

# Penggunaan Algoritma Backtracking Pada Permainan ABC Path

Mahesa Gandakusuma (13513091)  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
ezza3224@students.itb.ac.id

**Abstract**—Makalah ini akan membahas tentang ABC Path dan cara penyelesaiannya. ABC Path adalah salah satu permainan yang membutuhkan logika untuk menyelesaikannya. Salah satu cara penyelesaian permainan ABC Path yang akan dibahas dalam makalah ini adalah dengan backtracking. Backtracking atau runut balik adalah sebuah algoritma yang digunakan salah satunya untuk menyelesaikan puzzle berbentuk maze.

**Kata Kunci**—ABC Path, backtracking, permainan logika, problem solving, puzzle, runut balik

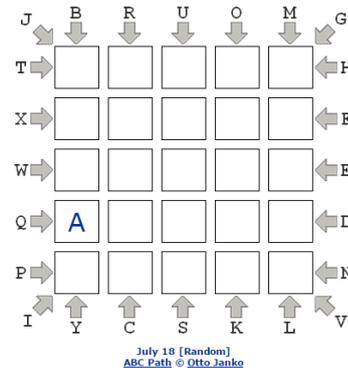
## I. PENDAHULUAN

Saat ini terdapat banyak jenis permainan yang tersebar dalam berbagai kategori, seperti arcade, puzzle, strategi, petualangan, simulator, dan lain-lain. Permainan puzzle adalah suatu permainan yang mempunyai minimal sebuah solusi untuk diselesaikan, dengan mengikuti aturan dalam permainan tersebut. Seseorang dapat dikatakan menang apabila solusi dalam permainan tersebut tercapai. Dalam suatu permainan puzzle biasanya digunakan logika dalam proses penyelesaiannya, sehingga permainan tipe ini sering juga disebut permainan logika. Setiap permainan logika memiliki logika yang berbeda-beda untuk menyelesaikannya, oleh karena itu diperlukan sebuah strategi atau teknik dasar tertentu untuk menyelesaikannya.

Saat ini permainan logika telah beragam macamnya. Contoh permainan logika adalah ABC path, sudoku, lights on, futoshiki, kakurasu, nonogram, battleship, dan lain-lain. Masing-masing permainan logika ini memiliki bentuk dan aturan yang unik dan variatif antara satu macam permainan dengan yang lain. Salah satu permainan logika yang akan dibahas dalam makalah ini adalah ABC Path.

## II. ABC PATH

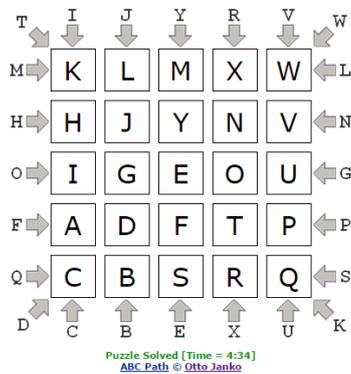
ABC Path adalah sebuah permainan logika yang terdiri dari sebuah grid berukuran 5x5. Setiap baris, kolom, dan diagonalnya terdapat huruf-huruf yang menandakan huruf tersebut ada dalam baris, kotak, atau diagonal tersebut sebagai clue. Salah satu kotak dalam grid tersebut berisi huruf A sebagai awal huruf dari ABC path. Kotak-kotak yang lainnya pada mulanya kosong, yang nantinya akan diisi dengan huruf-huruf sesuai clue yang berada di pinggir grid. Semua huruf clue yang berada di pinggir grid mengandung salah satu huruf antara B sampai dengan Y dan dipastikan tidak ada yang sama satu sama lain.



Gambar 1 Sebuah puzzle ABC path.

(Sumber: <http://www.brainbashers.com/showabcpath.asp/>)

Aturan permainan dari ABC path adalah sebagai berikut. Mulai dari huruf A, isilah huruf berikutnya berurutan sesuai dengan clue yang ada di pinggir grid yang menunjukkan baris, kolom, atau diagonal di mana huruf tersebut harus berada. Setiap huruf tersebut harus berada di tepat sebelah huruf sebelumnya secara horizontal, vertikal, atau diagonal. Sebuah kotak hanya boleh diisi sebuah huruf. Sebuah ABC path dinyatakan telah selesai apabila semua kotak dalam grid telah berisi huruf A sampai Y seperti pada gambar berikut.



Gambar 2 Sebuah ABC path dari Gambar 1 yang telah diselesaikan

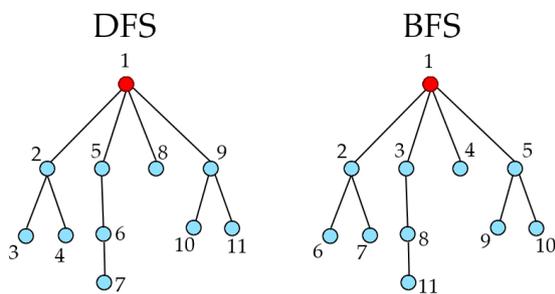
(Sumber: <http://www.brainbashers.com/showabcpath.asp/>)

### III. DASAR TEORI

#### A. Algoritma BFS dan DFS

Algoritma BFS dan DFS digunakan dalam penelusuran sebuah graf. Algoritma Breadth First Search (BFS) adalah teknik penelusuran graf secara melebar, sedangkan algoritma Depth First Search (DFS) adalah teknik penelusuran graf secara mendalam. Algoritma ini bersifat blind-search, artinya tidak ada informasi tambahan yang dapat digunakan untuk mempermudah pencarian.

Pada gambar berikut, akan dijelaskan bagaimana proses penelusuran graf dengan algoritma BFS dan DFS. Simpul awal dari tiap graf dinyatakan dengan simpul berwarna merah, dan setiap nomor di dekat setiap simpul menunjukkan urutan pencarian simpul dalam graf pada tiap algoritma.



Gambar 3 Skema Penelusuran Graf dengan algoritma BFS dan DFS

Graf yang dapat ditelusuri menggunakan algoritma BFS dan DFS dapat berupa graf statis atau graf dinamis. Graf statis adalah graf yang terbentuk sebelum dilakukan pencarian dan tidak berubah-ubah selama pencarian berlangsung. Graf dinamis adalah graf yang terbentuk saat proses pencarian dilakukan. Graf dinamis dapat berubah-ubah sesuai simpul dalam graf yang dikunjungi.

Untuk mengecek apakah sebuah algoritma pencarian lebih baik daripada yang lain, digunakan evaluasi teknik pencarian. Evaluasi ini terdiri dari 4 faktor, yaitu:

- **Completeness** : apakah algoritma yang dipakai dapat menjamin ditemukannya solusi jika memang ada solusinya
- **Optimality** : apakah teknik yang dipakai dapat menjamin ditemukannya solusi optimal
- **Time complexity** : waktu yang diperlukan untuk mencapai solusi
- **Space complexity** : memory yang digunakan untuk melakukan pencarian

Kompleksitas waktu dan ruang dapat ditentukan dengan variabel berikut:

- **Branching factor (b)** : maksimum percabangan yang mungkin dari satu simpul
- **Depth (d)** : kedalaman dari solusi terbaik (cost terendah)
- **m** : kedalaman maksimum dari ruang status, m bisa saja bernilai tak hingga

Berikut akan dijelaskan perbandingan algoritma BFS dan DFS berdasarkan evaluasi teknik pencarian.

**Completeness:**

- **BFS:** Ya, selama nilai b terbatas
- **DFS:** Ya, selama nilai b terbatas dan ada penanganan 'redundant paths' dan 'repeated state'

**Optimality:**

- **BFS:** Ya, jika langkah = cost
- **DFS:** Tidak

**Kompleksitas waktu:**

- **BFS:**  $1+b+b^2+b^3+\dots+b^d = O(b^d)$
- **DFS:**  $O(b^m)$

**Kompleksitas ruang:**

- **BFS:**  $O(b^d)$
- **DFS:**  $O(bm)$

Perbandingan kompleksitas waktu dengan kompleksitas ruang:

- **BFS:** Kurang baik dalam kompleksitas ruang
- **DFS:** Kurang baik dalam kompleksitas waktu

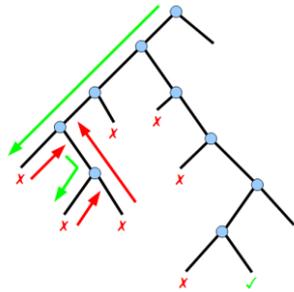
#### B. Algoritma Backtracking

Algoritma Backtracking adalah sebuah pengembangan dari algoritma DFS. Pada algoritma backtracking, graf yang terbentuk berupa sebuah pohon dimana daunnya adalah sebuah titik backtrack (dead end) atau goal.

Algoritma backtracking ini adalah sebuah algoritma perbaikan dari brute force. Pada brute force, semua kemungkinan solusi dicek satu persatu, sedangkan pada backtracking, hanya kemungkinan yang mengarah ke solusi saja yang ditelusuri.

Prinsip pencarian solusi menggunakan algoritma backtracking adalah sebagai berikut. Solusi dicari dengan membentuk lintasan dari akar ke daun dengan metode DFS. Simpul yang sudah dibangkitkan disebut simpul

hidup, dan simpul yang diperluas disebut simpul expand. Lintasan akan bertambah panjang tiap kali simpul expand diperluas. Jika lintasan tersebut tidak mengarah ke solusi (dead end), simpul tersebut di-'bunuh' menjadi simpul mati. Fungsi untuk mem-'bunuh' simpul yang menuju dead end adalah fungsi pembatas (bound). Simpul yang mati tidak dapat diperluas lagi. Jika terjadi dead end, lakukan proses backtrack ke simpul atasnya, kemudian membangkitkan simpul anak lainnya. Proses ini berulang terus sampai ada simpul yang mengarah ke goal node sebagai solusi penyelesaian.



Gambar 4 Sebuah ilustrasi backtracking  
(Sumber: <http://www.w3.org/2011/Talks/01-14-steven-phenotype/>)

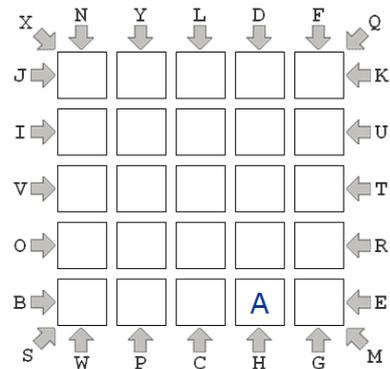
#### IV. ANALISIS DAN PEMBAHASAN

ABC Path dapat disamakan jenisnya dengan puzzle maze yang lain, di mana ABC Path mencari solusi dengan mengisi huruf A sampai dengan Y. Oleh karena itu, ABC path dapat diselesaikan dengan backtracking.

Salah satu metode yang dapat digunakan dalam menyelesaikan ABC path adalah dengan backtracking. Dalam menyelesaikan ABC path dengan metode backtracking, backtrack dilakukan apabila tidak ada tempat bagi huruf selanjutnya atau huruf selanjutnya tidak dapat diletakkan tepat bersebelahan dengan huruf yang terakhir ditulis di sebuah kotak.

Untuk tiap huruf dimulai dari A, huruf selanjutnya dapat ditempatkan di beberapa tempat yang sesuai dengan clue. Jika tidak ada tempat, lakukan backtracking dengan memindahkan huruf sebelumnya ke kemungkinan tempat lain yang belum dicoba. Jika semua kemungkinan tempat telah dicoba dan semuanya menuju dead end, lakukan backtracking.

Sebagai contoh, ABC path berikut ingin diselesaikan dengan metode backtracking.

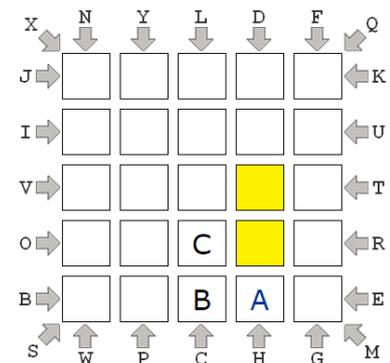


Gambar 5 Sebuah ABC Path yang ingin diselesaikan  
(Sumber: <http://www.brainbashers.com/showabcpath.asp/>)

Peletakan suatu huruf selalu berdasarkan huruf sebelumnya dan clue letak huruf tersebut berada. Pada Gambar 5, huruf B hanya dapat diletakkan di sebelah kiri A atau sebelah kanannya, karena clue menunjukkan B berada pada baris kelima. Terbentuklah 2 cabang pada graf penyelesaian; salah satu cabang menunjukkan huruf B diletakkan di sebelah kiri A dan cabang lainnya menunjukkan huruf B diletakkan di sebelah kanan A.

Mulailah dengan menelusuri cabang pertama. Setelah meletakkan huruf B di sebelah kiri huruf A, cek clue untuk menentukan di mana huruf C harus diletakkan. Ternyata clue menunjukkan bahwa huruf C berada di kolom ketiga. Tempat yang mungkin hanya di atas huruf B. Letakkan huruf C di sana.

Selanjutnya adalah pengisian huruf D. Ternyata huruf D berada di kolom keempat berdasarkan clue. Peletakan yang mungkin adalah D berada di sebelah kanan atas huruf C atau di sebelah kanan huruf C, seperti pada gambar 6 di bawah. Huruf D tidak dapat diletakkan pada sebelah kanan bawah C, karena telah terdapat huruf A di sana.

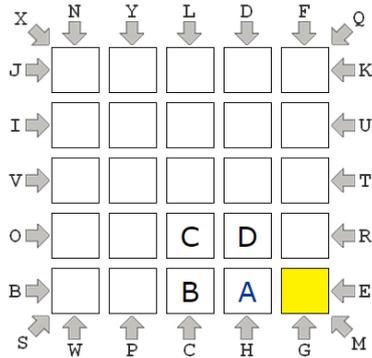


Gambar 6 Kemungkinan tempat untuk huruf D, ditandai dengan kotak kuning

Mulai dengan menelusuri cabang pertama, yaitu cabang yang menunjukkan huruf D di sebelah kanan atas huruf C. Setelah itu, cek di mana huruf E harus diletakkan. Ternyata huruf E berada pada baris kelima.

Huruf D berada pada baris ketiga kolom keempat. Karena huruf E berada pada baris kelima, tidak mungkin

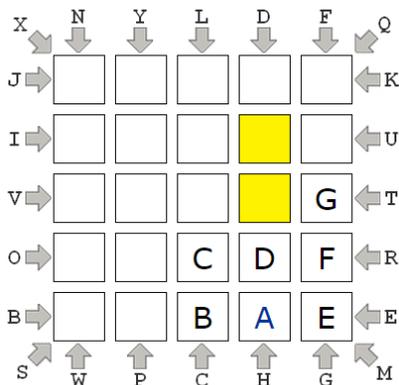
huruf E diletakkan tepat bersebelahan dengan huruf D. Lakukan backtracking dengan memindahkan huruf D ke kemungkinan tempat lain, yaitu menjadi di sebelah kanan huruf C seperti pada Gambar 7.



Gambar 7 Huruf D diletakkan di sebelah kanan huruf C

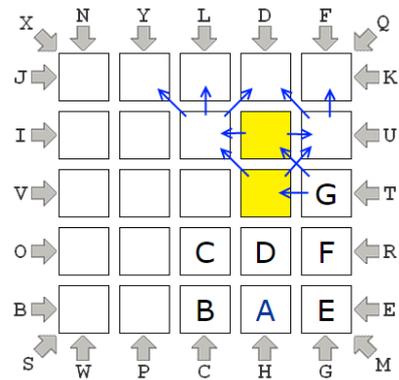
Huruf E sebelumnya telah dicek berada pada baris kelima, sehingga kemungkinan tempatnya hanya pada sebelah kanan bawah huruf D. Tempat lain tidak mungkin karena telah berisi huruf A dan B. Letakkan huruf E di sana.

Kemudian, baik huruf F dan G memiliki satu buah kemungkinan tempat, sehingga langsung dapat diletakkan di tempat yang mungkin. Hasil sementara dapat dilihat pada Gambar 8 berikut.



Gambar 8 Progress sementara penyelesaian ABC Path pada Gambar 5

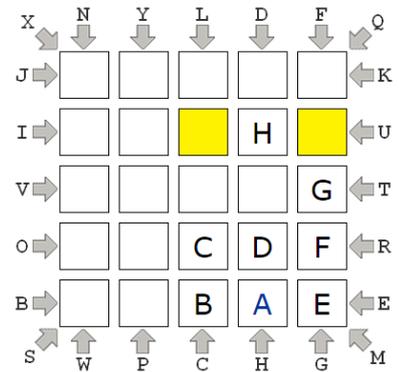
Kemungkinan peletakan huruf H adalah di sebelah kiri atas huruf G atau di sebelah kiri huruf G. Pada langkah selanjutnya konsep backtracking dipakai kembali karena kemungkinan-kemungkinan yang terbentuk untuk peletakan huruf H sampai J adalah seperti gambar 9.



Gambar 9 Kemungkinan peletakan untuk huruf H sampai J

Pada gambar 9, walaupun huruf I memiliki 2 kemungkinan tempat yang sama di manapun huruf H berada, keduanya dianggap sebuah solusi yang berbeda, karena nantinya semua kotak harus terisi dengan huruf-huruf dari A sampai Y.

Mulai dari huruf H dicoba diletakkan di kiri atas huruf G, maka kemungkinan peletakan huruf I adalah di sebelah kiri huruf H atau sebelah kanan huruf H.



Gambar 10 Kemungkinan peletakan huruf I

Lanjutkan dengan mencoba meletakkan huruf I di sebelah kiri huruf H. Maka huruf J harus diletakkan di kiri atas huruf I, di atas huruf I, atau di sebelah kanan atas huruf I. Selanjutnya karena huruf J dan huruf K berada pada baris yang sama berdasarkan clue, huruf K dapat diletakkan di sebelah kiri atau sebelah kanan huruf J.

Kemudian cek peletakan huruf L. Huruf L berada pada kolom ketiga. Yang menarik dalam peletakan huruf L adalah karena huruf K berada di baris pertama, huruf L harus berada di baris pertama atau kedua. Huruf I pada peletakan sebelumnya diletakkan pada baris kedua kolom ketiga, yang artinya huruf L tidak bisa di sana. Namun karena peletakan huruf I tersebut, apabila huruf J berada pada kolom ketiga, huruf L tidak mungkin diletakkan. Apabila huruf J diletakkan selain pada kolom ketiga, huruf K hanya dapat diletakkan pada kolom pertama atau kolom kelima, karena jika huruf K diletakkan pada kolom ketiga, huruf L kembali tidak bisa diletakkan karena tempatnya telah diisi huruf lain. Apabila huruf K berada pada kolom pertama atau kolom kelima, huruf L masih tidak dapat diletakkan karena baik kolom pertama maupun



## 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.

Bandung, 5 Mei 2015

ttd

A handwritten signature in black ink, consisting of stylized, cursive letters that appear to be 'MG' followed by a horizontal line and a small dash.

Mahesa Gandakusuma (13513091)