

Penerapan Algoritma Backtracking Dalam Menyelesaikan Permainan Knight Tour

Sendi Novianto *¹, Dian Pramadhana ²

^{1,2}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro;
Jl. Nakula 1 No. 5-11, Semarang, 50131, Telp : (024) 351 7261, Fax (024) 352 0165
e-mail: *sendi.novianto@dsn.dinus.ac.id, dianpramadhana@gmail.com⁴

Abstrak

Dalam permainan Knight tour banyak pemain merasa kesulitan dalam menyelesaikan permainan ini, karena aturan permainannya yaitu bidak kuda harus melewati seluruh kotak papan catur tepat satu kali dengan langkah harus membentuk huruf "L" dan tanpa mengulangi kotak yang sudah pernah dilalui.

Dengan demikian di buat sistem untuk mencari solusi dalam menyelesaikan permainan dengan menerapkan algoritma backtracking untuk permainan catur yang bernama knight tour dengan papan permainan yang berukuran $a \times b$.

Penelitian ini menggunakan metode Action Research, menurut jenis yang dipakai peneliti ini termasuk model Kemmis. algoritma backtracking ini akan lebih cepat dan tepat dalam proses pencarian solusi sehingga pemain akan lebih mudah dalam menyelesaikan permainan knight tour.

penelitian ini akan menguraikan tentang proses pencarian solusi agar papan terisi semua. Desain penerapan algoritma backtracking ini meliputi langkah kuda agar papan terisi semua dan banyaknya langkah bidak kuda dalam menyelesaikan permainan.

Algoritma Backtracking mampu menemukan solusi pada permasalahan permainan knight tour pada papan catur $a \times b$ dengan cepat dan tepat. Diharapkan kedepannya dapat dijadikan sebagai inspirasi untuk penelitian yang lebih lanjut. Misalnya seperti menggunakan algoritma yang berbeda dalam menyelesaikan permasalahan pada permainan knight tour.

Kata kunci—Algoritma Backtracking, knight tour, Kecerdasan Buatan.

Abstract

In Knight's tour game many players find it difficult to finish this game, because the rules of the game that must pass through the entire knight box chess board exactly once with step must form the letter "L" and without repeating a box that has been passed.

Thus the system is made to find solutions in completing the game by applying a backtracking algorithm on game knight tour with a game board using size axb or random.

This study uses Action Research, according to the researchers used this type include Kemmis models. The backtracking algorithm will be faster and more accurate in the process of finding a solution so that the player will be easier to resolve game knight tour.

This research will elaborate on the process of finding a solution to fill all boards. Design the application of this algorithm includes a step backtracking knight that filled all the boards and the number of steps in finishing games or knight tour.

Backtracking algorithms are able to find solutions to problems on the board game chess knight tour axb quickly and accurately. Future expected to serve as an inspiration for further research. For example, like using different algorithms to solve the problems in the game knight tour.

Keywords—Algorithm Backtracking, Knight Tour, Artificial Intelligent

1. PENDAHULUAN

Masyarakat Persia dan masyarakat arab adalah yang pertama kali menemukan permainan catur. H.J.R. Murray merupakan penulis buku yang berisikan mengenai sejarah dari permainan catur (*History of Chess*) pada tahun 1913. Dalam buku itu diceritakan bahwa pada abad ke-6, permainan catur datang dari India, dimana permainan ini dikenal dengan nama *chaturanga*, yang mempunyai arti empat unsur yang terpisah. Awal pertama kali permainan catur ini ada, bidak catur hanya terdiri dari 4 jenis, yaitu ksatria yang di lambangkan dengan kuda, benteng, uskup yang dilambangkan dengan gajah, dan raja. Masyarakat didunia sudah lama sekali mengenal permainan catur ini dan selalu memainkannya. Permainan catur ini dimainkan dari berbagai kalangan baik dari kalangan anak-anak, remaja, serta dewasa dan orang tua juga turut bermain dalam permainan catur ini. Pada umumnya permainan catur ini dimainkan oleh dua orang dalam satu permainan. Dimana permainannya di letakkan pada papan catur yang berbentuk bujur sangkar dengan ukuran yang sama lebar dan panjangnya. Dalam papan catur tersebut terdapat kotak-kotak yang membagi sama rata dengan jumlah delapan baris dan delapan kolom dengan warna hitam dan putih yang disusun secara bergantian. Pada permainan catur ini biasanya tiap orang akan dibekali oleh delapan bidak, 1 raja, 1 ratu serta 2 peluncur, 2 kuda, 2 benteng yang terletak di kanan dan kiri antara raja dan ratu. Pada tiap anggota catur memiliki pola pergerakan yang sesuai dengan sifatnya masing-masing^[1].

Untuk dapat bermain catur, dibutuhkan adanya analisa dan ketajaman dari para pemainnya. Selain itu juga dibutuhkan adanya strategi yang dapat mengecohkan lawannya, sehingga kemenangan dari permainan catur ini dapat diperoleh. Pengecohkan lawan dapat dilakukan dengan cara mengorbankan beberapa anggota dari bidak catur agar raja dari lawan main dapat dikalahkan. Salah satu bidak catur yang memiliki pergerakan unik adalah bidak kuda. Bidak kuda ini hanya dapat bergerak dengan arah pergerakan membentuk huruf L. Karena keunikannya itulah akhirnya terbentuk sebuah permasalahan yang berhubungan dengan pergerakan bidak kuda pada papan catur. Permasalahan tersebut menarik untuk dipecahkan dan akhirnya membentuk sebuah permainan pengasah otak yang disebut *knight tour*^[2].

Penerapan konsep Artificial Intelligence (kecerdasan buatan) sangat banyak membantu dalam berbagai masalah pencarian dan *problem solving*. Salah satu contoh penerapan kecerdasan buatan untuk *problem solving* adalah dalam penyelesaian *knight tour problem*. *Knight tour* adalah salah satu permainan catur klasik yang menggunakan langkah bidak kuda pada papan catur. Aturan dari permainan ini sederhana. Permainan dimulai dengan meletakkan bidak kuda pada sebuah titik di atas papan catur kosong. Aturannya yaitu dapat melewati seluruh kotak papan catur tepat satu kali dengan langkah dari bidak kuda tersebut harus membentuk huruf "L" dan tanpa mengulangi kotak yang sudah pernah dilalui^[3].

Ada dua variasi permainan *knight tour*, yang pertama adalah *closed knight tour* yaitu *knight* harus melewati semua kotak dalam papan kuda catur dan kembali lagi ke kotak tempat awal permainan, yang kedua adalah *open knight tour* yaitu *knight* harus melangkah melalui semua kotak pada papan tanpa harus dapat kembali lagi ke kotak tempat awal permainan. Dalam permainan ini banyak terdapat variasi langkah yang dapat diambil serta dapat menghasilkan beberapa solusi.

Untuk menyelesaikan permainan ini sistem harus memiliki kemampuan pencarian solusi untuk setiap kondisi yang berbeda. Banyak algoritma yang bisa digunakan untuk permasalahan seperti ini, salah satunya adalah algoritma *backtracking*. *Backtracking* mempresentasikan masalah kedalam sebuah *tree*. Pencarian ini berdasar pada pencarian *Depth First Search (DFS)* yaitu dengan mencari solusi sampai pada kedalaman tertentu dan apabila tidak didapatkan goal yang optimal maka akan dilakukan pencarian dengan cara kembali lagi pada node sebelumnya. Untuk mengoptimalkan kecepatan pencarian dalam permainan *knight tour* perlu ditambahkan heuristik^[4].

2. LANDASAN TEORI

2.1 Penelitian Terkait

Jurnal penelitian dari Sahat Nicholas Simangunsong (2010/2011) dengan judul Aplikasi Algoritma Brute Force Pada Knight Tour Problem. Pada jurnal ini, Algoritma brute force merupakan sebuah pendekatan yang straight forward untuk menyelesaikan suatu persoalan. Algoritma ini bekerja dengan mencari seluruh kemungkinan solusi yang ada. Brute force termasuk jenis algoritma yang sederhana jika dibandingkan dengan algoritma-algoritma lainnya seperti algoritma greedy, algoritma breadth first search^[5], algoritma divide and conquer, dan sebagainya^[6]. Jika dilihat dari sisi keakuratan algoritma, algoritma brute force merupakan algoritma yang kurang akurat dan kurang cocok digunakan untuk memecahkan persoalan dengan waktu yang cepat, apalagi jika digunakan untuk menyelesaikan persoalan closed knight tour, tentu saja tidak bisa dikerjakan dengan waktu yang sangat singkat dalam menyelesaikan persoalan tersebut. Mencari jalur yang tepat agar bisa kembali ke posisi awal tentunya membutuhkan waktu yang lama. Algoritma backtracking merupakan salah satu algoritma yang dapat dipakai untuk memecahkan persoalan yang ada secara lebih ringkas dan cepat dibandingkan dengan penggunaan algoritma brute force.

Tabel 1. State of the Art

Masalah	Metode	Hasil
Bagaimana caranya agar bidak kuda dapat melewati semua papan petak yang telah tersedia, dimana tiap petaknya hanya dapat dikunjungi tepat hanya satu kali.	algoritma <i>backtracking</i>	Simulasi penyelesaian permainan <i>knight tour</i> dengan algoritma <i>backtracking</i>

2.2 Knight Tour

Knight tour merupakan permainan catur klasik mengenai langkah bidak kuda pada papan catur. Aturan dari permainan ini sederhana. Permainan dimulai dengan meletakkan bidak kuda pada sebuah titik di atas papan catur kosong. Papan yang digunakan dapat berukuran 5x5, 6x6, 8x8, dan sebagainya. Ukuran papan minimum yang dapat digunakan adalah 5x5. Untuk papan berukuran 3x3, 4x4, atau yang lebih kecil, tidak akan ditemukan solusi permainannya. Jika papan berbentuk persegi panjang, ukuran minimum papan adalah 3x4. Bidak kuda yang tadi telah diletakkan tersebut harus berjalan sesuai dengan langkah kuda pada permainan catur. Permainan knight tour ini akan selesai jika bidak kuda telah menginjak semua titik pada papan catur tanpa pernah menginjak titik yang sama dua kali. Permainan knight tour ini akan selesai jika bidak kuda telah menginjak semua titik pada papan catur tanpa pernah menginjak titik yang sama dua kali. Ada dua jenis permainan knight tour, yaitu permainan knight tour open knight's tour dan permainan knight tour closed knight's tour.

2.2 Algoritma

Algoritma merupakan urutan langkah-langkah yang tersusun secara sistematis. Penulisan algoritma dapat dilambangkan dengan lambang matematika agar dapat lebih mudah dipahami. Untuk lebih dapat dipahami oleh orang awam, algoritma dapat dituliskan dengan kalimat deskriptif disertai dengan notasi dan simbol matematika. Seperti halnya jika ingin menjelaskan tentang petak-petak yang sudah dikunjungi oleh bidak kuda, maka kita bisa jelaskan dengan "tiap

petak yang belum dikunjungi, bisa kita lambangkan dengan angka 0, sedangkan petak yang sudah dikunjungi oleh bidak kuda, ganti lambangnya dengan angka 1”.

2.2 Algoritma Backtracking

Algoritma backtracking adalah salah satu algoritma yang akan digunakan dalam memecahkan permasalahan knight tour ini^[7]. Algoritma ini akan digunakan karena dapat memberikan solusi yang lebih cepat dibandingkan dengan algoritma brute force. Cara kerja dari algoritma ini adalah dengan cara mencari solusi yang ada dengan jalan yang sistematis^[8]. Dalam pencariannya, tidak semua ruang solusi akan diperiksa, solusi yang akan dilanjutkan prosesnya adalah solusi yang mengarah pada goal yang dituju^[2].

3. HASIL DAN PEMBAHASAN

Dari penerapan algoritma backtracking untuk permasalahan knight tour, untuk proses lebih lanjutnya diperlukan beberapa parameter yang harus dipenuhi antara lain :

- a. Papan board yang akan digunakan adalah ukuran $a \times b$
- b. Untuk letak koordinasi kotak di wakili oleh (x,y) .
- c. Nomor kotak yang telah dilewati akan ditandai dengan variabel jalan
- d. Status dengan jenis boolean untuk menunjukkan apakah semua kotak papan berhasil dilewati.

```
type chess_board is array
{medefinisikan susunan papan catur}

(1..n,1..n) of integer;

{Nilai papan atau angka}

procedure knight-Move(board : in out
chess_board;

{memulaiperhitungan/memunculkan papan
dari nilai integer (Angka) dari nilai yang di
inputkan}

letak, x, y : in out integer;

{koordinat awal dari jalan kuda}

status : in out Boolean;

a, b : integer;

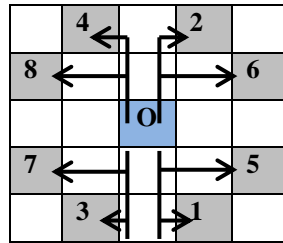
{koordinat awal dari jalan kuda}
```

```
begin
    if move = n^2+1 then
        Offsets: array [1..8]
        of Tpoint= ( (x:-2;y:-1), (x:-2;y:+1),
                    (x:-1;y:-2), (x:-1;y:+2), (x:+1;y:+2),
                    (x:+1;y:-2), (x:+2;y:+1), (x:+2;y:-1) );
        {Pergerakan kuda membentuk huruf L}
        ok := ( (x,y) = (1,1) );
        {knight menyesuaikan koordinat papan
        yang telah diinputkan}
    elsif board(x,y) /= 0 then
        {jika tak bergerak/muvo maka muncul nilai
        kesalahan}
        ok := false;
    else
        board(x,y) := move;
        {jika angka sudah muncul maka
        muncul angka dimana ada
        pergerakan kuda}
        loop
            (w,z) := Next position
            {Jika tidak muncul koordinat jalan
            kuda maka akan mengulang
            pergerakan kuda berikutnya}
        from (x,y);
        knight(board, w, z,
        move+1, ok );
        {nilai papan yang ditentukan}
```

```
exit when (ok or No moves remain);
end loop;
{jika pergerakan kuda mencapai akhir didapatkan
posisi kuda bergerak maksimal}
if not ok then
    board ( x,y ) :=0; –
    Backtracking
    {Pesan kesalahan, apabila tidak bisa loop
    akan mengembalikan pergerakan kuda
    sebelumnya}
end if;
end if;
end knight;
```

3.1 Implementasi Algoritma Backtracking

Dalam permainan knight tour pergerakan kuda harus melewati seluruh kotak papan catur tepat satu kali dengan rumusan $(x+2,y+1)$, $(x-2,y+1)$, $(x+2,y-1)$, $(x-2,y-1)$ atau $(x+1,y+2)$, $(x-1,y+2)$, $(x+1,y-2)$, $(x-1,y-2)$ yang akan membentuk lintasan berbentuk huruf L. Dengan demikian, knight memiliki maksimum 8 kemungkinan arah untuk tiap langkahnya, seperti gambar dibawah ini :



Gambar 1 Kemungkinan langkah kuda

Langkah-Langkah yang digunakan untuk memecahkan permainan knight tour adalah sebagai berikut :

1. Pilih posisi X secara random pada papan permainan dan tandai posisi tersebut sebagai posisi awal (tandai dengan angka 1)

1A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Gambar 2 Posisi awal dari permainan

2. Misalkan posisi-posisi selanjutnya yang dapat diakses dari posisi 1 disimpan dalam array Y pada contoh di bawah ini, posisi yang dapat diakses dari posisi 1 adalah Y1 dan Y2

1A	B	C	D	E
F	G	Y₁H	I	J
K	Y₂L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Gambar 3 Posisi selanjutnya dari permainan

3. Untuk setiap posisi yang tersimpan dalam Y, hitung berapa langkah maksimal yang dapat diakses dari posisi pada tiap Y. Posisi yang dihitung haruslah posisi yang belum pernah dilewati sebelumnya, langkah yang dapat dilakukan dari masing-masing posisi Y1 dan Y2 adalah :

Y1 : A,B,C,D,E = 5 Langkah

Y2 : A,B,C,D,E = 5 Langkah

Masukan jumlah langkah maksimal tersebut ke dalam array Y

1A	B	C	D	E
F	G	5H	I	J
K	5L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Gambar 4 Menghitung langkah maksimal

- Posisi selanjutnya dipilih berdasarkan nilai pada array Y yang paling kecil. Karena jumlah langkah yang diperoleh sama yaitu 5 langkah maka Pada contoh ini, posisi selanjutnya (posisi 2) adalah Y2. Masukkan angka 2 untuk menandakan bahwa posisi tersebut telah dilalui.

1A	B	C	D	E
F	G	H	I	J
K	2L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Gambar 5 Pemilihan pada array yang paling kecil

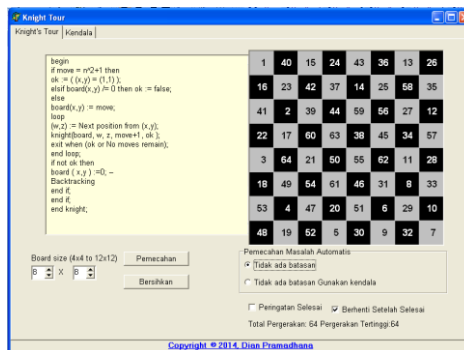
- Pencarian dilanjutkan dengan mengulangi langkah-langkah seperti diatas, sampai seluruh papan permainan terisi semua. Berikut Hasil permainan yang telah selesai dapat dilihat pada gambar di bawah ini.

1A	18B	13C	24D	7E
12F	25G	8H	19I	14J
17K	2L	23M	6N	9O
22P	11Q	4R	15S	20T
3U	16V	21W	10X	5Y

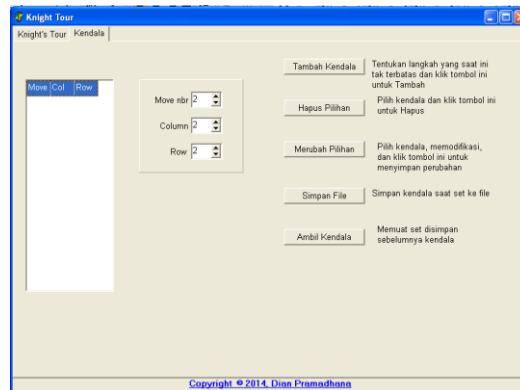
Gambar 6 Pengulangan langkah sebelumnya sampai akhir

3.2 Implementasi Algoritma Backtracking

Berikut ini adalah tampilan screen shoot dari hasil implementasi algoritma backtracking pada permainan knight tour



Gambar 7 Halaman Utama Permainan



Gambar 8 Panel Kendali

Berikut ini adalah tabel hasil pengujian dari permainan knight tour. Data yang dipakai adalah data permainan dari ukuran papan 4x4 sampai dengan papan ukuran 2x12

Tabel 1 Perbandingan hasil pengujian permainan knight tour

No	Ukuran Papan	Solusi Open Knight Tour	Solusi Close Knight Tour	Total Pergerakan Open Knight Tour	Total Pergerakan Close Knight Tour
1	4X4	Tidak Ada	Tidak Ada	Tidak Ada Solusi	Tidak Ada Solusi
2	5X5	Ada	Tidak Ada	25	Tidak Ada Solusi
3	6X6	Ada	Ada	36	57
4	7X7	Ada	Tidak Ada	49	Tidak Ada Solusi
5	8X8	Ada	Ada	64	64
6	9X9	Ada	Tidak Ada	81	Tidak Ada Solusi
7	10X10	Ada	Ada	100	119
8	11X11	Ada	Tidak Ada	121	Tidak Ada Solusi
9	12X12	Ada	Ada	144	144

Pada tabel 1 dapat kita lihat bahwa tidak semua ukuran papan dapat ditemukan solusi. Hal ini dikarenakan oleh keterbatasan langkah dari kuda, dimana langkah kuda harus selalu membentuk huruf "L". Semakin banyak ukuran papan, maka langkah yang diperlukan untuk menyelesaikan masalah yang ada juga semakin banyak juga.

4. KESIMPULAN

Kelebihan algoritma Backtracking untuk menemukan solusi permasalahan knight tour pada papan catur yang berukuran $a \times b$, dimana cara kerjanya dengan melakukan runut balik pada tiap langkahnya, sampai dengan semua posisi papan dicoba, sehingga ketika menemukan langkah yang diharapkan, dapat dengan mudahnya mencatat langkah-langkah yang telah dilaluinya yang merupakan solusi dari masalah yang dihadapinya saat itu.

DAFTAR PUSTAKA

- [1] Lewandowski, Gary. Project 1: The Knight's Tour Gary Lewandowski[M]. CSCI, 2001.
- [2] Munir, Rinaldi. Diktat Kuliah IF3051 Strategi Algoritma[A]. 见: Bandung: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2009.
- [3] Usman E. Definisi Algoritma dan Ciri Algoritma[EB/OL]. 2012.

- <http://blog.binadarma.ac.id/usman/2012/11/19/definisi-algoritma-dan-ciri-algoritma.html>. [Accessed 26 Maret 2014].
- [4] Putra. Penelitian Tindakan (Action Research)[EB/OL]. 2012. <http://firdausremistael.blogspot.com/2012/02/penelitian-tindakan.html>.
- [5] Mumtaz F. Penyelesaian Permasalahan Knight Tour Menggunakan Algoritma Breadth First Search (BFS)[A]. 见: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, 2010.
- [6] Simangunsong S N. Aplikasi Algoritma Brute Force Pada Knight's Tour Problem[M]. 2010.
- [7] Yudhistira D, Wamiliana, Kusniasari D. Algoritma Backtrack Untuk Pencarian Solusi Knight's Tour Problem Pada Papan Catur $m \times n$ [J]. Jurnal Komputasi, 2012, 1(1).
- [8] Wardani Wi, H D I S, Tanujaya A. Analisis Penerapan Algoritma Backtracking Dalam Pencarian Solusi Game "Crossword Puzzle"[A]. 见: Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, 2010.