Penerapan Algoritma Backtracking dalam Permainan Tents and Trees

Andrew 13519036
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519036@std.stei.itb.ac.id

***Abstraksi*—Algoritma Backtracking adalah salah satu algoritma yang cukup populer digunakan untuk menemukan solusi dari suatu permasalahan. Algoritma ini adalah bentuk perbaikan dari exhaustive search, sehingga dapat mengurangi waktu pencarian solusi permasalahan. Algoritma Backtracking dapat diterapkan dalam salah satu permainan puzzle, yaitu Tents and Trees.**

Kata kunci—algoritma; backtracking; Tents and Trees

I. PENDAHULUAN

Algoritma memiliki peranan yang sangat penting dalam menyelesaikan suatu permasalahan secara efektif dan efisien. Dalam menyelesaikan suatu permasalahan, kita perlu memilih algoritma apa yang paling efektif dan efisien dalam menyelesaikan permasalahan tersebut. Konsep pemilihan algoritma ini dinamakan strategi algoritma. Beberapa strategi algoritma yang cukup populer adalah Brute Force, Greedy, Divide and Conquer, Decrease and Conquer, Breadth First Search, Depth First Search, Backtracking, Branch and Bound, A-Star, Pattern Matching, Dynamic Programming, dan berbagai algoritma lainnya.

Salah satu algoritma yang digunakan untuk menemukan solusi atau penyelesaian dari suatu permasalahan adalah algoritma Backtracking. Algoritma Backtracking dapat diterapkan dalam menyelesaikan beberapa permasalahan, yaitu kuda di atas papan catur, permasalahan N-ratu, permasalahan sum of subsets, pewarnaan graf, permasalahan integer 1/0 knapsack, dan lainnya.

Puzzle adalah sebuah permainan atau permasalahan yang dibuat untuk mengasah pemikiran. Puzzle biasanya disajikan dengan kesulitan yang beragam dan membutuhkan kesabaran yang beragam pula [1]. Salah satu puzzle yang cukup menarik namun kurang terkanal adalah Tents and Trees. Dalam makalah ini, penulis akan menyajikan bagaimana algoritma Backtracking dapat digunakan untuk menyelesaikan permainan Tents and Trees.

II. LANDASAN TEORI

A. Algoritma Backtracking

Algoritma Backtracking adalah salah satu algoritma yang digunakan untuk menemukan solusi dari suatu permasalahan. Algoritma ini adalah perbaikan dari algoritma Exhaustive Search [2]. Pada algoritma Exhaustive Search, semua kemungkinan akan dieksplorasi, sedangkan pada algoritma Backtracking, Hanya kemungkinan yang mengarah ke solusi yang akan dieksplorasi. Semua kemungkinan yang tidak mengarah ke solusi tidak akan dipertimbangkan lagi. Hal ini dinamakan pruning atau memotong simpul yang tidak mengarah ke solusi.

Algoritma Backtracking memiliki beberapa properti umum, yaitu:

a. Solusi Persoalan, biasanya dinyatakan dalam bentuk vektor dengan n-tuple.

b. Fungsi Pembangkit, dinyatakan sebagai sebuah predikat untuk membangkitkan nilai solusi.

c. Fungsi Pembatas, dinyatakan sebagai sebuah predikat untuk memeriksa apakah kemungkinan mengarah ke solusi.

Semua kemungkinan solusi dari suatu permasalahan disebut ruang solusi. Ruang solusi biasanya diorganisasikan dalam struktur pohon berakar. Tiap simpul pada pohon menyatakan keadaan/status permasalahan, sedangkan sisi/cabang menyatakan nilai-nilai pada solusi persoalan. Lintasan dari akar ke daun akan menyatakan solusi yang mungkin dan seluruh lintasan dari akar ke daun akan menyatakan ruang solusi. Pengorganisasian pohon ruang solusi diacu sebagai pohon ruang status [2].

Dalam mencari solusi permasalahan menggunakan algoritma Backtracking, terdapat tiga istilah simpul yang digunakan, yaitu:

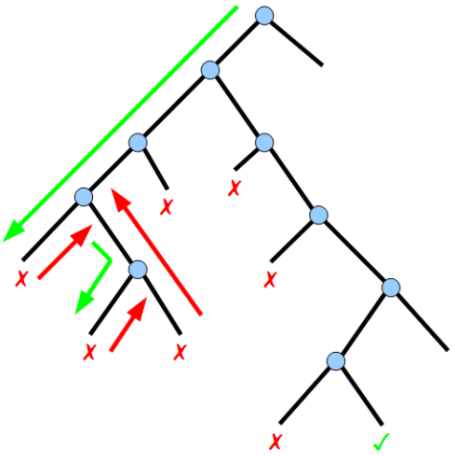
a. Simpul ekspan, yaitu simpul yang sedang diperluas.

b. Simpul hidup, yaitu simpul yang sudah dibangkitkan.

c. Simpul mati, yaitu simpul yang tidak mengarah ke solusi, sehingga simpul tersebut “dimatikan”.

Aturan pembangkitan simpul yang dipakai mengikuti aturan Depth First Search (DFS). Tiap kali simpul ekspan diperluas, lintasan yang dibangun juga bertambah panjang. Jika lintasan tersebut tidak mengarah ke solusi, maka simpul ekspan tersebut akan “dimatikan” sehingga menjadi simpul mati. Fungsi yang digunakan untuk “mematikan” simpul ekspan adalah dengan menerapkan fungsi pembatas. Ketika sebuah simpul “dimatikan”, secara implisit kita telah memangkas

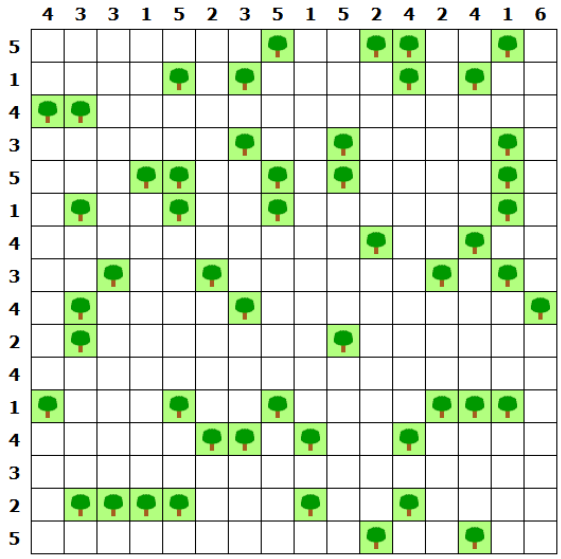
simpul-simpul anaknya. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian akan backtrack ke simpul pada tingkat di atasnya. Proses pencarian dihentikan apabila kita telah sampai pada simpul solusi (goal node).



Gambar 1 Skema Algoritma Backtracking

B. *Tents and Trees*

Tents and Trees adalah permainan puzzle yang terdiri dari tenda dan pohon. Tidak ada sumber pasti yang menyebutkan kapan permainan ini diciptakan.



Gambar 2 Permainan Tents and Trees

Permainan ini memiliki beberapa atribut, yaitu pohon, tenda dan rumput, kotak kosong, dan angka di pinggir puzzle. Pohon mengindikasikan permasalahan, tenda dan rumput mengindikasikan jawaban dari permasalahan, kotak kosong mengindikasikan tempat kita meletakkan tenda dan rumput, dan angka di pinggir puzzle mengindikasikan jawaban dari permasalahan.

Aturan dari permainan ini adalah sebagai berikut.

a. Setiap tenda diletakkan di samping pohon
b. Setiap tenda diletakkan sesuai dengan jumlah angka yang berkorespondensi terhadap baris atau kolomnya.
c. Setiap tenda tidak boleh berdekatan, baik secara horizontal, vertikal, maupun diagonal.

III. Hasil dan Pembahasan

Algoritma Backtracking dapat diterapkan dalam permainan Tents of Trees. Pada bagian ini, penulis telah membuat sebuah program sederhana untuk menyelesaikan puzzle Tents and Trees.

A. *Mapping Properti Algoritma Backtracking*

Berikut adalah mapping komponen-komponen dalam permainan Tents and Trees ke dalam properti Algoritma Backtracking.

a. Solusi persoalan pada permainan Tents and Trees adalah

$$S = \{T,G\}$$

dengan T adalah simbol untuk tenda dan G adalah simpul untuk rumput.

b. Fungsi pembatas pada permainan Tents and Trees adalah aturan-aturan pada permainan Tents and Trees. Dalam hal ini, aturan-aturan yang akan diperiksa adalah batasan jumlah tenda yang dapat diisi dan batasan kedekatan tenda.

B. *Konsep Umum*

Konsep umum penerapan algoritma Backtracking dalam permainan Tents and Trees adalah:

a. Mencari kotak yang masih kosong.
b. Memeriksa jika kotak tersebut dapat diisi oleh tenda. Jika kotak tersebut dapat diisi oleh tenda, buat asumsi awal bahwa kotak tersebut sudah terisi oleh tenda, lalu kurangi angka pada baris dan kolom yang berkorespondensi dengan 1.
c. Jika pada akhirnya asumsi awal tersebut tidak menghasilkan solusi, maka asumsi tersebut akan di-*backtrack* dan kotak tersebut akan diisi dengan rumput.
d. Ulangi langkah a-c sampai semua kotak terisi.

C. *Beberapa Fungsi dan Prosedur dalam Program*

a. GetPuzzle(filename)
Fungsi GetPuzzle menerima sebuah masukan nama file .txt dan mengembalikan matriks puzzle tersebut.
b. PrintPuzzle(puzzle)
Prosedur PrintPuzzle digunakan untuk menampilkan puzzle ke layar
c. IsValidCoordinate(i, j, N)
Fungsi IsValidCoordinate mengembalikan True jika koordinat (i,j) adalah koordinat puzzle berukuran NxN yang valid.

d. IsTileTentCandidate(puzzle, i, j)
   Fungsi IsTileTentCandidate mengembalikan True jika sebuah kotak kosong merupakan kandidat kotak yang dapat diisi dengan tenda.
e. IsTileTentArea(puzzle, i, j)
   Fungsi IsTileTentArea mengembalikan True jika sebuah kotak kosong berada dekat dengan suatu tenda.
f. MarkGrass(puzzle)
   Prosedur MarkGrass menandai semua kotak kosong yang tidak mungkin diisi dengan tenda dengan rumput.
g. FindEmptyTile(puzzle, coord)
   Fungsi FindEmptyTile mengembalikan True jika ditemukan kotak kosong pada puzzle, kemudian mencatat koordinat kotak kosong tersebut ke dalam coord.
h. IsSafe(puzzle, i, j)
   Fungsi IsSafe mengembalikan True jika kotak kosong pada koordinat (i,j) memenuhi aturan permainan saat kotak tersebut diisi dengan tenda.
i. VerifySolve(puzzle)
   Fungsi VerifySolve mengembalikan puzzle jika puzzle tersebut sudah memenuhi seluruh aturan pada permainan Tents and Trees. Jika tidak memenuhi, maka akan mengembalikan array kosong.
j. SolvePuzzle(puzzle)
   Fungsi SolvePuzzle adalah fungsi rekursif yang menerapkan algoritma Backtracking. Fungsi ini akan mengembalikan puzzle yang sudah terselesaikan atau array kosong.

D. Beberapa Keterangan dalam Program

Berikut adalah beberapa simbol yang digunakan dalam program.

a. “P” menandakan pohon
b. “T” menandakan tenda
c. “G” menandakan rumput
d. “.” menandakan kotak kosong

Program akan membaca sebuah file .txt yang berisi puzzle yang sudah direpresentasikan dengan simbol-simbol yang telah didefinisikan. Berikut adalah contoh representasi puzzle Tents and Trees 8x8 ke dalam file .txt.

```
 21212112
1.P......
2.P..P...
1....P..P
1....P...
1P.......
3.....P..
1..PP....
2..P...P.
```

Gambar 3 Contoh representasi Puzzle dalam file .txt

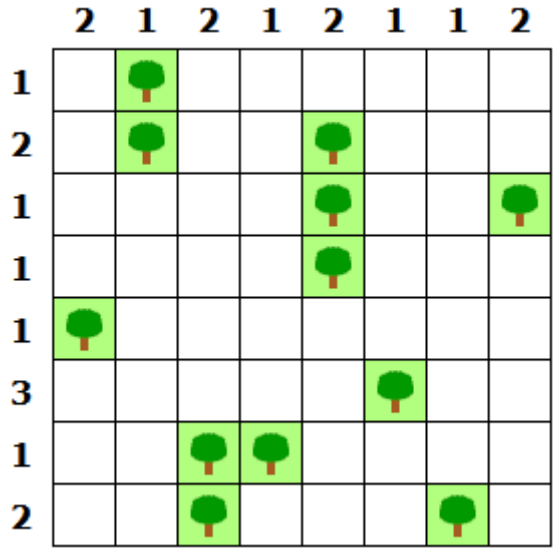
Setelah menemukan solusi dari sebuah puzzle Tents and Trees, program akan menampilkan *side-by-side comparison* puzzle mula-mula dan solusi puzzle. Pada mulanya, rumput pada solusi puzzle memiliki label simbol “G”. Simbol ini akan diubah menjadi “.” untuk memudahkan perbandingan puzzle sebelum dan sesudah ditemukan solusi. Berikut adalah contoh keluaran program.

```
E:\Coding\Makalah>python Puzzle.py
 21212112    00000000
1.P......   0.PT.....
2.P..P...   0TP..PT..
1....P..P   0...TP..P
1....P...   0....P..T
1P.......   0P...T...
3.....P..   0T.T..PT.
1..PP....   0..PPT...
2..P...P.   0.TP...PT
```

Gambar 4 Contoh keluaran program Puzzle.py

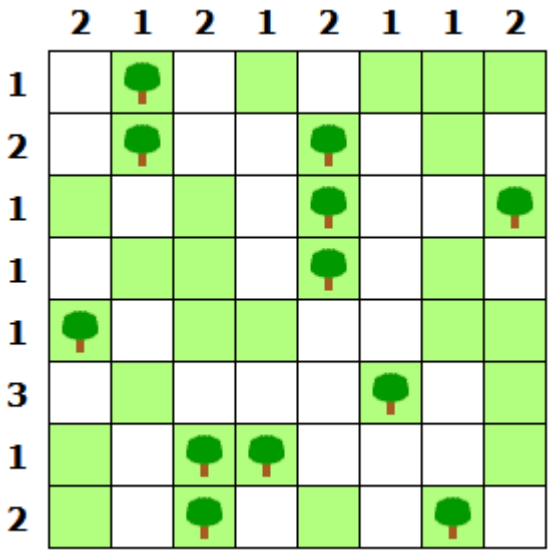
E. Konsep Algoritma Backtracking dalam Menyelesaikan Puzzle Tents and Trees

Berikut adalah salah satu puzzle Tents and Trees berukuran 8x8.



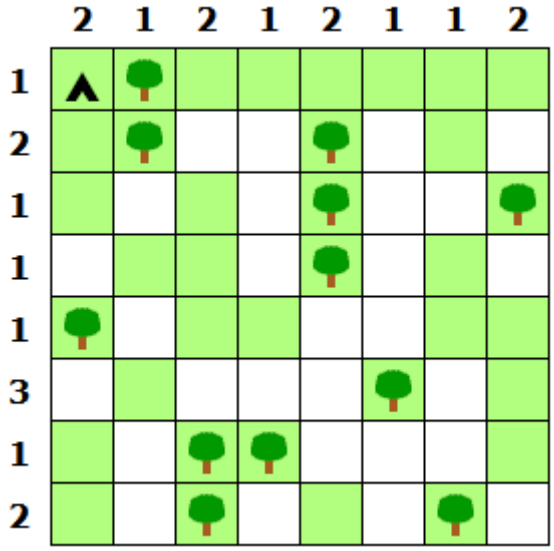
Gambar 5 Puzzle Tents and Trees 8x8

Untuk memudahkan penyelesaian dengan menggunakan algoritma Backtracking, pertama-tama kita akan menandai kotak-kotak kosong yang sudah pasti tidak akan diisi dengan tenda. Hasil dari penandaan awal adalah sebagai berikut.



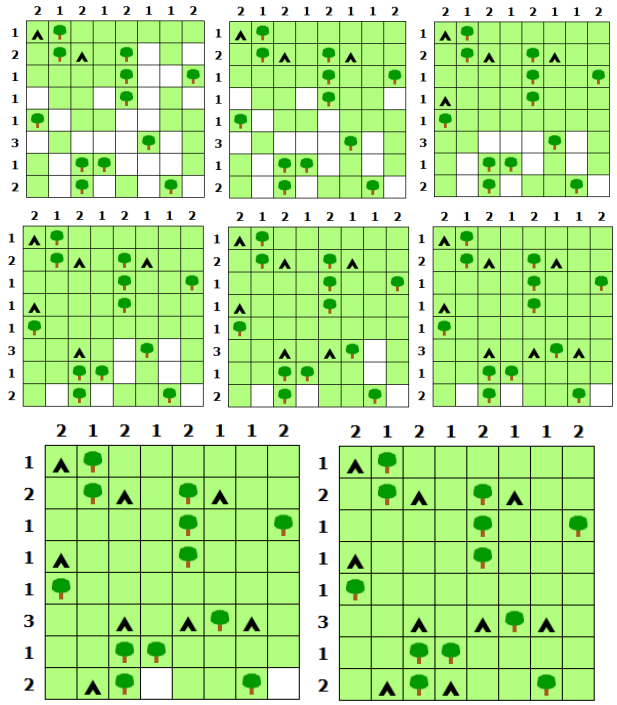
Gambar 6 Hasil penandaan awal pada puzzle

Kemudian, akan dicari koordinat pertama dimana kotak pada koordinat tersebut merupakan koordinat kosong. Dalam hal ini, koordinat (1,1) adalah koordinat yang pertama ditemukan. Selanjutnya, akan diperiksa apakah koordinat (1,1) merupakan tempat yang aman untuk meletakkan tenda. Dalam hal ini, koordinat (1,1) merupakan tempat yang aman. Maka, kita akan membuat asumsi awal bahwa kotak (1,1) sudah diisi dengan tenda. Setelah pengisian tenda ke kotak (1,1), kita akan menandai Kembali kotak-kotak kosong yang sudah pasti tidak akan diisi dengan tenda. Berikut adalah hasil dari peletakan tenda dan penandaan setelahnya.



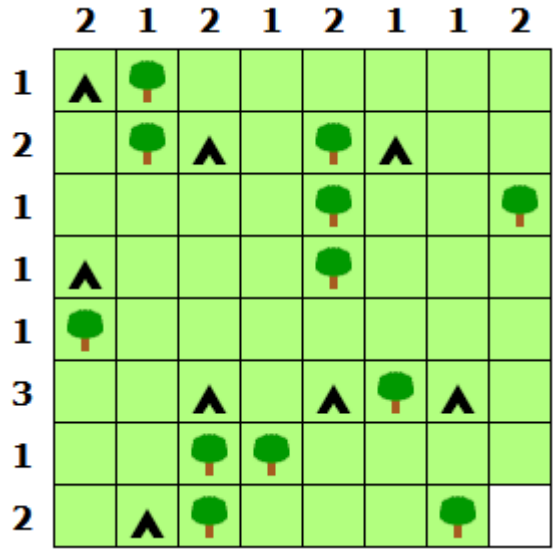
Gambar 7 Hasil pembuatan asumsi awal pada puzzle

Langkah di atas akan diulangi sampai tidak ada kotak kosong yang dapat ditemukan. Berikut adalah urutan semua hasil yang dibangun dari asumsi awal.



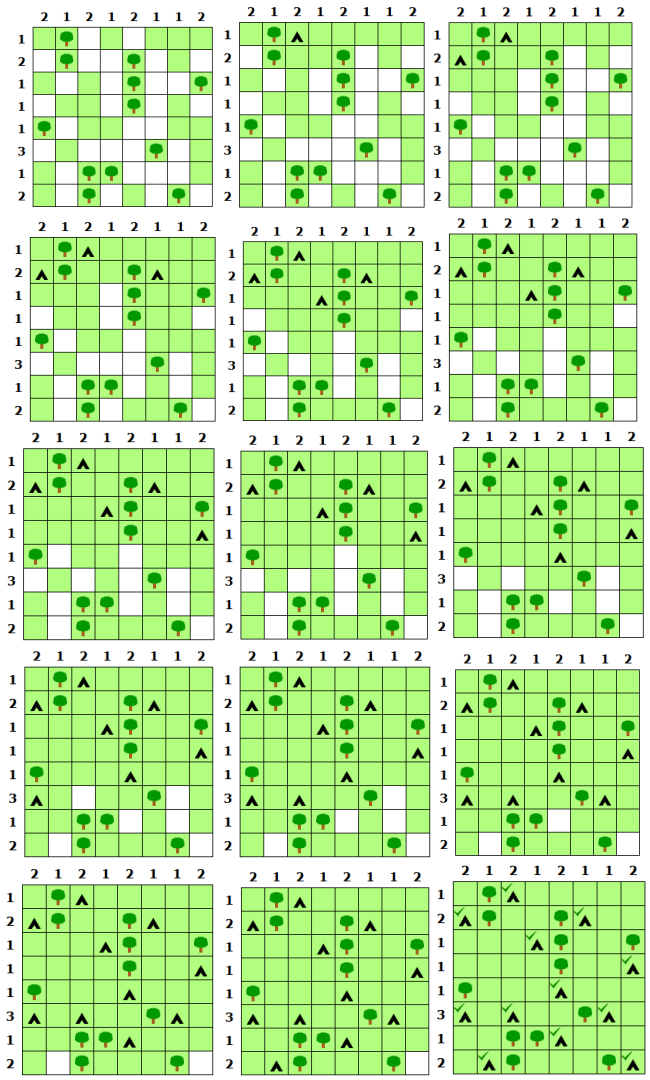
Gambar 8 Hasil lintasan menuju simpul daun

Dapat diperhatikan bahwa simpul daun yang dihasilkan tidak memenuhi aturan jumlah tenda pada beberapa baris dan kolom. Dalam hal ini, akan dilakukan backtracking ke simpul sebelumnya, dan kotak yang di-backtrack akan diisi dengan rumput.



Gambar 9 Hasil backtracking ke simpul sebelumnya

Proses pencarian dan backtracking akan dilakukan sampai ditemukan penyelesaian dari puzzle tersebut. Berikut adalah lintasan dari simpul akar ke simpul daun yang merupakan penyelesaian dari puzzle ini.



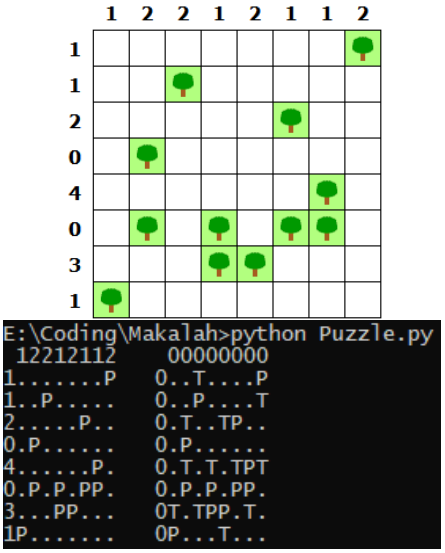
Gambar 10 Lintasan penyelesaian Puzzle Tents and Trees

Solusi permasalahan pada puzzle ini dapat dinyatakan sebagai vektor solusi berikut.

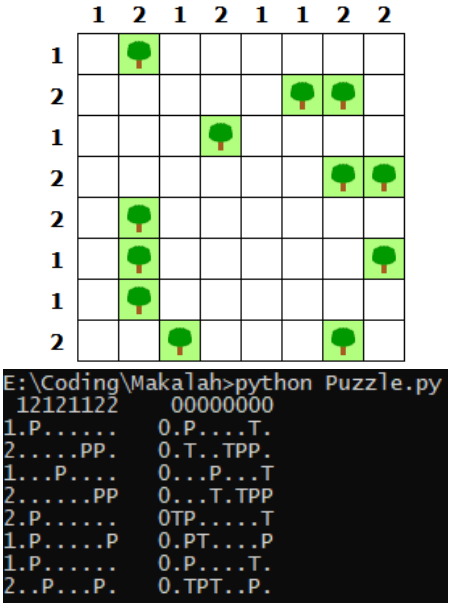
X = (G, T, T, T, T, G, T, G, T, T, T, T, T, T, T)

F. Hasil Pengujian Program

Berikut adalah beberapa cuplikan program dalam menyelesaikan beberapa puzzle Tents and Trees.



Gambar 11 Hasil penyelesaian program



Gambar 12 Hasil penyelesaian program

IV. KESIMPULAN

Algoritma Backtracking adalah salah satu algoritma yang cukup efektif dan efisien dalam menemukan solusi dari suatu permasalahan. Algoritma Backtracking dapat diterapkan untuk menyelesaikan puzzle Tents and Trees. Dalam hal ini, kita dapat memahami bagaimana algoritma Backtracking bekerja dan bagaimana penerapannya dalam menyelesaikan puzzle Tents and Trees.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas selesainya pembuatan makalah ini. Penulis juga mengucapkan terima kasih kepada setiap dosen pengampu

mata kuliah IF2211 Strategi Algoritma , terutama kepada Dr. Ir. Rinaldi Munir, M.T. atas penyediaan website kuliah dan video kuliah online, dan kepada Ir. Rila Mandala, M.Eng., Ph.D. atas bimbingan beliau dalam mengajar K01. Tak lupa penulis mengucapkan terima kasih kepada setiap teman-teman yang telah memberikan saran dan masukan kepada penulis.

LINK REPOSITORY GITHUB

https://github.com/andrcyes/Tents-and-trees

REFERENSI

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Makalah/Makalah-Matdis-2020%20(34).pdf , diakses pada tanggal 10 Mei 2021

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-backtracking-2021-Bagian1.pdf , diakses pada tanggal 10 Mei 2021

[3] https://www.brainbashers.com/tents.asp , diakses pada tanggal 10 Mei 2021

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.

Pekanbaru, 11 Mei 2021

Andrew 13519036