Penerapan Algoritma Backtracking dalam Maze Game

Michael Joseph Christian - 11419005 (*Author*)
Berliana Simamora - 11419016 (*Author*)
Agus Rokyanto Silaban - 11419045 (*Author*)
Meyliza Veronica Siregar - 11419058 (*Author*)

Program Studi Teknologi Informasi atau Teknologi Rekayasa Perangkat Lunak
Fakultas Vokasi Institut Teknologi Del, Laguboti, Sumatera Utara
E-mail (gmail): meylizasiregar65@gmail.com

***Abstract*— Maze (labirin) merupakan teka-teki berupa game atau permainan. Maze (labirin) memiliki jalur yanng bercabang dan terdapat jalan buntu. Algoritma pencarian memudahkan untuk menemukan jalan keluar dari maze (labirin), tetapi tidak semua algoritma pencarian dapat diimplementasikan dalam labirin. Oleh karena itu, menentukan algoritma yang tepat adalah salah satu faktor terpenting untuk mencari jalan keluar dari maze (labirin). Algoritma backtracking adalah salah satu algoritma pencarian yang paling efisien karena algoritma tersebut melacak kembali dari node target untuk melihat apakah solusi yang dicari mengarah ke node target yang diinginkan.**

Dalam makalah ini, algoritma backtracking atau runut balik digunakan pada game Maze (labirin) untuk mencari jalur yang tepat menuju jalan keluar dari maze (labirin).

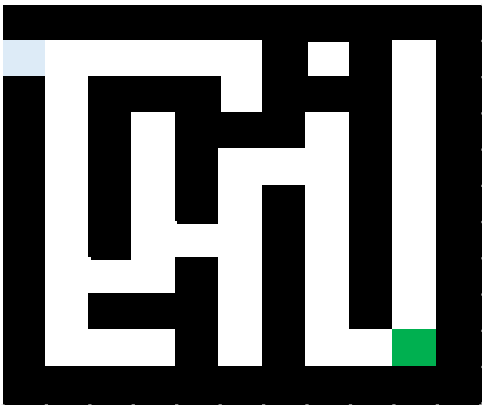
Algoritma backtracking, akan mencari semua kemungkinan solusi yang mengarah ke solusi yang sesungguhnya. Oleh karena itu, algoritma backtracking dapat mengurangi langkah-langkah yang tidak perlu dalam pencarian dan menemukan rute terpendek ke pencarian. Dengan demikian, algoritma Backtracking dapat membuat waktu pencarian yang dibutuhkan untuk mencari solusi pada permainan Maze menjadi lebih cepat.

Keywords—backtracking; game; maze

I. PENDAHULUAN

Game atau permainan merupakan sarana yang dapat digunakan sebagai hiburan baik untuk diri sendiri maupun kelompok. Game juga dapat digunakan untuk mengisi waktu kosong dan untuk melatih otak. Seiring dengan perkembangan waktu, jenis permainan yang ada pada saat ini juga semakin berkembang pula. Dan permainan yang sering dilakukan masyarakat pada jaman ini adalah permainan yang dilakukan di komputer. Permainan yang sudah ada saat ini juga memberikan permasalahan-permasalahan pada penggunanya dan sangat mungkin dihadapi di kehidupan nyata.

Salah satu permainan yang sering terdapat di dunia nyata maupun dalam komputer adalah Game Maze. Game Maze merupakan permainan yang ditujukan untuk mencari jalur yang tepat untuk mencapai tujuan yang telah ditetapkan. Dalam permainan maze, jalur yang digunakan adalah jalur yang berliku dan terdapat jalan buntu. Untuk sampai ke tujuan yang telah ditetapkan, pemain harus menemukan jalur yang tepat.



Gambar 1. Sebuah Maze (Labirin)

Untuk menyelesaikan permasalahan pada Game Maze dapat dengan menggunakan beberapa algoritma, seperti algortima Brute Force maupun algoritma Bactracking. Algoritma Brute Force dilakukan dengan mencari solusi persoalan diantara semua kemungkinan solusi. Algoritma Brute Force tentu membutuhkan waktu yang lebih lama karena mencoba semua kemungkinan solusi walaupun solusi tersebut tidak mengarah ke solusi yang sesungguhnya. Sedangkan pada algoritma Backtracking, hanya akan mencari semua kemungkinan solusi yang mengarah ke solusi yang sesungguhnya. Maka waktu pencarian solusi akan lebih cepat.

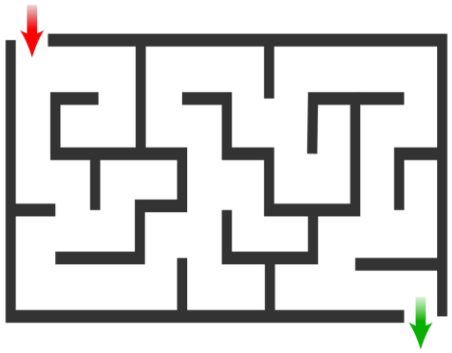
Gambar 2. Solusi Maze (Labirin)

II. LANDASAN TEORI

A. *Maze Game*

Permainan maze merupakan permainan yang cukup populer baik di kalangan anak-anak maupun dewasa. Permainan maze juga dapat ditemukan pada kehidupan nyata yaitu struktur gang-gang kecil dan sempit di pemukiman penduduk yang padat.

Game Maze adalah sebuah permainan sederhana yang bertujuan untuk menemukan jalur yang tepat untuk mencapai tujuan yang ditetapkan. Dalam permainan maze, pemain perlu menemukan beberapa jalur berbentuk persegi di bagian maze. Jalur ini dilalui di setiap baris atau kolom. Permainan maze merupakan permainan yang cukup populer baik di kalangan anak-anak maupun dewasa. Permainan maze juga dapat ditemukan pada kehidupan nyata.



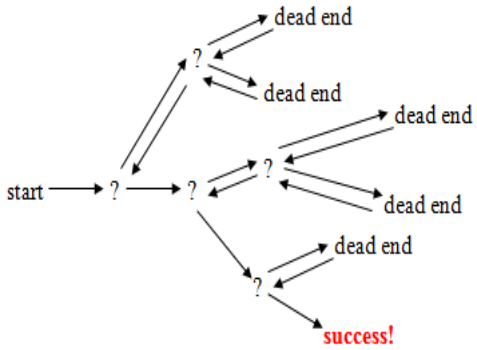
Gambar 3. Maze Game
Sumber : http://rmstechclass.weebly.com/maze-game.html

Arah pergerakan jalur yang dilalui oleh pemain antar lain atas (up), bawah (down), kiri (left), kanan (right). Setiap jalur yang sudah dilewati oleh pemain akan ditandai untuk menunjukkan bahwa posisinya telah lewat, lalu lanjut lagi ke titik berikutnya.

B. *Algoritma Backtracking*

Algoritma backtracking pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Dalam perkembangannya beberapa ahli seperti RJ Walker, Golomb, dan Baumert menyajikan uraian umum tentang backtracking dan penerapannya dalam berbagai persoalan dan aplikasi. Beberapa game populer seperti Sudoku, Labirin, Catur juga dapat diimplementasikan dengan menggunakan algoritma backtracking atau runut balik.

Algoritma backtracking atau runut-balik merupakan perbaikan dari algoritma brute-force, secara sistematis algoritma bactracking mencari solusi persoalan di antara semua kemungkinan yang ada. Namun hanya pencarian yang mengarah ke solusi sesungguhnya saja yang dikembangkan, sehingga waktu pencarian dapat lebih cepat. Pada algoritma backtracking semua solusi dibuat dalam bentuk pohon solusi (pohon ini tentunya berbentuk abstrak) dan algoritma akan menelusuri pohon tersebut secara DFS (Depth Field Search) sampai ditemukan solusi yang sesungguhnya.



Gambar 4. Ilustrasi Algoritma Backtracking

Sumber : https://www.ilmuskripsi.com/2016/05/algoritma-runut-balik-backtracking.html

Properti umum Algoritma Backtracking :

- Solusi Persoalan
  Solusi direpresentasikan sebagai vektor dengan n-tupel.
  Bentuk : X = (x1,x2,x3,..,xn)
- Fungsi Pembangkit
  Predikat T() digunakan untuk membangkitkan nilai untuk xk
  Bentuk : T(x[1], x[2], x[3],...,x[k-1])
- Fungsi Pembatas
  Predikat B() untuk menentukan untuk melanjutkan solusi atau dibuang
  Bentuk : B(x1, x2, x3,.., xk)

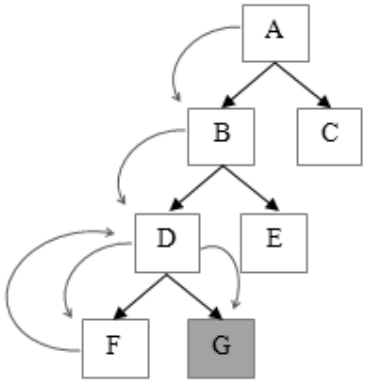
Solusi persoalan adalah kemungkinan solusi yang didapatkan dari permasalahan yang diberikan, sedangkan fungsi pembatas merupakan fungsi yang akan menentukan langkah selanjutnya berupa penerusan pencarian solusi ataupun melakukan backtrack.

Semua solusi yang mungkin disimpan dalam ruang solusi. Ruang solusi diatur dalam struktur pohon. Setiap simpul pohon mewakili keadaan masalah, dan tepi (cabang) ditandai dengan nilai xi. Rute root-to-leaf mewakili solusi yang mungkin, dan seluruh rute root-to-leaf membentuk ruang solusi. Organisasi pohon ruang solusi disebut pohon ruang status.

C. Prinsip Pencarian Solusi dengan Algoritma Backtracking

Prinsip dasar algoritma Backtracking adalah mencoba semua kemungkinan solusi yang ada.
Langkah-langkah untuk menemukan solusi menggunakan backtracking:

1. Solusinya dicari dengan membuat jalur root-to-leaf. Aturan formasi yang digunakan adalah yang mengikuti aturan Dephtfirst Search (DFS).
2. Simpul-simpul yang sudah dihasilkan dinamakan simpul hidup (live node).
3. Simpul hidup yang sedang diperluas dinamakan simpul-E (Expand-node).
4. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang.
5. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut "dibunuh" sehingga menjadi simpul mati (dead node).



Gambar 5. Algoritma Backtracking

6. Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan fungsi pembatas (bounding function).
7. Simpul mati tidak akan pernah diperpanjang.
8. Jika pembentukan jalur berakhir pada simpul mati, proses pencarian akan menelusuri simpul di atas
9. Kemudian lanjutkan untuk membuat simpul anak lainnya.
10. Kemudian simpul ini menjadi simpul baru E.
11. Ketika node tujuan tercapai, pencarian berakhir.

III. PENERAPAN METODE DAN ANALISIS

Algoritma yang tepat untuk menemukan jalan keluar dari labirin adalah algoritma backtracking. Algoritma ini mencoba suatu jalur hingga menemui jalan buntu, melakukan langkah sebelumnya (mundur) hingga menemukan jalur lain, kemudian mengulangi jalur tersebut lagi. Pada akhirnya, temukan jalan menuju pintu keluar, atau coba semua jalan dan putuskan bahwa tidak ada solusi. Untuk menguraikan algoritma backtracking, bagi jalur menjadi serangkaian langkah. Salah satu langkahnya adalah dengan memindahkan sel satuan ke arah tertentu. Arah yang dapat dilalui: atas (up), bawah (down), kiri (left), kanan (right).

Algoritma backtracking adalah algoritma rekursif yang bertujuan untuk memecahkan masalah yang diberikan dengan menguji semua jalur yang mungkin menuju solusi sampai solusi ditemukan. Setiap kali sebuah jalur diuji, jika solusi tidak ditemukan, algoritm akan mundur untuk menguji jalur lain dan seterusnya hingga solusi ditemukan atau semua jalur telah diuji.

Skenario algoritma backtracking pada game Maze adalah ketika mencoba menemukan jalan keluar dalam labirin. Setiap kali algoritma mencapai jalan buntu, maka algoritma akan mundur untuk mencoba jalan lain sampai menemukan jalan keluar atau semua jalan telah dijelajahi.

Secara garis besar, algoritma Backtracking dapat dipresentasikan sebagai berikut.

```
While belum sampai pada tujuan do
   If terdapat arah yang benar sehingga
       kita belum pernah berpindah ke sel
       pada arah tersebut
   then
       pindah satu langkah ke arah
       tersebut
   else
      backtrack langkah sampai terdapat
       arah seperti yang disebutkan di
       atas
   endif
endwhile
```

Algoritma backtracking untuk masalah Maze (labirin) yaitu prameter fungsinya adalah Labirin M, lalu labirin disimpan (labirin dapat diimplementasikan sebagai matriks 0/1, jalur direpresentasikan sebagai deretan 0 dan dinding direpresentasikan sebagai deretan 1).

```
function
 SolveMaze(input M : labirin)→ boolean
{ true jika solusi ditemukan, false jika
tidak }

Deklarasi
   arah : integer
{ up = 1, down, 2, left = 3, right = 4 }
```

```
Algoritma:
if
 solusi sudah ditemukan then
   return true
else
   for tiap arah gerakan (up, down, left,
right) do
       move(M, arah) { pindah satu
              langkah (satu sel)
              sesuai arah tersebut }
       if SolveMaze(M) then
         return true
       else
          unmove(M, arah) { backtrack }

       endif
   endfor
   return false { semua arah sudah
       dicoba, tetapi tetap buntu,
       maka
       kesimpulannya: tidak ada solusi }
endif
```

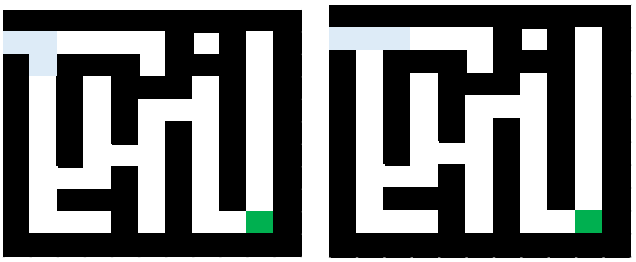
Pada algoritma backtracking di atas, jika pemanggilan rekursif ke SolveMaze (M) bernilai true, berarti perpindahan yang dilakukan mengarah ke solusi. Oleh karena itu, gambar labirin yang dihasilkan ditampilkan ke layar. Gerakan langkah yang dihasilkan dicetak dalam urutan terbalik saat panggilan rekursif kembali, menyebabkan masalah kecil. Untuk memperbaikinya, perlu menyimpan gerakan langkah pada tumpukan dan mencetak seluruh langkah setelah panggilan SolveMaze alih-alih langsung mencetaknya.

Algoritma backtracking untuk masalah Maze (labirin) dapat dianggap sebagai pembentukan pohon state-space. Akar pohon adalah labirin pertama, dan anak-anaknya adalah labirin yang dihasilkan dari pergerakan satu langkah dari labirin semula.

Contoh pencarian solusi yang dapat dilakukan sebagai salah satu penerapan algoritma backtracking.

Gambar 6. Tampilan awal Maze

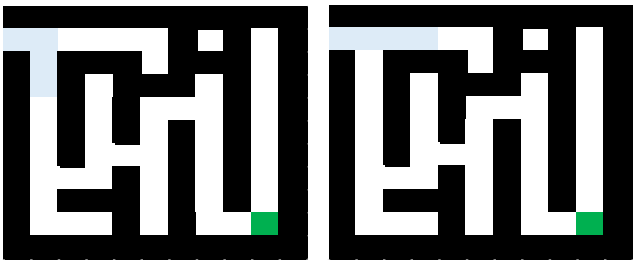
a. Dari titik awal, program akan memilih titik terdekat yang bisa dikunjungi atau tidak terhalang tembok. Dalam hal ini, hanya dapat bergerak ke atas, belok kanan, belok kiri, dan turun.



(i) (ii)

Gambar 7. (i) Program bergerak ke bawah dan (ii) Program bergerak ke kanan

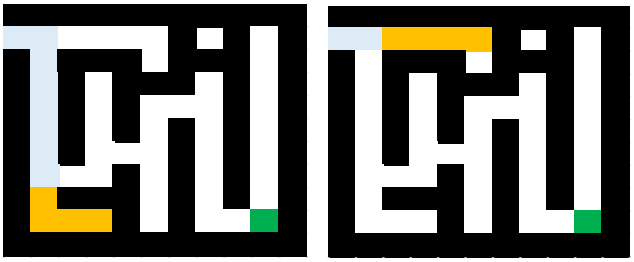
b. Jika titik yang dikunjungi adalah titik yang belum pernah dikunjungi sebelumnya, program akan terus bergeser dari titik tersebut ke dalam tumpukan solusi dan menandai titik yang sudah dilewati untuk menunjukkan bahwa posisinya telah lewat, lalu lanjut lagi ke titik berikutnya.



(i) (ii)

Gambar 8. (i) Program bergeser ke titik berikutnya ke bawah ; (ii) Program bergeser ke titik berikutnya ke kanan

c. Jika titik yang dikunjungi adalah jalan buntu atau menabrak dinding, program akan mundur kembali ke posisi sebelumnya.

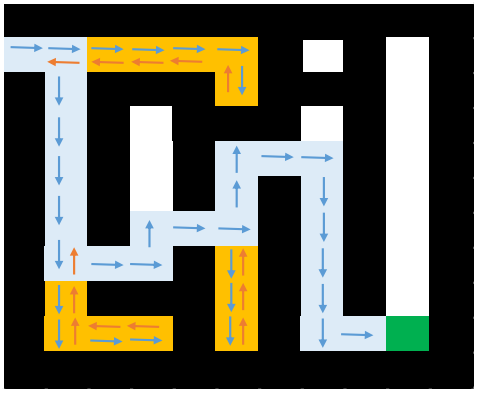


(i) (ii)

Gambar 9. (i) Program mundur ke posisi sebelumnya ke atas; (ii) Program mundur ke posisi sebelumnya ke kiri

d. Program akan mengulang langkah b dan c sampai ditemukan solusi titik keluar dari maze. Jika solusi tidak ditemukan maka, program akan menampilkan pesan bahwa solusi tidak ditemukan.

Gambar 10. Solusi Akhir Maze

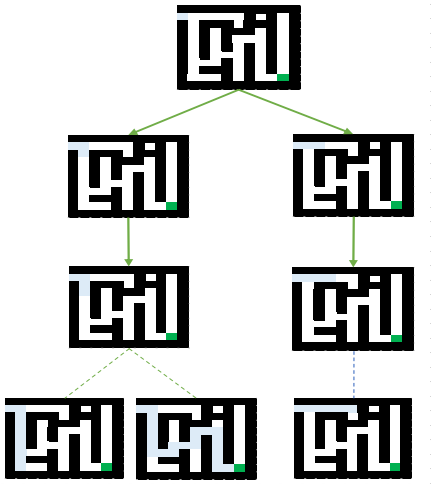


Gambar 11. Backtracking pada Maze (labirin)

Gambar 9 menunjukkan lintasan yang benar yang dilalui di dalam maze (labirin) pada gambar 4. Sedangkan

Gambar 10 menunjukkan lintasan yang dibentuk dengan algoritma backtracking (runut-balik) pada sebuah maze (labirin). Backtracking diperlihatkan dengan tanda panah merah.

Node-node pada pohon dibentuk dengan skema traversal DFS. Node daun menunjukkan apakah status gerak panggung telah mencapai jalan buntu atau jalan keluar telah ditemukan.



Gambar 12. Sebagian pohon ruang status pada persoalan Maze gambar 4

Ketika gerakan langkah mencapai jalan buntu, simpul daun yang mewakili keadaan labirin dialihkan ke jalan buntu, dan jalur pencarian ditelusuri kembali ke simpul di atas. Pohon ruang keadaan diperluas sampai solusi ditemukan. Atau tidak ada solusi.

Penggunaan algoritma backtracking ini dapat dilihat dengan mengikuti setiap jalur untuk mencapai tujuan yang diinginkan. Saat komputer memulai permainan, komputer memutuskan jalur mana yang harus diikuti setiap jalur. Ketika komputer menemukan jalan buntu, ia kembali ke jalur sebelumnya dan menjalankan proses lacak balik hingga menemukan jalur baru yang belum pernah diambilnya.

Kemampuan algoritma untuk memecahkan masalah permainan maze (labirin) menunjukkan bahwa algoritma backtracking sangat efektif dalam mencari solusi masalah. Kerja sistematis dari algoritma backtracking dan karakteristiknya dalam memeriksa hanya solusi yang mungkin yang benar-benar dapat dianggap sebagai solusi akhir dievaluasi sebagai solusi yang efektif dan efisien untuk masalah ini.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari analisis perancangan serta hasil implementasi program aplikasi yang dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Penerapan algoritma backtracking pada maze game berhasil dibuat dengan menggunakan bahasa pemrograman python.
2. Algoritma backtracking dapat diimplementasikan kedalam maze game untuk menemukan jalan keluar dari maze (labirin).
3. Algoritma Backtracking memiliki pencarian yang baik karena memilki fungsi pembatas, sehingga ketika

pencarian itu dianggap tidak akan menemukan solusi lebih baik maka pencari diberhentikan.

4. Algoritma Backtracking memeriksa solusi yang dipertimbangkan saja, tanpa harus memriksa semua kemungkinan solusi yang ada Algoritma backtracking sangat efektif sehingga banyak diterapkan dalam banyak persoalan game dan dalam bidang kecerdasan buatan lainnya.

B. Saran

Berdasarkan hasil yang diperoleh maka saran untuk meningkatkan pemahaman pada penerapan algoritma backtracking adalah sebagai berikut:

1. Menambahkan fungsionalitas yang baru, seperti terdapatnya tingkatan permainan.
2. Membandingkan algoritma backtracking dengan algoritma lainnya untuk mengetahui optimasi algoritma ini sebagai kecerdasan buatan pada permainan dam-daman.

VIDEO LINK AT YOUTUBE

https://youtu.be/Xkb4e3KVGvE

REFERENCES

[1] *Maze game*. Welcome to RMS Technology Classes. (n.d.). Retrieved December 31, 2021, from http://rmstechclass.weebly.com/maze-game.html

[2] Ilmuskripsi. (2019, March 27). *Algoritma runut-balik (backtracking)*. Skripsi Teknik Informatika. Retrieved December 31, 2021, from https://www.ilmuskripsi.com/2016/05/algoritma-runut-balik-backtracking.html

[3] *Backtracking - university of illinois urbana-champaign*. (n.d.). Retrieved December 31, 2021, from https://jeffe.cs.illinois.edu/teaching/algorithms/book/02-backtracking.pdf

[4] *Mencari Jalan Keluar di Dalam Labirin (maze problem).* PDF Free Download. (n.d.). Retrieved December 31, 2021, from https://docplayer.info/36455042-Mencari-jalan-keluar-di-dalam-labirin-maze-problem.html

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Laguboti, 30 Desember 2021

Meyliza Veronica Siregar - 11419058