

Penerapan Teori Lintasan Hamilton pada Permainan *Knight's Tour* dengan Algoritma *Backtracking* dan *Divide-and-Conquer*

Bintang Fajarianto 13519138
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519138@std.stei.itb.ac.id

Abstrak—Catur merupakan sebuah permainan strategi yang dimainkan oleh dua orang. Pada umumnya, dua pihak tersebut dilambangkan dengan warna hitam dan putih. Dalam permainan ini, terdapat enam jenis bidak dengan pola pergerakan yang berbeda-beda, yaitu *king*, *queen*, *rook*, *bishop*, *knight*, dan *pawn*. Tujuan utama permainan ini adalah menjebak sang raja sehingga tidak ada lagi pergerakan yang dapat melindungi sang raja dari strategi serangan lawan.

Kini, permainan catur telah memiliki beragam variasi. Salah satunya adalah *Knight's Tour* yang hanya menggunakan satu jenis bidak, yaitu *knight* atau sering kita disebut dengan kuda. Dengan pergerakannya yang menyerupai “letter L”, permainan ini mengharuskan pemain untuk melewati seluruh kotak dalam papan permainan tepat sekali dan kembali menuju titik awal menggunakan bidak tersebut. Dalam makalah ini, akan dibahas penerapan dari teori Graf mengenai Lintasan Hamilton pada permainan *Knight's Tour* dengan algoritma *backtracking* dan *divide-and-conqueror*.

Keywords—lintasan Hamilton, *knight's tour*, *backtracking*, *divide-and-conquer*.

I. PENDAHULUAN

Catur merupakan permainan strategi yang dimainkan oleh dua orang. Permainan ini diyakini berasal dari India dan dikenal pertama kali dengan sebutan “*Chaturanga*”. Pada awalnya, bidak catur hanya terdiri dari 4 jenis yang sering dihubungkan dengan unsur-unsur kehidupan, yaitu air, tanah, api, dan udara. Selain itu, dahulu catur juga dianggap sebagai simbol dari cara manusia menjalani kehidupan.



Gambar 1.1. Permainan catur masa kini (www.istockphoto.com)

Kini, permainan catur telah mengalami banyak perubahan. Sejak abad 12, bidak-bidak catur ditetapkan menjadi enam jenis dengan pola pergerakan yang berbeda-beda, yaitu *king*, *queen*, *rook*, *bishop*, *knight*, dan *pawn*. Aturan permainannya juga telah banyak mengalami perubahan. Permainan akan berakhir apabila salah satu pemain kehilangan bidak rajanya dan pihak yang masih memiliki raja dinyatakan sebagai pemenang. Selain itu, apabila hingga akhir permainan kedua pemain tidak dapat membunuh raja, maka permainan akan berakhir. Keadaan tersebut juga dikenal sebagai keadaan remis.

II. KNIGHT'S TOUR

Permainan catur memiliki banyak variasi permainan, salah satunya adalah *Knight's Tour* atau dikenal juga sebagai Tur Kuda. Dalam permainan ini, hanya satu pemain yang dapat memainkannya dan hanya satu bidak yang dapat digunakan, yaitu *knight* atau kuda.

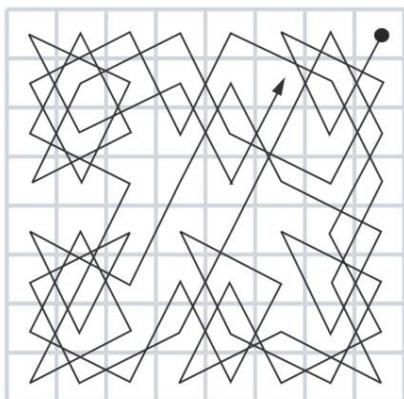


Gambar 2.1. *Knight* atau kuda (www.shutterstock.com)

Dalam permainan catur, kuda memiliki pola pergerakan yang menyerupai “letter L”. Pada permainan *Knight's Tour*, pemain diminta untuk melewati seluruh petak pada papan permainan **tepat sekali**, baik pada papan berukuran 8x8 (ukuran umum) maupun papan dengan bentuk variasi.

Permainan ini berkembang pada abad 18 di daerah Eropa. Permainan ini menarik perhatian para ilmuwan karena permainan ini dianggap memiliki tingkat kesulitan yang tinggi dan memerlukan logika yang mendalam.

Setelah teori Graf ditemukan oleh Sir William Rowan Hamilton, pergerakan bidak kuda dalam papan catur direpresentasikan sebagai sebuah graf dengan tiap kotak papan merupakan simpul dan perpindahan kuda dari satu kotak ke kotak lain merupakan sebuah sisi.



Gambar 2.2. Pergerakan kuda pada papan catur (www.lastwordonnothing.com)

Pada permainan ini, terdapat dua sebutan untuk kondisi ketika berhasil menyelesaikan perjalanan kuda pada papan, antara lain *Closed Tour* dan *Open Tour*. *Closed Tour* adalah kondisi ketika kotak awal mampu diraih oleh kuda ketika mencapai kotak terakhir, seperti pada Gambar 2.2. *Open Tour* adalah kondisi ketika kotak awal tidak mampu diraih oleh kuda ketika mencapai kotak terakhir.

III. GRAF

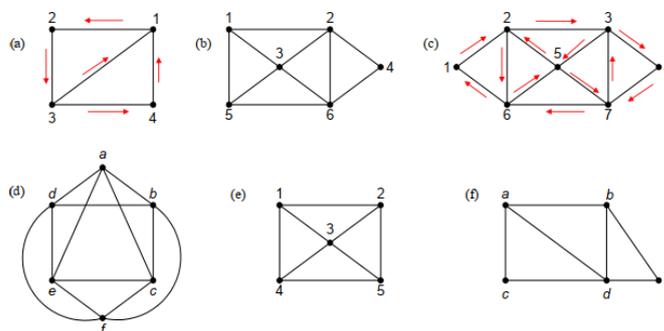
Graf merupakan himpunan yang beranggotakan himpunan tidak kosong yang berisi simpul-simpul (vertices) serta himpunan yang berisi sisi-sisi (edges) yang menghubungkan dua simpul. Dengan demikian, dapat dikatakan bahwa graf hanya terdefinisi hanya jika memiliki sebuah simpul dan graf dinyatakan tidak kosong apabila memiliki minimal dua buah simpul dan sebuah sisi.



Gambar 3.1. Simple graph (Rinaldi Munir, Graf bagian 1)

A. Graf Euler

Graf Euler (Eulerian Graph) adalah graf yang mempunyai sirkuit Euler. Sirkuit Euler adalah sirkuit yang melewati setiap sisi tepat satu kali. Sementara itu, graf yang memiliki lintasan Euler disebut sebagai graf semi-Euler. Lintasan Euler adalah lintasan yang melalui setiap sisi dalam graf tepat satu kali.



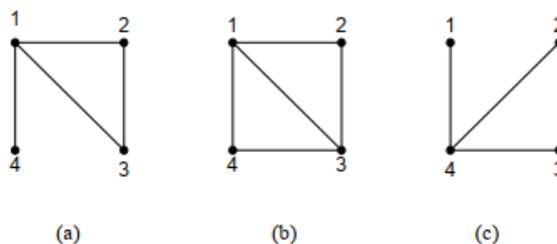
Gambar 3.2. (a), (b), dan (f) Graf semi-Euler, (c) dan (d) Graf Euler, (e) bukan keduanya (Rinaldi Munir, Graf bagian 3)

Suatu graf tidak berarah dikatakan memiliki lintasan Euler jika dan hanya jika saling terhubung dan memiliki dua buah simpul berderajat ganjil atau tidak ada simpul berderajat ganjil sama sekali. Selain itu, graf tidak berarah G merupakan graf Euler jika dan hanya jika setiap simpulnya berderajat genap.

B. Graf Hamilton

Graf Hamilton merupakan graf yang hampir sama dengan graf Euler. Perbedaannya hanya terletak pada simpul yang dilewati oleh sirkuit Hamilton tepat satu kali, kecuali simpul asal yang dilalui dua kali (berperan sebagai simpul akhir).

Graf Hamilton adalah graf yang memiliki sirkuit Hamilton. Sirkuit Hamilton adalah sirkuit yang melalui setiap simpul di dalam graf tepat satu kali, kecuali simpul asal yang dilalui dua kali (juga berperan sebagai simpul akhir). Sementara itu, graf yang memiliki lintasan Hamilton disebut sebagai graf semi-Hamilton. Lintasan Hamilton adalah lintasan yang melalui setiap simpul di dalam graf tepat satu kali.



Gambar 3.3. (a) Graf semi-Hamilton, (b) Graf Hamilton, (c) bukan keduanya (Rinaldi Munir, Graf bagian 3)

Terdapat suatu teorema bernama Teorema Ore yang digunakan untuk menentukan apakah suatu graf merupakan graf Hamilton atau bukan. Bunyi Teorema Ore adalah sebagai berikut:

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

$$\text{deg}(u) + \text{deg}(v) \geq M,$$

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

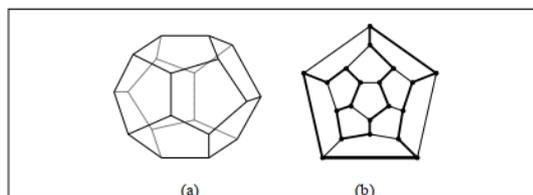
Berikut ini merupakan teorema lain yang masih berhubungan dengan graf Hamilton:

Teorema.

“Setiap graf lengkap adalah graf Hamilton.”

Teorema.

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



Gambar 3.4. (a) Dodecahedron Hamilton, (b) graf yang mengandung sirkuit Hamilton (Rinaldi Munir, Graf bagian 3)

IV. BACKTRACKING

Algoritma *Backtracking* merupakan pengembangan dari algoritma brute force. Algoritma brute force merupakan algoritma yang mampu mencoba semua kemungkinan yang dapat terjadi hingga 1.2688693×10^{89} . Dari semua kemungkinan tersebut, tentu tidak sedikit kemungkinan yang tidak diperlukan karena tidak mungkin terjadi. Oleh karena itu, algoritma *Backtracking* hadir sebagai algoritma dengan tingkat efisiensi lebih tinggi dibandingkan dengan algoritma brute force, yaitu dengan membuang semua kemungkinan yang tidak diperlukan.

Algoritma *Backtracking* pertama kali diperkenalkan pada tahun 1950 oleh D. H. Lehmer. Penggunaan algoritma ini telah terbukti mampu menyelesaikan berbagai permasalahan, seperti *Knight's Tour*, *tic-tac-toe*, *mazeMath*, dan lain-lain.

Pada Algoritma *Backtracking*, solusi dicari dengan membuat lintasan dari akar menuju daun. Dalam hal ini, aturan yang digunakan adalah aturan DFS atau *Depth First Search*. Simpul-

simpul yang telah dihasilkan dinamakan simpul hidup atau *live node*, sedangkan simpul yang sudah tidak terpakai lagi dinamakan simpul mati atau *dead node*. Selain itu, simpul yang diperluas dinamakan simpul E atau *expand node*.

Algoritma *Backtracking* dilakukan dengan mencari solusi parsial pada sebuah simpul tertentu yang kemudian simpul tersebut diperluas dan dianalisis. Apabila lintasan yang terbentuk tidak mengarah kepada solusi, maka *node* tersebut akan dibuang dan menjadi *dead node*.

Dalam proses pembuangan simpul mati, digunakan fungsi pembatas (*bounding function*). Apabila saat pencarian berakhir pada *dead node*, maka pencarian akan dilakukan pada simpul anak yang lainnya. Namun, jika sudah tidak ada lagi simpul anak yang belum dicari, maka akan dilakukan *backtracking* ke simpul sebelumnya. Sementara itu, pencarian baru akan berhenti dilakukan jika sudah solusi sudah didapatkan atau *backtracking* sudah tidak dapat dilakukan lagi.

Berikut ini adalah Algoritma *Backtracking* pada permainan *Knight's Tour*:

- (1) Data semua kemungkinan langkah yang dapat dilalui kuda dari posisi awal.
- (2) Pilih salah satu langkah dan perluas kembali langkah tersebut.
- (3) Tempatkan kuda pada petak yang dipilih.
- (4) Ulangi langkah pertama hingga ketiga untuk petak saat ini.
- (5) Apabila solusi belum ditemukan, kembalilah ke langkah sebelumnya (*backtracking*).

*note: pencarian dihentikan hanya jika solusi telah ditemukan atau tidak ada lagi langkah yang memungkinkan.

Berikut ini adalah contoh pseudocode Algoritma *backtracking* dalam permainan *Knight's Tour*:

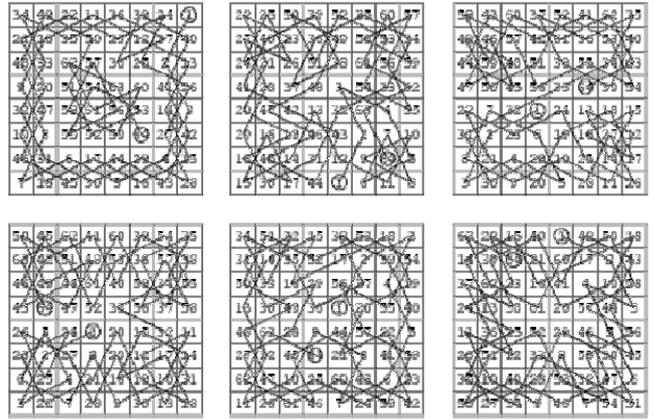
```

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 4.1. Algoritma *backtracking* pada permainan *Knight's Tour* (www.gamatika.wordpress.com)

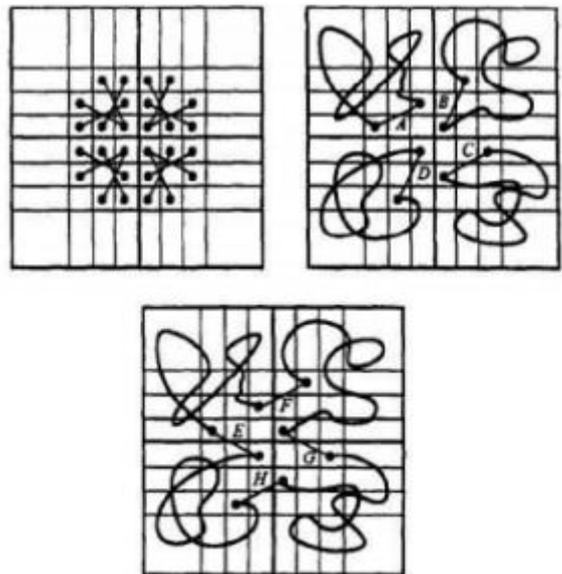


Gambar 4.2. Contoh solusi permainan *Knight's Tour* (www.mathworld.wolfram.com)

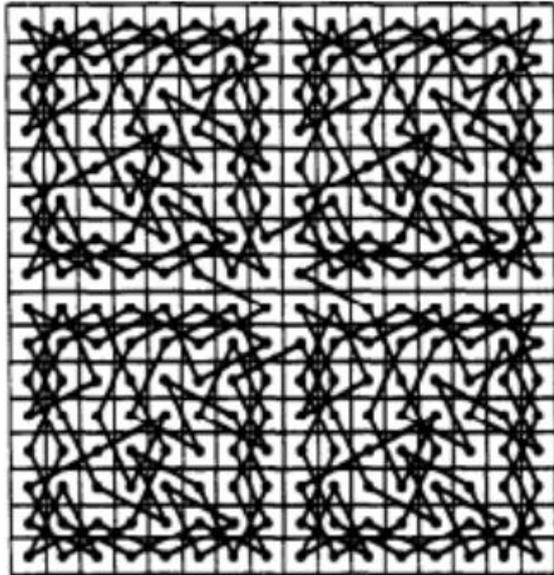
V. DIVIDE-AND-CONQUER

Algoritma *Divide-and-Conquer* merupakan salah satu algoritma yang digunakan untuk menyelesaikan permasalahan *Knight's Tour*. Proses kerja dari algoritma ini adalah dengan membagi papan menjadi bagian yang lebih kecil sehingga lebih mudah untuk diselesaikan.

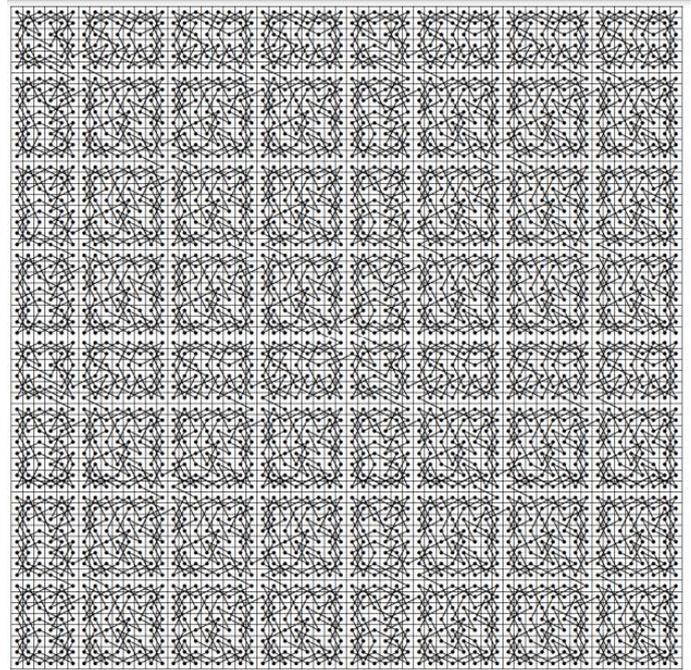
Pada papan berukuran $n \times n$ dengan n genap, papan dibagi menjadi empat bagian berukuran sama yang lebih kecil dari ukuran papan awal. Sebagai contoh, ukuran papan awal adalah 16×16 , maka papan dibagi menjadi ukuran yang lebih kecil lagi, yaitu 8×8 . Bagian kecil tersebut dinamakan sebagai basis. Pada masing-masing basis tersebut, akan diterapkan metode *backtracking*. Apabila basis masih terlalu besar, maka metode *Divide-and-Conquer* kembali diterapkan. Setelah itu, setiap basis dihubungkan dengan melakukan rotasi.



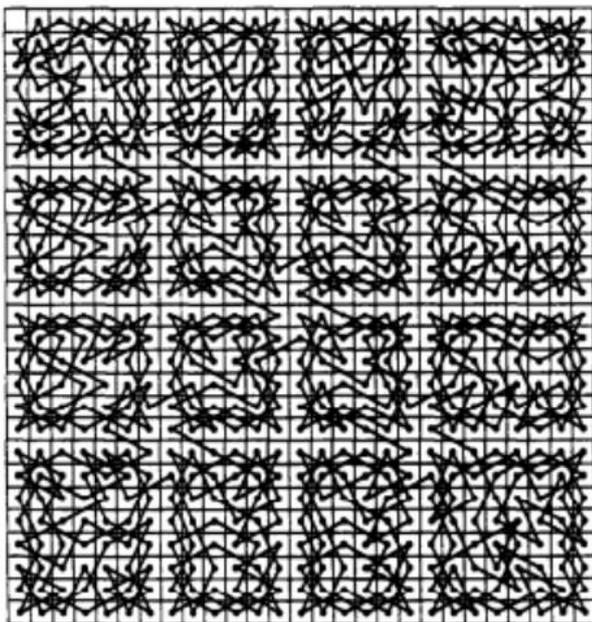
Gambar 5.1. Contoh menghubungkan basis-basis (*Ian Parbery, Discrete Applied Mathematics*)



Gambar 5.2. Contoh solusi *Knight's Tour* pada papan 16x16 dengan basis 8x8 (Ian Parbery, *Discrete Applied Mathematics*)



Gambar 5.4. Contoh solusi *Knight's Tour* pada papan 60x60 (larc.unt.edu)



Gambar 5.3. Contoh solusi *Knight's Tour* pada papan 27x27 (Ian Parbery, *Discrete Applied Mathematics*)

VI. KESIMPULAN

Terdapat banyak contoh graf Hamilton yang dapat kita temukan dalam kehidupan, salah satunya pada penerapan permainan *Knight's Tour* ini. Untuk menyelesaikan permainan ini, ada berbagai metode yang dapat digunakan. Dalam proses pembentukan graf Hamilton pada permainan ini dapat dibuat lebih mudah dengan menggunakan algoritma *Backtracking*. Algoritma ini terbukti lebih efisien daripada algoritma lain karena prinsip algoritma ini untuk mencegah singgah di tempat yang sama berkali-kali dengan membuang segala kemungkinan yang bukan merupakan bagian dari solusi. Pada Algoritma *Divide-and-Conquer* pun algoritma *backtracking* merupakan hal yang fundamental karena digunakan pada basis-basis algoritma. *Divide-and-Conquer*.

REFERENSI

- [1] Munir, Rinaldi, 2010, "Matematika Diskrit", Bandung: Informatika ITB.
- [2] Parbery, Ian. "Discrete Applied Mathematics" . NH Elsevier, Texas : 1997
- [3] <https://www.geeksforgeeks.org/the-knights-tour-problem-backtracking-1/> diakses pada tanggal 11 Desember 2020
- [4] https://id.wikipedia.org/wiki/Lintasan_Hamilton diakses pada tanggal 11 Desember 2020
- [5] <https://mathworld.wolfram.com/KnightsTour.html> diakses pada tanggal 11 Desember 2020

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, 11 Desember 2020

A handwritten signature in black ink, consisting of several loops and a long horizontal stroke extending to the right.

Bintang Fajarianto - 13519138