Penerapan Algoritma *Greedy* dan Warnsdorf's Rule dalam Puzzle Knight's Tour

Muhammad Rafli Fadillah - 13515115
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Bandung, Indonesia
13515115@std.stei.itb.ac.id

***Abstract*— Knight's Tour adalah sebuah puzzle yang cukup popular pada permainan catur. Puzzle ini mengharuskan sebuah pion kuda untuk melewati setiap kotak pada papan catur sebanyak tepat satu kali. Puzzle ini dapat menguras banyak waktu dan tenaga apabila diselesaikan secara manual atau menggunakan brute force terutama pada papan catur yang berukuran besar. Warnsdorff's Rule adalah salah satu metode heuristic yang dapat digunakan untuk menyelesaikan puzzle Knight's Tour. Cara ini mengharuskan pion kuda untuk melangkah ke kotak yang memiliki kemungkinan jalan (possible moves) paling sedikit. Algoritma Greedy adalah salah satu algoritma yang digunakan dalam pathfinding dimana Algoritma tersebut mengharuskan pengguna untuk menggunakan jalan yang paling menguntungkan pada saat itu. Ide atau prinsip dari Warnsdorff's Rule dan Algoritma Greedy dapat dibilang sejalan.**

Keywords—Knight's Tour, Warnsdorf's Rule, Greedy Algorithm

I. PENDAHULUAN

Catur adalah olahraga dan permainan strategi yang sangat populer di seluruh penjuru dunia. Salah satu alasannya adalah catur dapat dimainkan oleh semua orang, dari yang tua sampai yang muda. Permainan catur juga berguna untuk mengasah otak. Sampai sekarang, belum ada orang yang tahu pasti siapa penemu permainan ini. Beberapa literatur yang ditemukan seperti buku Kavyalankara yang ditulis oleh Rudrata menunjukkan bahwa pada abad ke-9 permainan catur sudah ada di daerah India. Permainan catur memiliki mekanisme yang unik dan mudah untuk dipahami. Seringkali, catur dijadikan bahan atau materi persoalan matematis sepert misal 8 Queens Puzzle dan Knight's Tour. Penulis akan membahas lebih lanjut tentang persoalan Knight's Tour.

Knight's Tour adalah sebuah persoalan matematis dimana sebuah pion *knight* (kuda) pada sebuah papan catur harus melewati setiap kotak pada papan catur sebanyak tepat satu kali. Persoalan ini akan sangatlah mudah apabila pion kuda berjalan secara lurus. Namun, pion kuda hanya dapat berjalan ke kotak yang berwarna berbeda dengan yang ditempati saat itu dan kotak tersebut tidak boleh bersebelahan dengan posisi awal sehingga pion kuda berjalan mirip dengan huruf "L". Meskipun aturan berjalannya aneh,pion kuda adalah salah satu pion yang dapat bergerak ke semua kotak pada papan catur selain pion raja, ratu, dan banteng.

Fig. 1. Contoh persoalan dan solusi Knight's Tour

Persoalan Knight's Tour ini bisa saja dicari penyelesaiannya menggunakan algoritma Brute Force. Namun, algoritma Brute Force tidak mangkus untuk memecahkan persoalan ini karena persoalan Knight's Tour memiliki kompleksitas $O(k^N)$ dimana k adalah konstanta dan N adalah ukuran papan. Hal ini berarti untuk menemukan solusi dari Knight's Tour pada papan berukuran 8x8 akan memerlukan waktu yang sangat lama. Ada banyak cara yang dapat dilakukan untuk menemukan solusi persoalan ini selain Brute Force seperti Diveide and Conquer, Neural Network, dll. Namun, penulis ingin menelusuri lebih jauh tentang pemecahan persoalan Knight's Tour menggunakan Algoritma Greedy.

II. DASAR TEORI

A. *Algoritma* Greedy

Algoritma *Greedy* merupakan algoritma yang menyelesaikan suuatu permasalahan dengan mencari nilai maksimum pada setiap langkahnya. Nilai maksimum yang

dicari disini adalah nilai maksimum lokal yaitu nilai maksimum yang mungkin didapat pada langkah tersebut. Sesuai namanya yaitu "greedy" yang berarti serakah atau tamak, algoritma ini menggunakan prinsip "take what you can get now".

Algoritma Greedy menggunakan dua jenis optimasi yaitu optimasi maksimum dan optimasi minimum. Optimasi maksimum digunakan apabila solusi yang dicari adalah solusi yang membutuhkan nilai sebesar-besarnya dan optimasi minimum digunakan untuk mencari solusi yang membutuhkan nilai sekecil-kecilnya.

Untuk diperhatikan, solusi yang dihasilkan oleh algoritma ini belum tentu hasil yang paling optimal. Algoritma Greedy digunakan dengan harapan hasil yang ditemukan pada setiap langkahnya akan mengarah kepada hasil yang paling optimal.

Ada 5 elemen yang dibutuhkan agar algoritma ini dapat berjalan yaitu himpunan kandidat, himpunan solusi, fungsi seleksi, fungsi layak, dan fungsi obyektif.

- Himpunan kandidat. Himpunan kandidat merupakan himpunan dari pilihan yang akan diambil dengan jumlah tertentu untuk menjadi solusi.
- Himpunan solusi. Himpunan solusi merupakan himpunan dari kandidat-kandidat yang telah dipilih.
- Fungsi seleksi. Fungsi seleksi merupakan fungsi yang digunakan untuk menentukan pilihan kandidat yang paling optimal dari kandidat yang ada.
- Fungsi layak. Fungsi layak adalah fungsi yang digunakan untuk memeriksa apakah solusi yang didapat dari hasil algoritma greedy adalah solusi yang memenuhi syarat atau valid.
- Fungsi obyektif. Fungsi obyektif merupakan fungsi lain yang diperlukan jika mampu mengoptimalkan solusi dari algoritma.

Meskipun algoritma ini tidak selalu menghasilkan solusi yang optimum, namun paling tidak solusi yang dihasilkan mendekati solusi yang optimum. Mekanisme dari algoritma ini juga sangatlah sederhana dan kerjanya sangatlah cepat.

B. *Knight's Tour*

Knight's Tour adalah sebuah puzzle yang dapat kita temukan pada permainan catur. Terdapat sebuah pion kuda pada sebidang papan catur dengan posisi tertentu. Pemain harus dapat menjalankan pion kuda sehingga pion kuda menyinggahi masing-masing kotak sebanyak tepat 1 kali. Papan yang biasa digunakan adalah papan 8x8. Namun, ukuran papan bisa saja diubah untuk menambah kompleksitas puzzle.

Ada 2 jenis Knight's Tour yaitu open dan closed. Closed Knight's Tours adalah Knight's Tours dimana pion kuda harus berangkat dari posisi awal, menyinggahi seluruh kotak pada papan, dan posisi terakhirnya dapat ditempuh dengan 1 gerakkan pion kuda dari tempat awal. Sederhananya, closed Knight's Tours adalah Knight's Tours yang posisi awal dan posisi akhirnya sama.

Sedangkan, open Knight's Tours adalah semua Knight's Tours yang bukan merupakan bagian dari closed Knight's Tours.

Persoalan Knight's Tour ini dapat kita temukan pada buku "Kavyalankara" yang ditulis oleh Rudrata pada abad ke-9. Namun, matematikawan yang pertama kali meneliti persoalan ini adalah Leonhard Euler.

Knight's Tour adalah salah satu contoh persoalan Hamiltonian Path pada Teori Graf. Pencarian solusi dari Knight's Tour dapat

C. *Warnsdorf's Rule*

Warnsdorf's Rule ada salah satu heuristic yang dapat digunakan untuk mencari solusi dari Knight's Tour. Aturan ini pertama kali dikemukakan oleh H. C. von Warndsdorf pada tahun 1823 dalam buku "Des Rösselsprungs einfachste und allgemeinste Lösung".

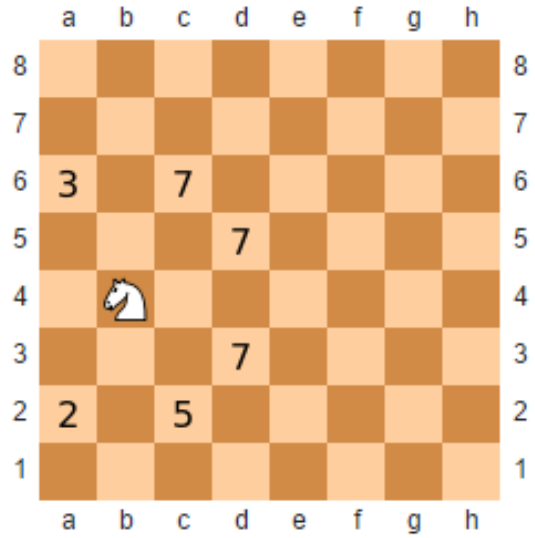

Fig. 2. Contoh penerapan Warnsdorf's Rule

Warnsdorf's Rule mengharuskan pion kuda untuk berjalan ke kotak dimana kotak tersebut memiliki kemungkinan jalan (possible moves) yang paling sedikit. Nilai kemungkinan terkecil adalah 0 dan nilai kemungkinan terbesar adalah 7 karena posisi awal pion kuda tidak dihitung ke dalam kemungkinan jalan. Dalam penerapannya, mungkin saja terdapat 2 atau lebih nilai kemungkinan yang sama di beberapa kotak. Untuk menyelesaikannya, kita dapat memilih secara acak jalan yang akan diambil dari beberapa kemungkinan jalan yang sama tersebut atau kita dapat menggunakan algoritma tie-breaking yang sudah ada.

Misal pada (Fig. 2) kita memiliki sebuah pion kuda yang berada di kotak B4. Maka untuk situasi ini pion kuda harus mengambil kotak dengan nilai kemungkinan jalan yang terkecil, yaitu dua.

Untuk situasi yang membutuhkan *tie-breaking*, contoh pada (Fig. 2). Pada situasi biasa tentu pion kuda akan digerakkan ke

kotak A2 karena kotak A2 memiliki kemungkinan jalan terkecil diantara kotak lainnya. Apabila kotak-kotak yang memiliki kemungkinan jalan kecil yaitu A2, A6, dan C2 tidak dapat diambil, maka kotak yang tersisa memiliki kemungkinan jalan yang sama satu sama lain yaitu 7. Situasi inilah situasi yang membutuhkan fungsi *tie-break*. Fungsi *tie-break* inilah yang akan dipakai untuk menentukan kotak mana yang akan diambil dari ketiga kotak tadi.

Perlu diperhatikan bahwa aturan ini tidak selalu menghasilkan solusi yang feasible. Ini dikarenakan belum ada yang bisa membuktikan bahwa Warnsdorf's Rule akan menghasilkan solusi dari *tour*. Ditambah lagi, metode *tie-breaking* yang dipakai tidak selalu mengarah pada solusi yang *feasible*.

III. PEMBAHASAN

A. *Implementasi Greedy*

Dari penjelasan yang ada di Bab II, penulis menyimpulkan bahwa Warnsdorf's Rule dapat diimplementasikan untuk pencarian solusi puzzle Knight's Tour. Penulis mencoba menuliskan pencarian solusi persoalan di atas menjadi program komputer dengan menggunakan algoritma Greedy.

Terdapat 5 elemen penting yang harus ada agar algoritma Greedy dapat berjalan. 5 elemen yang ada di program yang ditulis oleh penulis adalah :

- Himpunan kandidat. Himpunan kandidat adalah himpunan kotak-kotak yang sudah dihitung nilai kemungkinan jalannya. Himpunan kandidat ini memiliki rentang 0 sampai 7 elemen.
- Himpunan solusi. Himpunan solusi adalah himpunan jalan yang diambil oleh pion kuda.
- Fungsi seleksi. Fungsi seleksi yang digunakan adalah fungsi untuk memilih jalan yang memiliki nilai kemungkinan jalan terkecil.
- Fungsi layak. Fungsi layak yang digunakan adalah fungsi untuk memeriksa apakah solusi memenuhi syarat atau tidak. Syarat yang perlu dipenuhi disini adalah pion kuda menyinggahi setiap kotak pada papan catur sebanyak tepat satu kali.
- Fungsi obyektif. Fungsi obyektif adalah fungsi lain yang diperlukan jika mampu mengoptimalkan solusi. Solusi yang dicari pada program ini tidak terlalu mementingkan optimalnya solusi. Hal ini disebabkan oleh tipe program yang hanya memastikan bahwa untuk suatu ukuran papan tertentu dan dengan posisi pion kuda awal tertentu terdapat serangkaian gerak yang merupakan solusi dari puzzle Knight's Tour.

Program tergolong sebagai algoritma Greedy dengan optimasi minimum karena elemen yang dimasukkan ke dalam himpunan solusi adalah elemen yang memiliki kemungkinan jalan terkecil (minimum).

Selain 5 elemen penting algoritma Greedy, program juga memerlukan suatu fungsi tie-breaking untuk memilih jalan apabila terdapat 2 atau lebih kandidat jalan yang memiliki kemungkinan jalan yang sama. Penulis tidak menggunakan fungsi melainkan menetapkan perjanjian jalan seperti pada gambar di bawah ini.

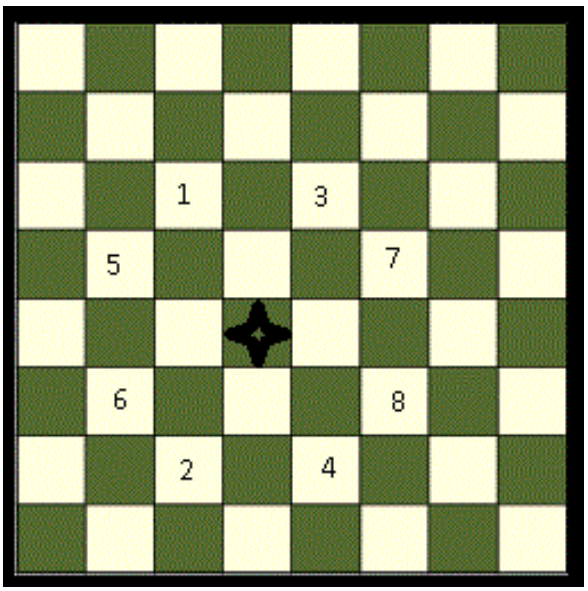

Fig. 3. Cara pengambilan jalan yang digunakan.

Pengambilan jalan yang digunakan menggunakan sistem pengambilan indeks paling kecil. Misal terdapat kesamaan kemungkinan jalan di 2 kotak yaitu kotak C6 dan kotak F5 (Fig. 3), maka jalan atau kotak yang akan dituju adalah kotak C6 karena nilai indeksnya lebih kecil yaitu 1 dibanding dengan kotak F5 yang nilai indeksnya adalah 3.

B. *Cara Kerja Program*

Program akan menerima input berupa 3 buah bilangan bulat (integer). Integer pertama adalah ukuran papan. Sedangkan integer kedua dan ketiga adalah posisi awal pion kuda. Misal program menerima input "8 1 1", maka program akan menciptakan papan catur berukuran 8 x 8 dan menempatkan pion kuda di koordinat 1,1 (range koordinat adalah 1 sampai ukuran papan).

Setelah papan catur dibuat dan pion kuda diletakkan, program akan memulai pencarian tour. Setelah pencarian selesai, program akan mengeluarkan pesan gagal atau berhasil dan menampilkan papan catur terakhir yang didapat dari pencarian. Cara kerja pencarian secara singkat adalah sebagai berikut :

1) Memeriksa apakah semua kotak sudah disinggahi. Jika belum, lanjut ke langkah 2). Apabila sudah, loncat ke langkah 5).

2) Mencari langkah dengan nilai kemungkinan jalan terkecil. Apabila ada 2 atau lebih jalan yang memiliki nilai kemungkinan jalan yang sama, gunakan perjanjian jalan.

Apabila tidak ada jalan yang ditemukan, lanjutkan ke langkah 4).

3) Pindahkan pion kuda ke kotak yang dihasilkan pada langkah 2). Simpan langkah dan posisi pion kuda. Ulangi ke langkah 1).

4) Tampilkan papan catur terakhir. Keluarkan pesan bahwa tour tidak dapat ditemukan. Lanjutkan ke langkah 6)

5) Tampilkan papan catur terakhir. Keluarkan pesan tour dapat ditemukan. Lanjutkan ke langkah 6).

6) Terminasi program.

IV. PENGUJIAN

Pengujian awal dilakukan dengan mencoba menyelesaikan puzzle Knight's Tour standar yaitu pada papan catur berukuran 8x8. Pengujian dilakukan dengan menggunakan komputer dengan processor i5-5200U 2,2 Ghz 64-bit, cache 3 MB, dan RAM 4 GB.

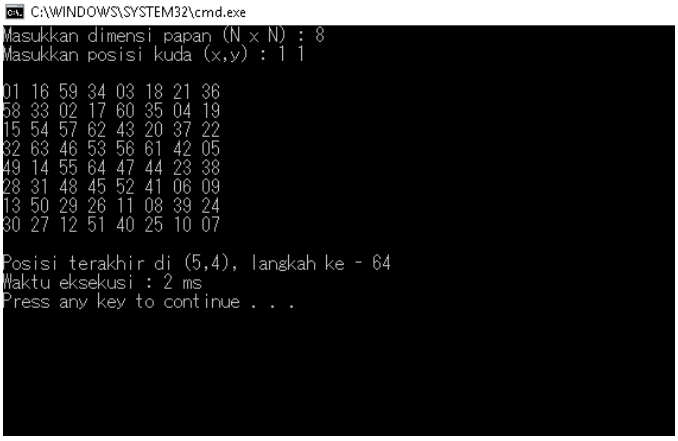

Fig. 4. Pengujian untuk Knight's Tour standar.

Fig. 5. Pengujian untuk Knight's Tour standar.

Setelah melakukan pengujian dengan menyelesaikan Knight's Tour standar. Timbul rasa ingin tahu dari penulis untuk melakukan pengujian pada papan catur dengan ukuran lain.

Pengujian dilakukan kembali dengan ukuran papan catur 5 x 5. Beberapa pengujian berlangsung dengan lancar. Namun, penulis menemukan kombinasi input yang tidak dapat ditemukan solusi tour-nya.

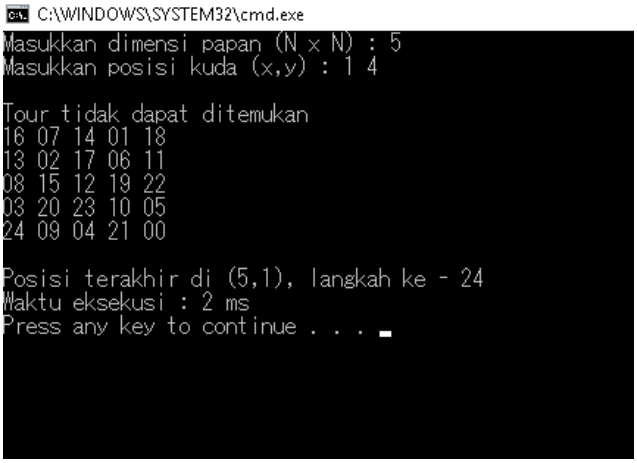

Fig. 6. Pengujian untuk Knight's Tour 5x5 di posisi awal 1,4.

Fig. 7. Pengujian untuk Knight's Tour 5x5 di posisi awal 1,5.

Setelah itu, penulis memutuskan untuk menguji program pada berbagai ukuran antara 5 – 60 dan pada setiap posisi awal pion kuda yang ada. Berikut adalah tabel rangkuman dari hasil pengujian yang didapat oleh penulis.

TABLE I. TABEL RANGKUMAN HASIL PENGUJIAN

N	Hasil Pengujian		
	Jumlah Pengujian	***Jumlah Gagal***	***Waktu Eksekusi***
2	4 kali	4 kali	*0 ms
3	9 kali	9 kali	*0 ms
4	16 kali	16 kali	*0 ms
5	25 kali	16 kali	5 ms
6	36 kali	0 kali	26 ms
7	49 kali	31 kali	19 ms
8	64 kali	1 kali	90 ms
9	81 kali	41 kali	77 ms
10	100 kali	2 kali	226 ms
11	121 kali	60 kali	224 ms
12	144 kali	1 kali	498 ms
13	169 kali	84 kali	389 ms
14	196 kali	4 kali	928 ms
15	225 kali	113 kali	710 ms
20	400 kali	10 kali	4193 ms
40	1600 kali	251 kali	64169 ms
60	3600 kali	1369 kali	261957 ms

V. KESIMPULAN

Warnsdorf's Rule adalah aturan yang sangat efektif untuk mencari solusi dari puzzle Knight's Tour standar. Hal ini terbukti dengan tingkat keberhasilan program yang sangat tinggi (98%) di papan catur berukuran 8x8. Algoritma ini juga tergolong cepat karena rata-rata waktu pencarian solusinya tidak mencapai 1 detik.

Program ini juga memiliki beberapa kelemahan. Programtidak dapat mencari jalur tour untuk papan catur berukuran lebih kecil dari 5. Hal ini dibuktikan dengan tingkat kegagalan 100% pada pengujian dengan papan catur berukuran 2,3 dan 4. Untuk papan catur berukuran 2, sudah jelas tidak akan dapat ditemukan solusinya karena untuk melakukan 1 langkah, pion kuda membutuhkan 1 kotak ke samping dan 2 kotak ke depan. Dengan kata lain, pion kuda tidak dapat melangkah di papan catur berukuran 2x2. Untuk papan catur berukuran 3 dan 4, penulis belum bisa memberikan alasan yang tepat.

Selain itu, program juga memiliki tingkat kegagalan yang cukup tinggi (50%~) saat dilakukan pengujian pada papan berukuran ganjil. Untuk kelemahan ini, penulis belum bisa memberikan alasan yang konkrit.

Waktu eksekusi terlihat lebih rendah untuk ukuran papan ganjil dikarenakan, apabila solusi *tour* tidak ditemukan maka waktu eksekusi pencarian tersebut tidak dihitung ke dalam waktu eksekusi total. Karena itu, waktu eksekusi yang tercantum di Tabel 1 adalah waktu eksekusi yang dibutuhkan untuk mencari solusi (*Jumlah Pengujian – Jumlah Gagal)* posisi awal pion kuda.

Namun, program ini dapat dibilang mencapai tujuan pembuatannya meskipun tidak 100%. Hal ini dikarenakan program dapat menemukan solusi tour untuk Knight's Tour pada papan berukuran tertentu selama ukuran tidak terlalu kecil (n >= 5) dan tidak terlalu besar (n <= 60).

VI. PESAN PENULIS

Untuk penelitian lebih lanjut, penulis menyarankan untuk memusatkan penelitian pada fungsi tie-breaking atau mencoba menggunakan strategi algorima lain selain Algoritma Greedy.

Fungsi tie-breaking sangatlah penting. Hal ini dikarenakan fungsi ini yang nantinya akan mengarahkan pencarian ke arah yang diharapkan akan menghasilkan solusi. Penulis tidak mendalami fungsi tie-breaking lebih lanjut dikarenakan penulis belum menemukan materi yang cocok untuk diimplementasikan sebagai fungsi¬ tie-breaking.

Penulis berharap makalah ini dapat membantu penelitian lain yang mungkin dapat menciptakan sebuah fungsi tie-breaking yang efektif untuk program pencarian solusi Knight's Tour agar program dapat berkembang.

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa, Allah SWT, karena atas berkat dan karunia-Nya penulis dapat menyelesaikan makalah ini dengan tepat waktu. Penulis mengucapkan terima kasih kepada dosen pengajara matakuliah IF2211 Strategi Algoritma, Dr. Ir. Rinaldi Munir, MT. dan Masayu Leylia Khodra, ST., MT. yang telah memberikan kesempatan kepada penulis dan teman-teman untuk menuliskan makalah ini sebagai kontribusi dan tugas dalam memberikan kebaikan kepada dunia.

REFERENSI

[1] Chaudhary Satyadev, *Kavyalankara of Rudrata (Sanskrit Text, with Hindi translation).* Delhitraversal : Parimal Sanskrit Series No.30.

[2] Karla Alwan, K. Waters, Finding Re-entrant Knight's Tours on N-by-M Boards. New York : ACM, 1992, pp. 377-382.

[3] Rinaldi Munir, Strategi Algoritma. Bandung : Penerbit Institut Teknologi Bandung, 2009.

[4] Edward D. Collins, A Knight's Tour. http://www.edcollins.com/chess/knights-tour.htm . Diakses pada 19 Mei 2017, 13.00 WIB.

[5] Manoel Ribeiro, How to get to the solution faster. https://codereview.stackexchange.com/questions/79995/optimizing-this-knights-tour . Diakses pada 19 Mei 2017, 13.05 WIB

[6] Unknown, Knight's Tour Analysis. http://interactivepython.org/runestone/static/pythonds/Graphs/KnightsTourAnalysis.html . Diakses pada 19 Mei 2017, 13.08 WIB

[7] Unknown, Warnsdorff's algorithm for Knight's tour problem. http://www.geeksforgeeks.org/warnsdorffs-algorithm-knights-tour-problem/ .Diakses pada 19 Mei 2017, 13.11 WIB.

[8] Douglas Squirrel, P. Cull, A Warnsdorff-Rule Algorithm for Knight's Tours on Square Boards. 1992, https://github.com/douglassquirrel/warnsdorff/blob/master/5_Squirrel96.pdf?raw=true .

[9] Martin Loebbing, Ingo Wegener, The Numbe of Knight's Tours Equals 33,439,123,484,294 – Counting with Binary Decision Diagrams. The Electronic Journal of Combinatorics.

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, 19 Mei 2017

Muhammad Rafli Fadillah / 13515115