Setiap Titik *Grid*: Simbol untuk melakukan iterasi melalui setiap titik *grid* 2 dimensi. Mulai dari posisi awal, algoritma akan mempertimbangkan semua kemungkinan langkah (ke atas, ke bawah, ke kiri, ke kanan) yang dapat diambil dalam *grid*.
5. Cek Kemungkinan Langkah: Simbol yang menunjukkan pengujian apakah langkah tersebut valid atau tidak, dengan memeriksa batasan *grid*, apakah langkah mengarah ke rintangan atau tidak, dan sebagainya.
6. Jika Langkah *Valid*: Jika langkah tersebut *valid*, simbol ini akan menambahkan langkah tersebut ke jalur sementara dan memperbarui posisi saat ini.
7. Jika Tujuan Tercapai: Jika posisi saat ini sama dengan koordinat tujuan, simbol ini akan memeriksa apakah jalur sementara merupakan jalur terpendek yang ditemukan sejauh ini. Jika iya, jalur terpendek akan diperbarui.
8. *Backtrack*: Jika tujuan belum tercapai atau tidak ada langkah yang *valid*, algoritma akan melakukan *backtrack*. Ini berarti menghapus langkah terakhir dari jalur sementara dan mengembalikan posisi saat ini ke langkah sebelumnya.
9. Iterasi Berikutnya: Simbol ini menunjukkan iterasi ke titik *grid* berikutnya yang belum dieksplorasi.
10. Ulangi Langkah 5-9: Langkah-langkah 5 hingga 9 akan diulang hingga semua titik *grid* telah dieksplorasi.
11. *Output* Jalur Terpendek: Simbol yang menunjukkan output jalur terpendek yang telah ditemukan oleh algoritma *backtracking*.
12. Selesai: Simbol akhir yang menandakan akhir dari aliran program.

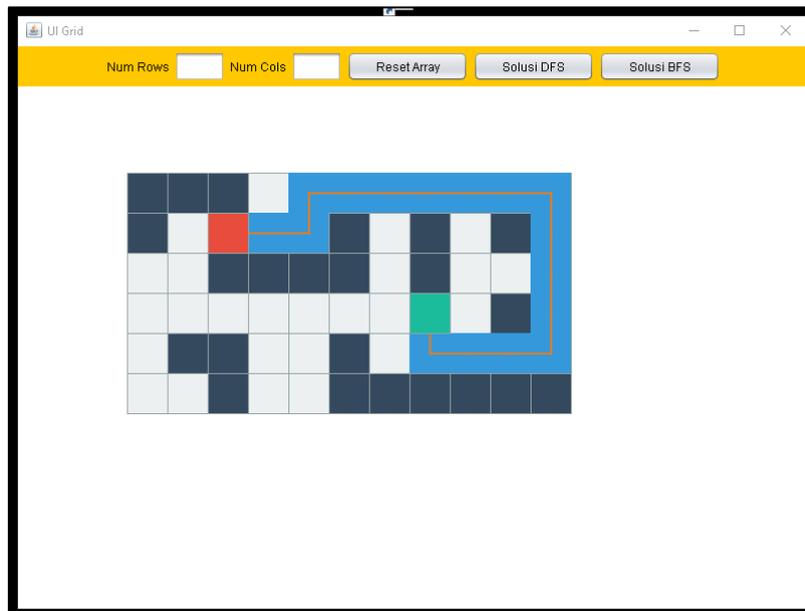
3. HASIL DAN PEMBAHASAN

A. HASIL

1. Hasil Visualisasi

Hasil visualisasi dari setiap kasus uji ditampilkan dalam aplikasi. Hasil visualisasi untuk jalur yang diambil akan diberi warna biru, titik start akan diberi warna merah, titik finish diberi warna hijau, dan untuk titik-titik rintangannya diberi warna hitam. Selain itu terdapat kolom *num rows* dan kolom *num cols* untuk mengatur ukuran *grid*. terdapat juga tombol solusi DFS dan solusi BFS sebagai tombol pencarian solusi dari jalur terpendek. Hal tersebut untuk memudahkan pemahaman pengguna.

Berikut adalah beberapa hasil visualisasi dari kasus uji:



Gambar 2 Visualisasi Kasus Uji 1

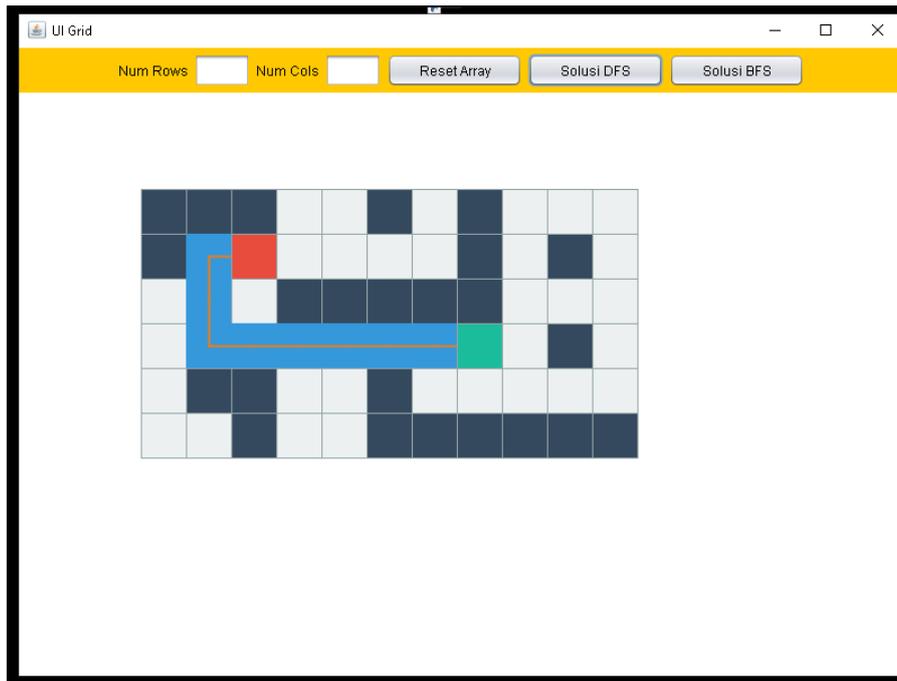
Kasus Uji 1:

	0	1	2	3	4	5	6	7	8	9	10
0	-1	-1	-1	0	4	5	6	7	8	9	10
1	-1	0	1	2	3	-1	0	-1	0	-1	11
2	0	0	-1	-1	-1	-1	0	-1	0	0	12
3	0	0	0	0	0	0	0	0	0	-1	13
4	0	-1	-1	0	0	-1	0	17	16	15	14
5	0	0	-1	0	0	-1	-1	-1	-1	-1	-1
X	1	1	1	0	0	0	0	0	0	0	1
Y	2	3	4	4	5	6	7	8	9	10	10
			2	3	4	4	4	4	3		
			10	10	10	9	8	7	7		

Berikut perhitungan dari kasus uji 1:

1. $x=1, y=2$
2. $(x,y+1)=(1,2+1)=(1,3)$ arah utara (kanan)
3. $(x,y+1)=(1,3+1)=(1,4)$ arah timur (depan)
4. $(x-1,y)=(1-1,4)=(0,4)$ arah timur (kiri)
5. $(x,y+1)=(0,4+1)=(0,5)$ arah utara (kanan)
6. $(x,y+1)=(0,5+1)=(0,6)$ arah timur (depan)
7. $(x,y+1)=(0,6+1)=(0,7)$ arah timur (depan)
8. $(x,y+1)=(0,7+1)=(0,8)$ arah timur (depan)
9. $(x,y+1)=(0,8+1)=(0,9)$ arah timur (depan)
10. $(x,y+1)=(0,9+1)=(0,10)$ arah timur (depan)

11. $(x+1,y)=(0+1,10)=(1,10)$ arah timur (kanan)
12. $(x+1,y)=(1+1,10)=(2,10)$ arah selatan (depan)
13. $(x+1,y)=(2+1,10)=(3,10)$ arah selatan (depan)
14. $(x+1,y)=(3+1,10)=(4,10)$ arah selatan (depan)
15. $(x,y-1)=(4,10-1)=(4,9)$ arah selatan (kanan)
16. $(x,y-1)=(4,9-1)=(4,8)$ arah barat (depan)
17. $(x,y-1)=(4,8-1)=(4,7)$ arah barat (depan)
18. $(x-1,y)=(4-1,7)=(3,7)$ arah barat (kanan)



Gambar 3 Visualisasi Kasus Uji 2

Kasus Uji
2:

	0	1	2	3	4	5	6	7	8	9	10
0	-1	-1	-1	-2	-2	-1	-2	-1	0	0	0
1	-1	2	1	-2	-2	-2	-2	-1	0	-1	0
2	-2	3	0	-1	-1	-1	-1	-1	0	0	0
3	-2	4	5	6	7	8	9	0	0	-1	0
4	-2	-1	-1	0	0	-1	0	0	0	0	0
5	-2	-2	-1	0	0	-1	-1	-1	-1	-1	-1

X	1	1	2	3	3	3	3	3	3	3
Y	2	1	1	1	2	3	4	5	6	7

Berikut perhitungan dari uji kasus 2:

1. $x=1,y=2$
2. $(x,y-1)=(1,2-1)=(1,1)$ arah utara (kiri)

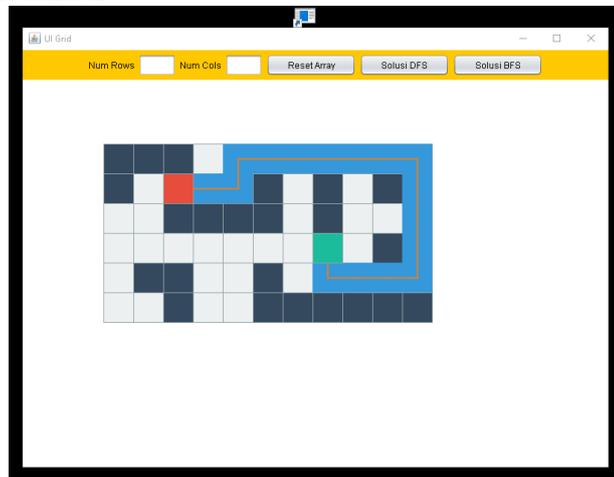
3. $(x+1,y)=(1+1,1)=(2,1)$ arah barat (kiri)
4. $(x+1,y)=(2+1,1)=(3,1)$ arah selatan (depan)
5. $(x,y+1)=(3,1+1)=(3,2)$ arah selatan (kiri)
6. $(x,y+1)=(3,2+1)=(3,3)$ arah timur (depan)
7. $(x,y+1)=(3,3+1)=(3,4)$ arah timur (depan)
8. $(x,y+1)=(3,4+1)=(3,5)$ arah timur (depan)
9. $(x,y+1)=(3,5+1)=(3,6)$ arah timur (depan)
10. $(x,y+1)=(3,6+1)=(3,7)$ arah timur (depan)

2. Analisis Hasil

Hasil pengujian menunjukkan bahwa aplikasi visualisasi algoritma *backtracking* berhasil menampilkan jalur terpendek dari posisi awal ke posisi tujuan pada *grid 2D*. Pengujian juga menunjukkan bahwa algoritma *backtracking* mampu mencari jalur alternatif yang menghindari rintangan dan mencapai tujuan.

B. PEMBAHASAN

1. Deskripsi Aplikasi Visualisasi



Gambar 4 Aplikasi Visualisasi

Aplikasi visualisasi (Gambar 4) yang dikembangkan dalam penelitian ini menggunakan bahasa pemrograman java untuk mengimplementasikan algoritma *backtracking* pada *grid 2D*. Aplikasi ini dirancang untuk membantu pengguna memahami proses pencarian jalur terpendek menggunakan algoritma *backtracking*. Pada aplikasi, pengguna dapat mengatur ukuran grid, posisi awal dan tujuan, serta menambahkan rintangan pada *grid* sesuai keinginan.

2. Algoritma pencarian solusi *shortest path* pada *grid 2D*

Algoritma yang diterapkan untuk pencarian solusi pada *shortest path problem grid 2D* adalah sebagai berikut :

- a. Pilih sel pertama kali secara acak pada *grid*, akan di kenali sebagai *current* sel (*ptCurrent*).
- b. Cek pada sel tetangga, apakah sel sudah pernah dikunjungi. Gunanya untuk menentukan *current* sel berikutnya. Karena ini adalah sel pertama maka belum ada sel yang dikunjungi. Maka data sel disimpan di *Stack generate*.
- c. Cara pengecekan yaitu dengan gerakan yang sudah ditentukan yaitu atas $(x,y-1)$, kanan $(x+1, y)$, kiri $(x-1, y)$ bawah $(x, y+1)$.

- d. Proses ini dilakukan terus menerus sampai kondisi total sel = sel yang dikunjungi dan $ptCurrent$ sel = $ptEnd$ sel. Apabila ditemui sel yang sudah dikunjungi maka dilakukan proses *backtracking* yaitu dengan mengeluarkan data dari *stack generated* yang mana data tersebut mengarah ke sel yang telah dikunjungi kembali ke data di mana sel masih memiliki kemungkinan pengecekan sel tetangga lainnya.

4. KESIMPULAN

Penelitian ini bertujuan untuk membangun media dan menampilkan visualisasi algoritma *backtracking* dalam penyelesaian *shortest path problem* pada *grid 2D*. Dari hasil penelitian dan analisis yang telah dilakukan, dapat diambil kesimpulan bahwa media visualisasi algoritma *backtracking* telah berhasil dibangun dan menampilkan visualisasi algoritma *backtracking* dalam penyelesaian *shortest path problem* pada *grid 2D* dengan menggunakan bahasa pemrograman java. Media ini memungkinkan pengguna untuk memahami dan mengikuti langkah-langkah algoritma *backtracking* secara visual.

REFERENSI

- [1] dosenit.com/kuliah-it/pemrograman/algoritma-backtracking
- [2] A.Suryadibrata , J. C. hristian Young“Visualisasi Algoritma sebagai Sarana Pembelajaran *K-Means Clustering*“ *ULTIMATICS*, Vol. XII, No. 1, pp. 25-29 Juni 2020
- [3] Alvin Lander, Angel Metanosa Afinda,Zakiya Ainur Rohman “PERBANDINGAN *PROBLEM N-QUEEN* MENGGUNAKAN ALGORITMA *BACKTRACKING* DAN ALGORITMA GENETIKA” 2019
- [4] <https://dosenit.com/kuliah-it/depth-first-search>
- [5] Intan Cahyaningtyas, “Menangani Masalah Lintasan Terpendek Menggunakan Algoritma *Bellman-Ford*” 2016
- [6] PENERAPAN ALGORITMA *BACKTRACKING* PADA PERMAINAN *MATH MAZE* | Teneng | Jurnal Informatika (ukdw.ac.id)