**PENERAPAN ALGORITHMA BACKTRACKING DALAM MENYELESAIKAN PERMAINAN KNIGHT TOUR**

**Dian Pramadhana1, Sendi Novianto2**

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang

Jl. Nakula 1 No. 5-11, Semarang, 50131, Telp : (024) 351 7261, Fax (024) 352 0165

E-mail : [dianpramadhana@gmail.com](mailto:dianpramadhana@gmail.com)1, sendi.novianto@dsn.dinus.ac.id2

***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 pada permainan knight tour dengan papan permainan menggunakan ukuran m x n atau acak.*

*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 permainanan knight tour.*

*Laporan tugas akhir ini akan menguraikan tentang proses pencarian solusi agar papan terisi semua. Desain penerapan algoritma backtraking ini meliputi langkah kuda agar papan terisi semua dan banyaknya langkah bidak kuda dalam menyelesaikan permaianan.*

*Algoritma Backtracking mampu menemukan solusi pada permasalahan permainan knight tour pada papan catur m x n dengan cepat dan tepat. Diharapkan kedepannnya 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.*

***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 mxn 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 final report will elaborate on the process of finding a solution to fill all boards. Design the application of this algorithm includes a step backtraking 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 mxn 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: Algorithms Backtracking, knight tour*.

# 1. PENDAHULUAN

Permainan catur pertama kali ditemukan di masyarakat Persia dan Arab. Menurut H. J. R. Murray, penulis buku History of Chess (1913), catur berasal dari India dan mulai ada pada abad ke-6. Di sana catur dikenal dengan nama chaturanga, yang artinya empat unsur yang terpisah. Awalnya, bidak catur memang hanya empat jenis, yaitu raja, benteng, ksatria (kuda), dan uskup (gajah). Catur sudah lama sekali dikenal oleh masyarakat dunia dan dimainkan dari semua kalangan baik itu anak-anak, remaja, dewasa dan para manula. Permainan catur ini dimainkan oleh dua orang dalam suatu papan bujur sangkar yang terdiri dari delapan kolom dan delapan baris, yang berbentuk kotak dan pada umumnya berwarna hitam dan putih secara bergantian. Permainan catur terdiri dari satu raja, satu ratu, dua benteng, dua peluncur, dua kuda dan delapan bidak pada masing-masing pemainnya. Setiap jenis bidak catur memiliki pola gerakan tertentu.

Dalam permainannya, catur mengandalkan analisa dan ketajaman otak pemain, disertai keterampilan strategi dalam menentukan langkah, rencana, risiko, dan menentukan kapan harus berkorban agar menang. 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*.

Penerapan konsep *Artificial Intelegence* (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.

Ada dua variasi permaian *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. Salah satu *algoritma* yang dapat digunakan adalah *algoritma backtrackin*g. *Algoritma* ini 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 mengoptimalisasikan kecepatan pencarian dalam permainan *knight tour* perlu ditambahkan *heuristik*.

# 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.* Algoritma *brute force* merupakan sebuah pendekatan yang *straight forward* untuk menyelesaikan suatu persoalan. Algoritmaini bekerja dengan mencari seluruh kemungkinan solusi yang ada. *Brute force* termasuk jenis algoritmayang sederhana jika dibandingkan dengan algoritma-algoritma lainnya seperti algoritma *greedy*, algoritma *breadth first search*, algoritma *divite and conquer*, dan sebagainya. 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 dibutuhkan waktu yang sangat lama untuk menyelesaikan persoalan ini. Mencari jalur yang tepat agar bisa kembali ke posisi awal tentunya membutuhkan waktu yang lama. Algoritma runut balik (*backtracking*) merupakan algoritma yang digunakan untuk mencari solusi persoalan secara lebih mangkus daripada menggunakan algoritma *brute force.*

**Tabel I**

STATE OF THE ART

|  |  |  |
| --- | --- | --- |
| **Masalah** | **Metode** | **Hasil** |
| Sulitnya bidak kuda dalam melewati seluruh petak yang ada di papan catur tepat hanya sekali. | Metode algoritma *backtracking* | Aplikasi penyelesain permainan *knight tour* menggunakan metode algoritma *backtracking* dengan bahasa pemograman *borland delphi* |

**Tabel I.**

STATE OF THE ART

|  |  |  |  |
| --- | --- | --- | --- |
| **Masalah** | **Atribut** | **Metode** | **Hasil** |
| Kurangnya informasi mengenai kapan dan dalam kondisi apa pelanggan membeli produk tertentu | umur(*Age*), status pelajar atau bukan pelajar(Student), pendapatan(Income) dan tingkat peminjaman(Credit Rating) | Metode mamdani dan metode sugeno | Metode Sugeno mememperoleh hasil yang lebih akurat daripada metode Mamdani dalam memprediksi perilaku pembelian. |

**2.2 Landasan Teori**

1. Knight Tour

*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. 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 *open knight’s tour* dan *closed knight’s tour*.

1. Algoritma

Menurut Rinaldi Munir (2005 : 176) “Algoritma adalah urutan logis langkah-langkah penyeleseian masalah yang disusun secara sistematis”. Algoritma dapat dituliskan dalam berbagai notasi, misalnya dalam notasi kalimat-kalimat *deskriptif*. Dengan notasi kalimat *deskriptif*, deskripsi setiap langkah dijelaskan dengan bahasa sehari-hari secara jelas. Setiap langkah biasanya diawali dengan kata kerja seperti ‘baca’, ‘hitung’, ‘masukan’, ‘bagi’, ‘ganti’, dan sebagainya. Sedangkan pernyataan bersyarat dinyaakan dengan ‘jika’,’maka’, dan sebagainya.

1. Algoritma Runut Balik (*Backtracking*)

Algoritma runut balik (*backtracking*) merupakan algoritma yang digunakan untuk mencari solusi persoalan secara lebih mangkus daripada menggunakan algoritma *brute force.* Algoritma ini akan mencari solusi berdasarkan ruang solusi yang ada secara sistematis namun tidak semua ruang solusi akan diperiksa, hanya pencarian yang mengarah kepada solusi yang akan diproses. (Rinaldi Munir, Diktat Strategi Algoritmik, Teknik Informatika ITB. 2005).

**3. PEMBAHASAN**

Berikut adalah contoh algoritma backtraking untuk kasus knight tour :

* Boardnya adalah mxn (ukuran papan)
* (x,y) adalah koordinat letak kotak
* Move adalah nomor kotak yang telah dilewati
* Ok adalah boolean apakah sukses atau gagal

type chess\_board is array

{medefinisikan susunan papan catur}

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

{Nilai papan atau angka}

procedure knight (board : in out chess\_board;

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

x,y,move : in out integer;

{koordinat awal dari jalan kuda}

ok : in out Boolean) is

w, z : 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}

a

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 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **4** |  | **2** |  |
| **8** |  |  |  | **6** |
|  |  | **O** |  |  |
| **7** |  |  |  | **5** |
|  | **3** |  | **1** |  |

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** |

1. 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** |

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** |

4. 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** |

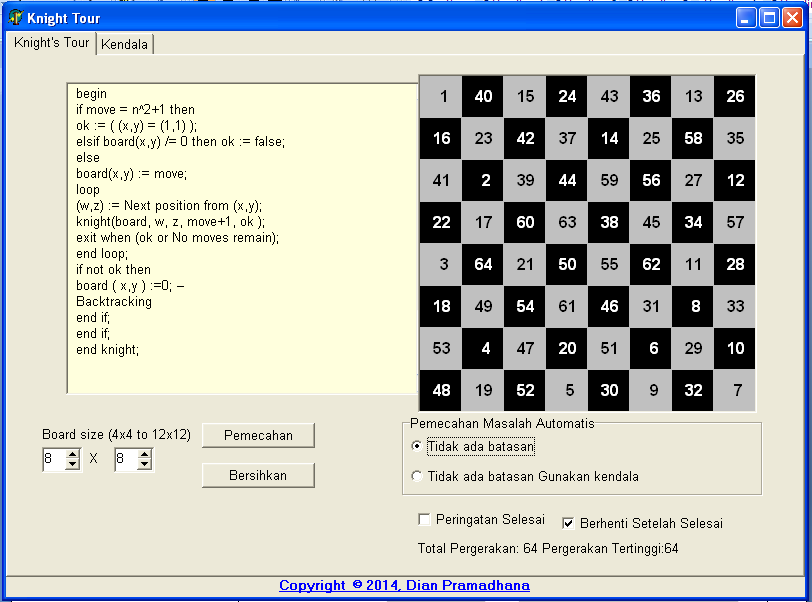
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1A** | **18B** | **13C** | **24D** | **7E** |
| **12F** | **25G** | **8H** | **19I** | **14J** |
| **17K** | **2L** | **23M** | **6 N** | **9O** |
| **22P** | **11Q** | **4R** | **15S** | **20T** |
| **3U** | **16V** | **21W** | **10X** | **5Y** |

5. 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.

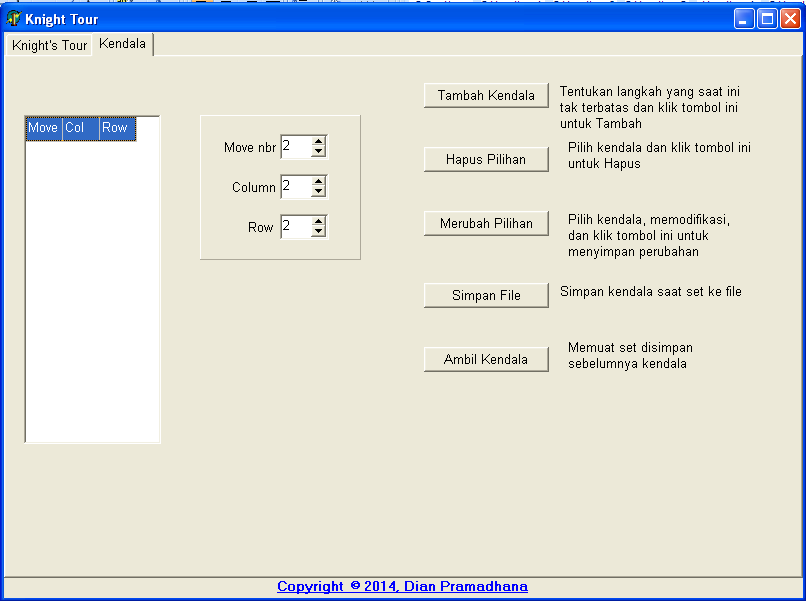
**3.2 Implementasi Program**

3.2.1 Tampilan untuk user

Tampilan halaman Utama



Panel Kendala



Pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 Sollusi |
| 9 | 12X12 | Ada | Ada | 144 | 144 |

**4. KESIMPULAN**

Kelebihan algoritma *Backtracking* untuk menemukan solusi permasalahan knight tourpada papan catur *m x n* yaitu mudah merunut balik suatu langkah, sehingga apabila menemui langkah terakhir maka dapat kembali pada posisi sebelumnya yang mempunyai langkah solusi sehingga dapat menemukan solusi lebih cepat.

# DAFTAR PUSTAKA

[1] Munir, Rinaldi. 2009. “*Diktat Kuliah IF3051 Strategi Algoritma*”, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[2] Lewandowski, Gary. “*Project 1: The Knight’s Tour Gary Lewandowski*”, CSCI 220, fall 2001

[3] Dolly Yudhistira, Wamiliana, Dian Kurniasari, “*Algoritma Backtrack Untuk Pencarian Solusi Knight’s Tour Problem Pada Papan Catur m xn*, ” Jurnal komputasi, Vol 1, No. 1, 2012.

[4] Sahat Nicholas Simangunsong, “*Aplikasi Algoritma Brute Force Pada Knight’s Tour Problem*,” Strategi Algoritma, 2010/2011.

[5] Widya Wardani, Dian Intania Savitri H, Allentine Tanujaya, “*Analisis Penerapan Algoritma Backtracking Dalam Pencarian Solusi Game “Crossword Puzzle*,” Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, 2010.

[6] **Ependi** Usman, “*Definisi Algoritma dan Ciri Algoritma”*, 19 November 2012. [Online]. Available: [http://blog.binadarma.ac.id/usman/2012/11/19/definisi-algoritma-dan-ciri-algoritma.html. [Accessed](http://blog.binadarma.ac.id/usman/2012/11/19/definisi-algoritma-dan-ciri-algoritma.html.%20%5bAccessed) 26 Maret 2014].

[7] Putra, “*Penelitian Tindakan (Action Research*)”, 06 Februari 2012. [Online]. Available: <http://firdausremistael.blogspot.com/2012/02/penelitiantindakan.html>. [Accessed 03-14 jam 11.00]

[8] Fahmi Mumtaz, *“Penyelesaian Permasalahan Knight Tour Menggunakan Algoritma Breadth First Search (BFS)”,* Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, 2010.