

Asah Otak dengan Knight's Tour Menggunakan Graf Hamilton dan Backtracking

Rama Febriyan (13511067)
 Program Studi Teknik Informatika
 Sekolah Teknik Elektro dan Informatika
 Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
 13511067@std.stei.itb.ac.id

Abstract—Makalah ini akan membahas bagaimana penerapan graf dalam permainan catur. Permainan yang dimaksud adalah permainan *knight's tour*. Permainan ini merupakan penerapan dari salah satu bentuk graf, yaitu Graf Hamilton. Selain itu, pada permainan ini juga dimanfaatkan algoritma *backtracking*.

Index Terms—*knight's tour*, graf, *backtracking*, Hamilton.

I. PENDAHULUAN

Catur merupakan permainan yang sangat terkenal. Permainan ini berasal dari India yang dikenal dengan nama *chaturanga* yang dalam bahasa Sanskerta yang berarti empat divisi (dalam militer). Di zaman India kuno, permainan catur dimainkan oleh empat orang yang berada pada empat sudut yang berbeda.

	a	b	c	d	e	f	g	h	
8	X			X	X			X	8
7									7
6									6
5	X			X	X			X	5
4	X			X	X			X	4
3									3
2									2
1	X			X	X			X	1
	a	b	c	d	e	f	g	h	

Gambar 1 Ashtāpada, papan permainan *chaturanga*

Permainan catur modern yang kita kenal sekarang berawal pada tahun 1000 yang langkah dasarnya mengadopsi langkah yang digunakan di Italia dan Spanyol. Sedangkan turnamen catur baru dimulai sekitar tahun 1850.

Selain itu permainan catur banyak dikembangkan dalam berbagai versi seperti, catur cepat, catur kilat, catur buta, dan berbagai versi lainnya. Salah satu permainan berbasis catur yang cukup terkenal adalah *knight's tour*. Pada permainan ini, pemain hanya menggunakan bidak kuda.

Ketika bermain, bidak harus melewati seluruh kotak pada papan catur tanpa ada yang terlewat dua kali. Permainan ini merupakan penerapan dari salah satu teori graf dan algoritma.

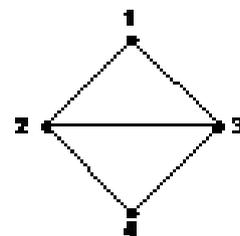
II. GRAF

“A graph $G(V,E)$ consists of a set V of elements called *vertices* and a set E of unordered pairs of members of V called *edges*.”(McHugh, 1990: 1). Dari terminologi tersebut diketahui bahwa definisi graf adalah himpunan tidak-kosong dari simpul dan himpunan sisi yang menghubungkan simpul. Perlu diketahui bahwa sebuah graf mungkin saja tidak memiliki sisi, namun harus memiliki simpul, minimal satu simpul.

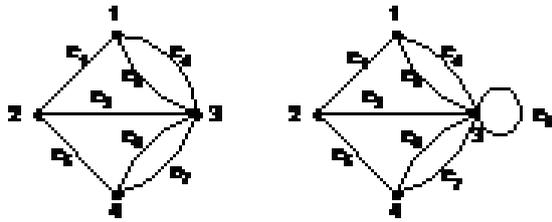
A. Jenis-Jenis Graf

Jika dilihat berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf maka graf terbagi atas dua jenis:

- Graf sederhana, yaitu graf yang tidak mengandung gelang pada salah satu simpul.
- Graf tak-sederhana, yaitu graf yang mengandung sisi ganda atau gelang. Graf yang mengandung sisi ganda disebut graf ganda, sedangkan graf yang mengandung gelang disebut graf semu.



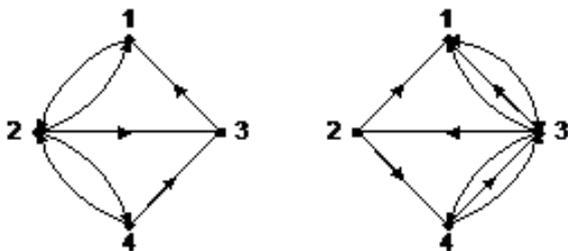
Gambar 2 Graf Sederhana



Gambar 3 Graf tak-sederhana

Jika dilihat dari jumlah simpul, maka graf dibedakan atas graf berhingga dan graf tak-berhingga. Berdasarkan arah pada sisi-sisi graf, graf juga dibedakan menjadi 2, yaitu:

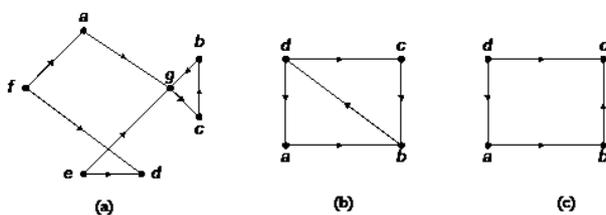
- Graf tak-berarah, yaitu graf yang sisinya tidak mempunyai orientasi arah.
- Graf berarah (busur), yaitu graf yang sisinya memiliki orientasi



Gambar 4 Graf Berarah

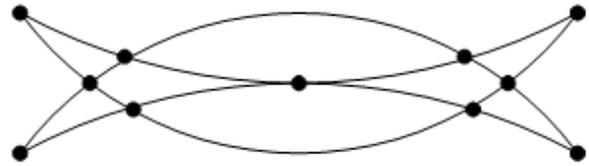
B. Graf Euler

Graf Euler adalah graf yang melalui masing-masing sisi graf tepat satu kali dan kembali ke simpul asal, graf ini dinamakan juga sebagai sirkuit Euler. Lintasan Euler adalah lintasan yang melalui masing-masing sisi graf tepat satu kali namun tidak kembali ke simpul asal. Lintasan Euler juga dapat disebut graf semi-Euler.



Gambar 5 (a) Graf berarah Euler, (b) Graf semi-Euler dan (c) Graf berarah yang tidak memiliki lintasan dan sirkuit Euler

Salah satu teorema menyatakan bahwa graf terhubung tak-berarah merupakan graf Euler jika dan hanya jika setiap simpul memiliki derajat genap. Contoh untuk teorema tersebut adalah Bilan Sabit Muhammad yang digambarkan tanpa mengangkat pensil.



Gambar 6 Bulan Sabit Muhammad

C. Graf Hamilton

Graf Hamilton hampir sama dengan graf Euler. Namun perbedaannya terletak pada graf Hamilton melewati satu simpul tepat satu kali. Jadi, lintasan Hamilton adalah lintasan yang melalui tiap simpul pada graf tepat satu kali. Sedangkan sirkuit Hamilton adalah lintasan Hamilton yang kembali ke simpul awal.

Untuk menentukan apakah suatu graf merupakan graf Hamilton atau tidak, digunakan teorema Ore. Bunyi Teorema Ore adalah sebagai berikut:

“misal $G(V,E)$ adalah graf dengan $|V| \geq 3$, dan setiap pasangan simpul tak terhubung u dan v pada G

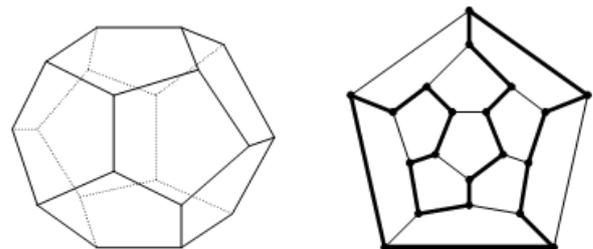
$$\deg(u) + \deg(v) \geq M,$$

untuk M bilangan integer. Jika M sama dengan $|V|$ maka G adalah graf Hamilton.”

Beberapa teorema lain terkait graf Hamilton yaitu:

TEOREMA “Setiap graf lengkap adalah graf Hamilton”

TEOREMA “Didalam graf lengkap G dengan n simpul ($n \geq 3$), terdapat $(n-1)!/2$ buah sirkuit Hamilton.”



Gambar 7 Dodecahedron Hamilton (kiri), dan graf yang mengandung sirkuit Hamilton (kanan)

D. Terminologi Graf Dasar

Berikut beberapa istilah (terminologi) dasar yang sering digunakan dalam graf:

- Bertetangga. Dua buah simpul dikatakan bertetangga jika dihubungkan oleh sebuah sisi.
- Bersisian.
- Simpul terencil, yaitu simpul yang tidak mempunyai sisi yang bersisian.
- Graf kosong yaitu graf yang himpunan sisinya merupakan himpunan kosong
- Derajat, jumlah sisi yang bersisian dengan suatu simpul.
- Lintasan adalah barisan berselang seling antara simpul dan sisi yang berurutan.

- Siklus atau sirkuit yaitu lintasan yang berawal dan berakhir pada satu simpul yang sama.
- Terhubung, dua buah simpul dikatakan terhubung jika terdapat lintasan antara kedua simpul tersebut.
- Upagraf adalah bagian dari graf dimana simpul dan sisinya merupakan anggota dari himpunan simpul dan sisi graf asal
- Upagraf merentang yaitu jika sebuah upagraf mengandung semua simpul dari graf asal
- *Cut-set* adalah himpunan sisi yang jika diabaikan akan menyebabkan sebuah graf terhubung menjadi tak-terhubung
- Graf berbobot adalah graf yang setiap sisinya diberi nilai

E. Beberapa graf sederhana khusus

a. Graf Lengkap

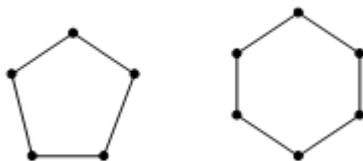
Graf Lengkap adalah graf yang setiap simpul mempunyai sisi yang menghubungkan ke sisi berikutnya.



Gambar 8 Graf Teratur

b. Graf Lingkaran

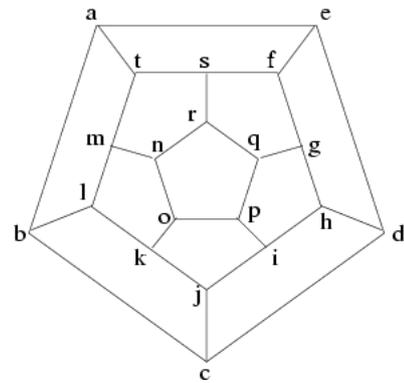
Graf lingkaran adalah graf yang setiap simpulnya masing-masing memiliki derajat 2



Gambar 9 Graf Lingkaran

c. Graf Teratur

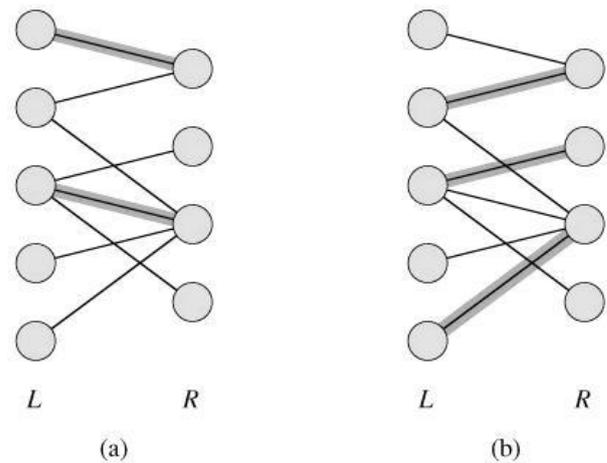
Graf teratur adalah graf yang setiap simpulnya memiliki derajat yang sama



Gambar 10 Graf teratur

d. Graf Bipartit

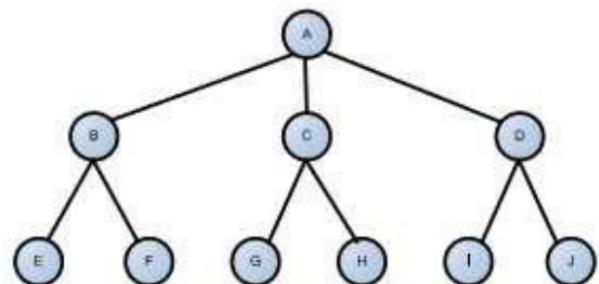
Graf yang simpulnya dapat dipisah menjadi dua himpunan bagian disebut graf bipartit



Gambar 11 Graf yang sama dengan 2 himpunan simpul yang berbeda

III. ALGORITMA BACKTRACK

Penerapan pencarian secara *brute force* dalam pencarian sebuah konfigurasi dapat disebut dengan *exhaustive search*. Pencarian ini sangat memakan biaya dan waktu yang besar, kecuali jika digunakan dalam skala kecil. Karena itu, dilakukan beberapa modifikasi untuk jenis pencarian ini dengan memanfaatkan sifat dari persoalan untuk mencegah pemeriksaan setiap kemungkinan. Pencarian ini disebut dengan *backtracking*.



Misalkan pohon di atas menggambarkan solusi dari

suatu persoalan. Jika kita ingin mencari solusi dari A ke E, maka jalur yang harus ditempuh adalah (A-B-E). Demikian juga untuk solusi-solusi yang lain. Algoritma Backtracking akan memeriksa jalur secara DFS. DFS adalah sebuah pencarian secara runut mulai dari akar suatu pohon, hingga simpul dari akar tersebut. Pada simpul tersebut juga dilakukan proses yang sama secara rekursif.

Algoritma Backtracking ini akan melakukan pencarian dimulai dari solusi, yaitu E. Jika ternyata E bukanlah solusi yang diharapkan, maka pencarian akan dilanjutkan ke F. Jalur yang harus dilalui untuk bisa mencapai E adalah (A-B-E) dan untuk mencapai F adalah (A-B-F). Kedua solusi tersebut memiliki jalur awal yang sama, yaitu (A-B). Jadi, daripada memeriksa ulang jalur dari A kemudian B, maka jalur (A-B) disimpan dulu dan memeriksa solusi untuk F.

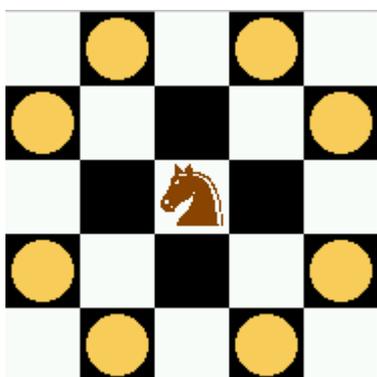
Pada Algoritma ini, jika penelusuran terhadap salah satu simpul diketahui tidak akan menemukan solusi, maka penelusuran pada simpul tersebut akan dihentikan, dan dilanjutkan pada simpul lain yang memiliki kedalaman yang sama.

Algoritma backtrack bekerja dengan cara menyimpan hasil dari perhitungan sebelumnya agar jika suatu saat dibutuhkan dapat dipanggil kembali.

IV. KNIGHT'S TOUR

Knight's Tour merupakan sebuah permainan hasil variasi dari permainan catur. Pada permainan ini, pemain hanya menggunakan bidak kuda. Bidak kuda ini akan digerakkan melewati seluruh kotak pada papan catur tanpa melewati kotak yang sama.

Pada permainan ini bidak kuda dapat diletakkan dimana saja untuk memulai permainan. Langkah bidak kuda yang unik yaitu membentuk huruf L membuat permainan ini menjadi permainan asah otak yang cukup baik.



Gambar 12 Langkah pada bidak kuda

Setelah kuda diletakkan pada kotak pertama, kuda digerakkan menurut aturan. Pada kotak berikutnya, kuda digerakkan kembali dengan langkah membentuk huruf L sehingga melewati semua kotak pada papan catur.

Terdapat dua jenis permainan *knight's tour*. Yaitu *open knight's tour* dan *closed knight's tour*. Pada *open knight's tour*, bidak akan berhenti pada satu kotak setelah semua kotak dilewati, sedangkan pada *closed knight's tour*, bidak kuda akan kembali pada titik awal permainan setelah semua kotak terlewati.

Permainan ini tidak hanya dapat digunakan pada papan berukuran 8x8, namun juga dapat dilakukan pada papan berukuran 6x6, dan 5x5. Untuk papan persegi panjang, ukuran minimal adalah 3x4.

34	3	12	15	28	1
13	22	35	2	11	16
4	33	14	29	36	27
21	30	23	8	17	10
24	5	32	19	26	7
31	20	25	6	9	18

Gambar 13 Penyelesaian Knight's Tour pada papan 6x6

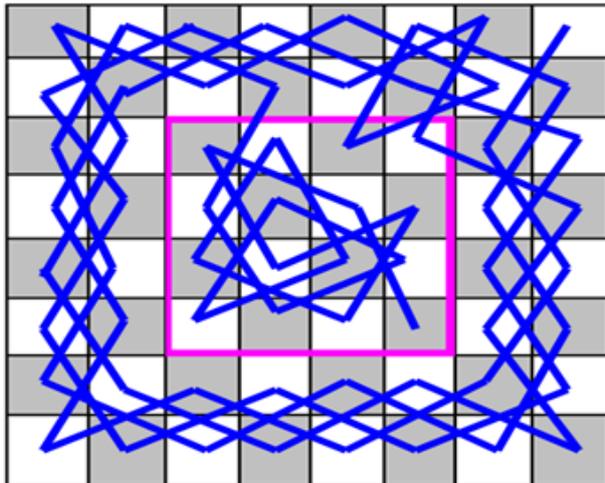
 1	48	31	50	33	16	63	18
30	51	46	3	62	19	14	35
47	2	49	32	15	34	17	64
52	29	4	45	20	61	36	13
5	44	25	56	9	40	21	60
28	53	8	41	24	57	12	37
43	6	55	26	39	10	59	22
54	27	42	7	58	23	38	11

Gambar 14 Penyelesaian Knight's Tour pada papan 8x8

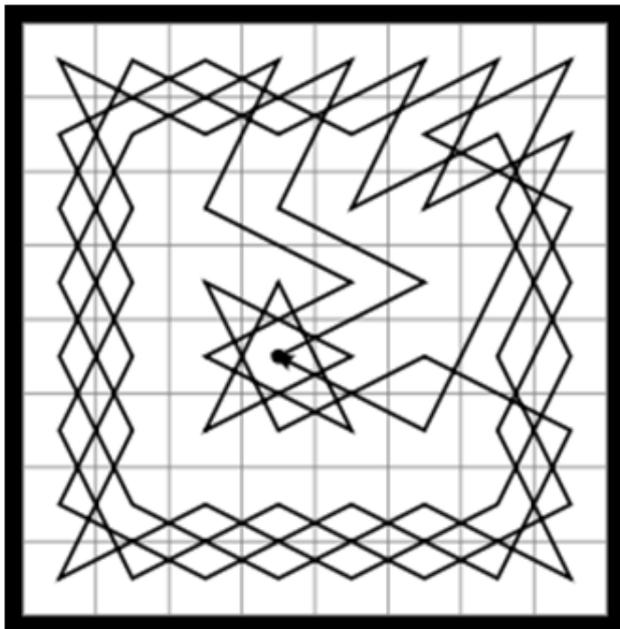
V. GRAF DAN ALGORITMA BACKTRACK PADA KNIGHT'S TOUR

Jika dilihat dari cara pergerakannya, permainan *knight's tour* ini menggunakan prinsip graf Hamilton. Graf Hamilton hanya melewati setiap simpul tepat satu

kali. Pada *knight's tour* bidak kuda tidak boleh melewati kotak yang sama lebih dari satu kali.



Gambar 15 Lintasan Hamilton pada Knight's Tour



Gambar 16 Sirkuit Hamilton pada Knight's Tour

Dalam menyelesaikan *knight's tour*, dapat digunakan berbagai cara. Namun tidak semua cara merupakan cara yang mangkus. Misalnya jika menggunakan konsep faktorial biasa, terdapat $64! = 1.27 \times 10^{89}$ kemungkinan. Akan membutuhkan waktu yang sangat lama untuk menyelesaikannya. Dengan menggunakan konsep eksponensial terdapat 64×4^{63} kemungkinan. Jika menggunakan konsep simetri, akan menghasilkan kemungkinan yang lebih kecil yaitu 8.5×10^{38} kemungkinan. Namun semua kemungkinan tersebut masih sangat besar.

Untuk itu digunakan algoritma *backtracking* dalam menyelesaikan masalah tersebut. Berikut contoh dari algoritma *backtracking* untuk menyelesaikan *knight's*

tour:

```

type chess_board is array
  (1..n,1..n) of integer;
procedure knight (board : in out
  chess_board;
  x,y,move : in out integer;
  ok : in out Boolean) is
  w, z : integer;

begin
  if move = n^2+1 then
    ok := ( (x,y) = (1,1) );
  elsif board(x,y) /= 0 then
    ok := false;
  else
    board(x,y) := move;
    loop
      (w,z) := Next position
        from (x,y);
      knight(board, w, z,
        move+1, ok );
      exit when (ok or No
        moves remain);
    end loop;
    if not ok then
      board
        ( x,y ) :=
        0;
      Backtracking
    end if;
  end if;
end knight;

```

Gambar 17 Contoh Algoritma Backtrack dalam Knight's Tour

Algoritma diatas akan berhenti apabila mencapai kondisi OK. Apabila kondisi OK tidak tercapai, maka akan dilakukan *backtracking* langkah yang telah dilakukan.

VI. KESIMPULAN

Graf dapat ditemukan dimana saja dalam kehidupan manusia, mulai dari jalan, rangkaian listrik hingga permainan. Permainan *knight's tour* merupakan permainan yang memanfaatkan teori graf Hamilton. Masih banyak hal lain yang juga memanfaatkan graf Hamilton seperti tukang pos yang mengantarkan surat harus memanfaatkan sifat dari graf Hamilton agar pekerjaannya lebih cepat.

Langkah-langkah dalam membentuk graf Hamilton pada permainan *knight's tour* dapat dibuat lebih mudah dengan menggunakan algoritma *backtracking*. *Backtracking* terbukti lebih mangkus daripada algoritma lain dalam menyelesaikan *knight's tour* karena algoritma ini membuang segala kemungkinan yang bukan

merupakan solusi dari permainan. Berbeda dengan faktorial yang akan melakukan pengecekan terhadap setiap kemungkinan yang ada. Lagipula, penggunaan konsep faktorial juga belum tentu memenuhi syarat dalam permainan *knight's tour* yang hasil dari setiap pergerakan bidak kuda harus membentuk lintasan Hamilton atau sirkuit Hamilton.

REFERENCES

- [1] Alfian, M Rijal. 2011. *Knight's Tour*. <http://gamatika.wordpress.com> . diakses tanggal 16 Desember 2012
- [2] Liu, C. L. 1985. *Element of Discrete Mathematics second edition* United State of America: McGraw-Hill
- [3] McHugh, James A. 1990. *Algorithmic Graph Teory*. New Jersey: Prentice-Hall
- [4] Munir, Rinaldi. 2008. *Diktat Kuliah IF2091 Struktur Diskrit*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
- [5] Thompson, Dan. 2008. *The Knight's Tour*. <http://www.boardchess.org>. Diakses tanggal 17 Desember 2012.

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, 17 April 2012



Rama Febriyan (13511067)