

Penerapan Teori Lintasan Hamilton untuk menyelesaikan Persoalan *Knight's Tour* pada Permainan Catur dengan *Backtracking*

Fathan Mubina - 13518064
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518064@std.stei.itb.ac.id

Abstract— Permainan Catur adalah sebuah permainan papan yang mengandalkan strategi. Dalam permainan ini pemain berusaha mengalahkan lawan dengan membuat lawan dalam kondisi skakmat dengan menggunakan bidak-bidak secara bergantian. Bidak dalam permainan ini terdapat enam jenis yaitu raja, menteri, gajah, kuda, benteng, dan pion. Masing-masing bidak memiliki pola pergerakan tertentu. Kuda memiliki pergerakan yang unik yaitu membentuk huruf L. Pergerakan kuda untuk melewati semua kotak pada papan catur tepat satu kali yang inilah yang disebut Persoalan *Knight's Tour*.

Persoalan *Knight's Tour* juga merupakan salah satu persoalan lintasan hamilton dalam teori graf. Dalam makalah ini akan dibahas penyelesaian persoalan *Knight's Tour* dengan memanfaatkan teori graf yaitu lintasan hamilton dengan algoritma *Backtracking*.

Kata kunci— Catur, kuda, graf, lintasan hamilton, *Knight's Tour*, rekursi, *Backtracking*.

I. PENDAHULUAN

Permainan catur adalah permainan papan bergenre strategi yang dimainkan oleh dua orang. Catur biasanya dimainkan di atas papan kotak-kotak yang terdiri dari 64 kotak, yang disusun dalam ukuran 8x8 kotak, yang terbagi menjadi dua warna yaitu warna hitam dan putih secara berselingan.



Gambar 1.1. Permainan Catur
(sumber : sejarahlengkap.com)

Sejarah mencatat Catur mulai dimainkan pada abad ke-7 di India. Di India catur melambungkan tentang alam semesta yang terbagi menjadi ke empat unsur yaitu api, udara, tanah dan air. Karena itu lah dinamakan Caturanga memiliki makna 'empat unsur yang terpisah'. Kemudian pedagang Islam dari India membawa permainan ini menuju ke Persia. Disana catur disebut dengan nama shatranj di Sassanid. Kemudian catur juga mulai dikenal di seluruh penjuru dunia. Disini catur juga mengalami perkembangan dari segi permainan sehingga lebih menarik dan lebih seru hingga menjadikan catur sebagai permainan rekreasi paling favorit di Persia. Karena keseruan ini kemudian catur menyebar lagi hingga ke daratan Arab dan akhirnya ke seluruh dunia hingga saat ini.

Cara bermainnya adalah dengan menggerakkan buah catur. Pada mulanya, setiap pemain memiliki 16 buah catur, yaitu satu raja (*king*), satu menteri (*queen* atau ratu), dua benteng (*rook*), dua kuda (*knight* atau kesatrian), dua gajah (*bishop* atau uskup), dan delapan pion. Setiap jenis buah catur memiliki pergerakan tertentu, dengan yang paling kuat adalah ratu dan yang paling lemah adalah pion.

Raja dapat bergerak satu persegi ke segala arah. Raja juga memiliki gerakan khusus yang disebut rokade yang turut melibatkan sebuah benteng. Benteng dapat bergerak sepanjang persegi secara horizontal maupun vertical, tetapi tidak dapat melompati bidak lain. Gajah dapat bergerak sepanjang persegi secara diagonal, tetapi tidak dapat melompati buah catur lain. Ratu memiliki gerakan kombinasi dari benteng dan gajah. Kuda memiliki gerakan mirip huruf L, yaitu memanjang dua kotak dan melebar satu kotak. Kuda lah satu-satunya buah catur yang dapat melompati buah catur lain. Pion dapat bergerak maju (arah lawan) satu kotak ke kotak yang tidak ditempati. Pada gerakan awal, pion dapat bergerak maju dua persegi. Pion juga dapat menangkap buah catur lawan secara diagonal, apabila buah catur lawan tersebut berada satu kotak di diagonal depannya.

Tujuan dari permainan ini adalah melakukan skakmat pada raja lawan dengan menempatkan buah tersebut pada posisi di mana ancaman untuk ditangkap atau dimakan tidak dapat

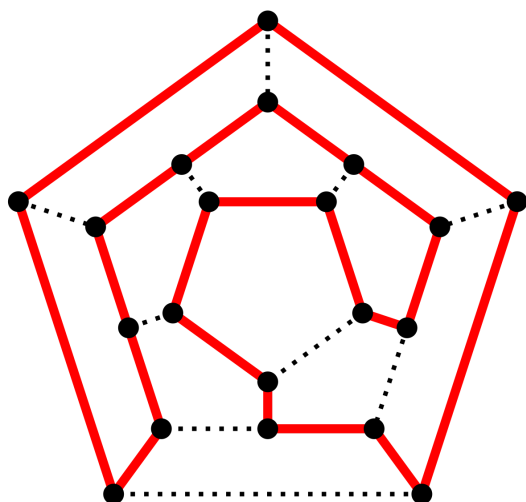
dihindari. Untuk mencapai tujuan ini, buah-buah milik pemain harus digunakan untuk menyerang dan menangkap buah lawan, sambil buah-buah tersebut tetap menjaga satu sama lain. Selama permainan berlangsung, jalan permainan biasanya melibatkan makan memakan buah dengan buah lawan, serta penemuan dan perekayasaan peluang untuk melakukan pertukaran secara menguntungkan atau untuk mendapatkan posisi yang lebih baik. Selain skakmat, seorang pemain memenangkan permainan jika pemain lawan menyerah, atau kehabisan waktu dalam permainan catur cepat atau berwaktu. Ada juga beberapa cara yang membuat permainan berakhir dalam keadaan remis atau seri.

Perkembangan permainan catur dari abad ke abad memunculkan banyak perkembangan dan variasi permainan catur dan muncul pula berbagai persoalan seperti persoalan *N-Queen* dan persoalan *Knight Tour*. Persoalan *Knight's Tour* Pergerakan kuda untuk melewati semua kotak pada papan catur tepat satu kali. Terdapat beberapa cara yang dapat digunakan untuk menyelesaikan permasalahan ini diantaranya adalah dengan algoritma *Divide and Conquer*, algoritma *Backtracking*, aturan *Warnsdorff*, dan metode *De Moivre*.

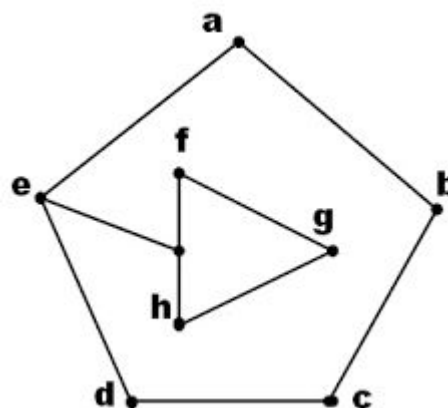
II. LINTASAN HAMILTON

Pada tahun 1859, Matematikawan dari Irish, Sir William Rowan Hamilton mengembangkan permainan yang di beli dari perusahaan mainan di Dublin. Permainan itu dinamakan *Prominent Cities*. Tujuan dari permainan itu adalah mencari sirkuit sepanjang jalan yang terbentuk sehingga di dalam itu terdapat 20 kota dan dapat dilewati tepat satu kali dengan membentuk jalur sehingga melewati setiap titik tepat satu kali. Masalah ini kemudian diselesaikan dengan menggunakan teori Graf dengan setiap titik tersebut diibaratkan sebagai simpul dan hubungan antar titik diibaratkan sebagai sisi.

Lintasan Hamilton adalah lintasan yang melalui tiap simpul di dalam graf tepat satu kali. Bila lintasan itu kembali ke simpul asal membentuk lintasan tertutup yang dinamakan sirkuit Hamilton.



Gambar 2.1. Sirkuit Hamilton
(sumber: id.wikipedia.org)



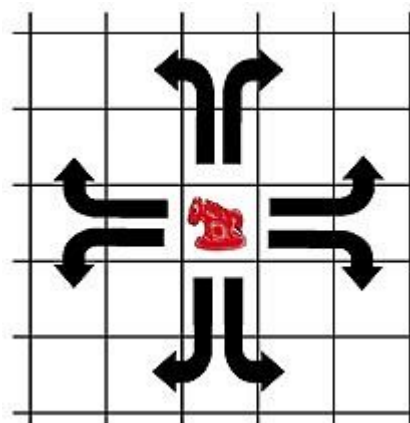
Gambar 2.2. Lintasan Hamilton
(Sumber: <http://kuliahmsi.blogspot.com>)

Beberapa teorema terkait Sirkuit dan Lintasan Hamilton:

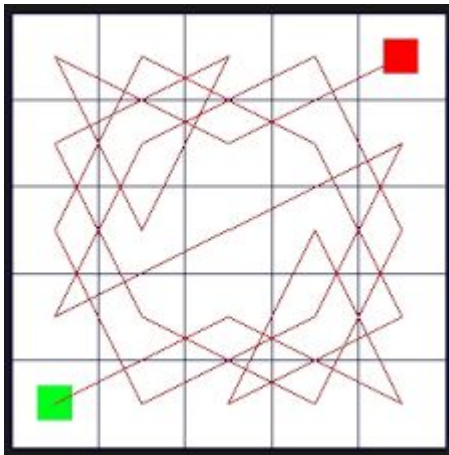
1. Syarat cukup supaya graf sederhana G dengan n buah simpul ($n \geq 3$) dikatakan graf Hamilton adalah jika derajat tiap simpul paling sedikit $n/2$.
2. Setiap graf lengkap adalah graf Hamilton.
3. Dalam graf lengkap dengan n buah simpul ($n \geq 3$), terdapat $(n-1)!/2$ buah sirkuit Hamilton.
4. Dalam graf lengkap dengan n buah simpul ($n \geq 3$) dan n ganjil, terdapat $(n-1)/2$ buah sirkuit Hamilton yang saling lepas.
5. Dalam graf lengkap dengan n buah simpul ($n \geq 4$) dan n genap, terdapat $(n-2)/2$ buah sirkuit Hamilton.

III. PERSOALAN KNIGHT'S TOUR

Persoalan *Knight's Tour* merupakan sebuah persoalan menggunakan kuda dengan cara melewati seluruh kotak yang ada tepat satu kali pada sebuah papan. Permainan selesai ketika seluruh kotak pada papan catur sudah dilewati oleh kuda dengan gerakan seperti dalam permainan catur yaitu membentuk huruf L.



Gambar 3.1. Gerak buah catur kuda
(sumber: mcmahel.blogspot.com)

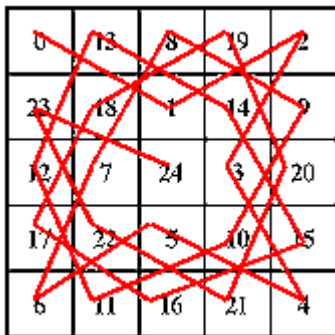


Gambar 3.2. Lintasan dari Knight's Tour
(sumber: ibmathsresources.com)

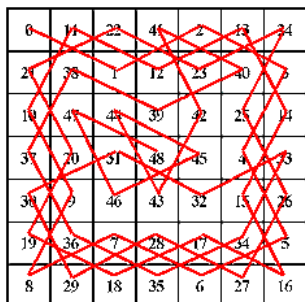
Ada dua jenis permainan The Knight's Tour yaitu Closed Tour dan Open Tour. Closed Tour yaitu kondisi dimana posisi awal buah catur kuda juga harus menjadi posisi akhir buah catur kuda setelah menempati semua petak 1 kali. Yang kedua yaitu Open Tour yaitu kondisi dimana posisi awal buah catur kuda dan posisi akhir buah catur kuda tidak harus sama.

Persoalan The Knight's Tour sama seperti menyelesaikan persoalan lintasan Hamilton. Kotak pada papan catur dapat diibaratkan sebagai simpul, sedangkan pilihan jalannya kuda dapat diibaratkan sebagai sisi. Persoalan lintasan Hamilton adalah bagaimana melewati semua simpul tepat satu kali, sama seperti permainan The Knight's Tour yaitu menempati setiap petak tepat satu kali.

A. Persoalan Knight's Tour pada papan catur $n \times n$

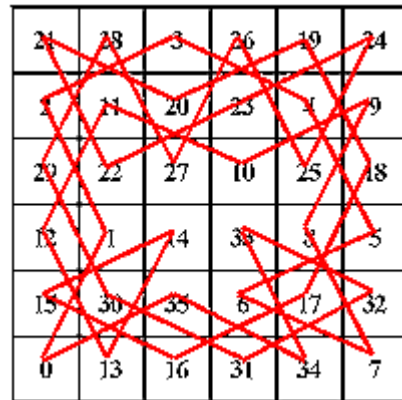


Gambar 3.3. Lintasan Knight's Tour pada papan catur 5 x 5
(sumber: www.markkeen.com)

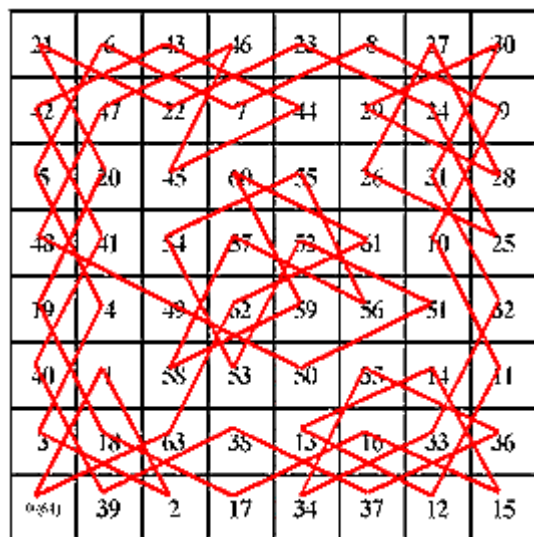


Gambar 3.4. Lintasan Knight's Tour pada papan catur 5 x 5
(sumber: www.markkeen.com)

Lintasan pada gambar 3.3. dan gambar 3.4 merupakan open knight's tour karena buah catur kuda tidak dapat kembali ke posisi awalnya. Tidak pernah ditemukan adanya closed knight's tour pada papan catur $n \times n$ dengan n ganjil. Kuda selalu berpindah dari satu warna ke warna yang lain. Pada saat n ganjil, jumlah kedua warna pada papan catur tentu saja berbeda maka closed knight's tour tidak pernah terjadi pada papan catur berukuran $n \times n$ dengan n ganjil.



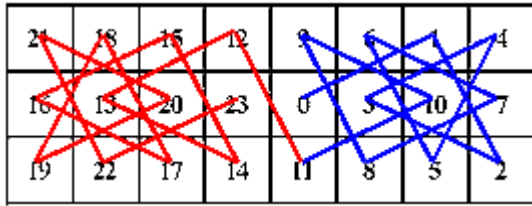
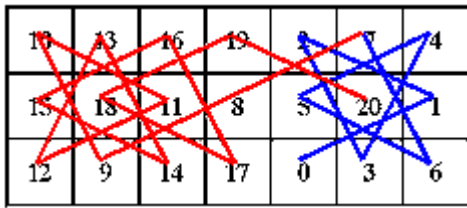
Gambar 3.5. Lintasan Knight's Tour pada papan catur 6 x 6
(sumber: www.markkeen.com)



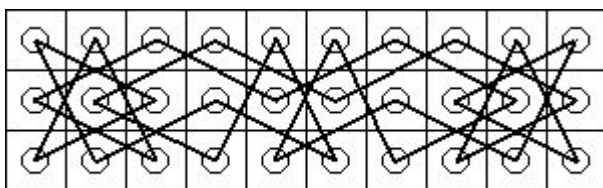
Gambar 3.6. Lintasan Knight's Tour pada papan catur 8 x 8
(sumber: www.markkeen.com)

Pada gambar 3.5, terlihat jelas bahwa buah catur kuda dapat kembali ke posisi awalnya (dari posisi 35 ke posisi 0). Hal ini menunjukkan bahwa closed knight's tour dapat terjadi pada papan catur berukuran 6x6 begitu juga pada gambar 3.6 dapat kembali ke posisi awalnya (dari posisi 63 ke posisi 0).

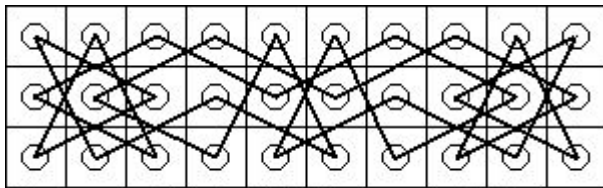
B. Persoalan Knight's Tour pada papan catur $m \times n$



Gambar 3.4. Lintasan Knight's Tour pada papan catur 3 x 7 dan 3 x 8
(sumber: www.markkeen.com)



By Tsu-wang Yang Aug.12.1999 amosyang@yahoo.com



By Tsu-wang Yang Aug.12.1999 amosyang@yahoo.com

Gambar 3.4. Lintasan Knight's Tour pada papan catur 3 x 10 dan 3 x 12
(sumber: www.markkeen.com)

Pada gambar-gambar diatas, dapat dilihat bahwa pada papan catur tersebut hanya dapat terjadi open knight's tour. Tidak dapat terjadi closed knight's tour karena bidak catur tidak dapat lagi kembali ke posisi awalnya.

Pada papan catur berukuran 1x4, jelas tidak terdapat knight's tour karena bahkan bidak kuda tidak dapat bergerak. Pada papan catur berukuran 2x4, knight's tour karena bahkan bidak kuda hanya dapat bergerak sekali lalu terhenti. Sehingga papan catur 3x4 merupakan papan catur berukuran minimal yang memiliki jalur knight's tour.

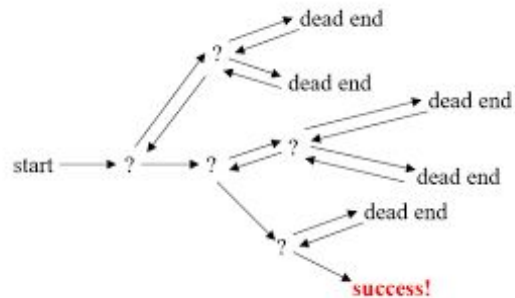
IV. BACKTRACKING

Algoritma backtracking pertama kali diperkenalkan oleh D. H. Lehmer pada tahun 1950. Algoritma backtracking merupakan pengembangan dari algoritma brute force yang algoritma tersebut mencoba semua kemungkinan bahan yang sebenarnya tidak perlu. Algoritma backtracking dibuat untuk membuang semua kemungkinan yang tidak perlu pada algoritma brute force. Algoritma backtracking banyak digunakan dalam

berbagai permasalahan seperti tic-tac-toe, mazeMath, Knight's Tour, 8 Queen, dan permasalahan artificial intelligence.

Algoritma backtracking bekerja dengan membentuk lintasan dari akar ke daun. Aturan yang digunakan adalah aturan DFS (Depth First Search). Simpul-simpul yang telah dihasilkan dinamakan simpul hidup (live node), sedangkan simpul yang diperluas dinamakan simpul E (expand node). Simpul yang sudah tidak dipakai lagi dinamakan simpul mati (dead node).

Algoritma backtracking akan mencari solusi parsial dari sebuah simpul tertentu dan kemudian simpul tersebut diperluas. Algoritma backtracking akan membuang simpul jika lintasan yang terbentuk tidak mengarah kepada solusi dan menjadi simpul mati. Untuk membuang simpul tersebut digunakan fungsi pembatas (bounding function). Jika pada saat pencarian berakhir pada simpul mati, maka pencarian akan dilakukan pada simpul anak lainnya. Akan tetapi, jika sudah tidak ada lagi simpul anak yang belum dicari, maka akan dilakukan backtracking ke simpul sebelumnya. Pencarian akan berhenti dilakukan ketika sudah didapatkan solusi atau jika sudah tidak bisa melakukan backtracking lagi yaitu tidak ada simpul hidup (live node) lagi.



Gambar 4.1. Ilustrasi Backtracking
(sumber: irsyadf.my.id)

Langkah-langkah backtracking pada persoalan The Knight'sTour yaitu:

1. Awalnya kuda ditempatkan, lalu mendata semua langkah-langkah yang mungkin dilalui oleh kuda dari penempatan kuda pertama kali.
2. Mengambil salah satu langkah yang merupakan simpul hidup (live node) lalu langkah tersebut diperluas lagi. dan menempatkan kuda pada petak yang telah dipilih.
3. Mengulangi langkah pertama sampai kedua untuk petak yang sedang ditempati.
4. Jika belum ditemukan solusi maka kembali ke langkah sebelumnya (backtracking) dengan menjadikan simpul mati pada simpul sebelumnya.
5. Pencarian dihentikan ketika solusi telah ditemukan atau tidak ada lagi langkah yang memungkinkan semua menjadi simpul mati (dead node).

Berikut adalah Algoritma Backtracking untuk menyelesaikan persoalan Knight's Tour pada papan catur 8 x 8 yang ditulis dalam bahasa C.


```
// C program for Knight Tour problem
#include<stdio.h>
#define N 8

int solveKTUtil(int x, int y, int movei, int sol[N][N],
               int xMove[], int yMove[]);

/* A utility function to check if i,j are valid indexes
for N*N chessboard */
int isSafe(int x, int y, int sol[N][N])
{
    return ( x >= 0 && x < N && y >= 0 &&
            y < N && sol[x][y] == -1);
}

/* A utility function to print solution matrix sol[N][N] */
void printSolution(int sol[N][N])
{
    for (int x = 0; x < N; x++)
    {
        for (int y = 0; y < N; y++)
            printf(" %2d ", sol[x][y]);
        printf("\n");
    }
}
```

```
/* A recursive utility function to solve Knight Tour
problem */
int solveKTUtil(int x, int y, int movei, int sol[N][N],
               int xMove[], int yMove[])
{
    int k, next_x, next_y;
    if (movei == N*N)
        return 1;

    /* Try all next moves from the current coordinate x, y */
    for (k = 0; k < 8; k++)
    {
        next_x = x + xMove[k];
        next_y = y + yMove[k];
        if (isSafe(next_x, next_y, sol))
        {
            sol[next_x][next_y] = movei;
            if (solveKTUtil(next_x, next_y, movei+1, sol,
                           xMove, yMove) == 1)
                return 1;
            else
                sol[next_x][next_y] = -1; // backtracking
        }
    }

    return 0;
}
```

```
/* This function solves the Knight Tour problem using
Backtracking. This function mainly uses solveKTUtil()
to solve the problem. It returns false if no complete
tour is possible, otherwise return true and prints the
tour.
Please note that there may be more than one solutions,
this function prints one of the feasible solutions. */
int solveKT()
{
    int sol[N][N];

    /* Initialization of solution matrix */
    for (int x = 0; x < N; x++)
        for (int y = 0; y < N; y++)
            sol[x][y] = -1;

    /* xMove[] and yMove[] define next move of Knight.
xMove[] is for next value of x coordinate
yMove[] is for next value of y coordinate */
    int xMove[8] = { 2, 1, -1, -2, -2, -1, 1, 2 };
    int yMove[8] = { 1, 2, 2, 1, -1, -2, -2, -1 };

    // Since the Knight is initially at the first block
    sol[0][0] = 0;

    /* Start from 0,0 and explore all tours using
solveKTUtil() */
    if (solveKTUtil(0, 0, 1, sol, xMove, yMove) == 0)
    {
        printf("Solution does not exist");
        return 0;
    }
    else
        printSolution(sol);

    return 1;
}
```

```
/* Driver program to test above functions */
int main()
{
    solveKT();
    return 0;
}
```

(Sumber: <https://www.geeksforgeeks.org/the-knights-tour-problem-backtracking-1/>)

setelah dijalankan akan menghasilkan,

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 0 | 59 | 38 | 33 | 30 | 17 | 8 | 63 |
| 37 | 34 | 31 | 60 | 9 | 62 | 29 | 16 |
| 58 | 1 | 36 | 39 | 32 | 27 | 18 | 7 |
| 35 | 48 | 41 | 26 | 61 | 10 | 15 | 28 |
| 42 | 57 | 2 | 49 | 40 | 23 | 6 | 19 |
| 47 | 50 | 45 | 54 | 25 | 20 | 11 | 14 |
| 56 | 43 | 52 | 3 | 22 | 13 | 24 | 5 |
| 51 | 46 | 55 | 44 | 53 | 4 | 21 | 12 |

V. KESIMPULAN

Teori Graf adalah cabang ilmu dari matematika diskrit yang memiliki banyak aplikasi dalam berbagai permasalahan salah satunya adalah persoalan Knight's Tour. Begitu juga dengan backtracking yang merupakan algoritma dengan metode Rekursi. Dengan analisis graf dalam masalah ini Lintasan Hamilton dengan algoritma Backtracking persoalan Knight's Tour dapat ditemukan solusinya.

VI. UCAPAN TERIMA KASIH

Penulis makalah ini mengucapkan syukur kepada Allah Subhanahu wata'ala dan mengucapkan terima kasih kepada bapak dan ibu dosen pengampu mata kuliah IF2120 Matematika Diskrit khususnya Dr. Ir. Rinaldi Munir, M.T dan semua pihak yang telah ikut membantu dalam menyelesaikan pembuatan makalah ini.

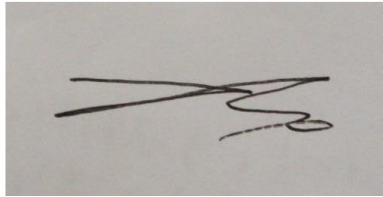
REFERENSI

- [1] Munir, Rinaldi, "Matematika Diskrit", Informatika, Bandung: 2015.
- [2] <http://asalmula-permainancatur.blogspot.co.id/> diakses pada tanggal Desember 2019
- [3] <https://www.geeksforgeeks.org/the-knights-tour-problem-backtracking-1/> diakses pada tanggal 6 Desember 2019
- [4] <https://ibmathsresources.com/2013/11/19/knights-tour/> diakses pada tanggal 6 Desember 2019
- [5] <https://www.programming-algorithms.net/article/39907/Knight%27s-tour> diakses pada tanggal 6 Desember 2019
- [6] <https://www.brainbashers.com/knight.asp> diakses pada tanggal 6 Desember 2019
- [7] gaebler.us/share/Knight_tour.html diakses pada tanggal 6 Desember 2019
- [8] <http://www.markkeen.com/knight/> diakses pada tanggal 6 Desember 2019
- [9] <https://sejarahlengkap.com/olahraga/sejarah-catur> diakses pada tanggal 6 Desember 2019
- [10] <https://gurupenjaskes.com/olahraga-catur-sejarah-aturan-dan-cara-bermainnya> diakses pada tanggal 6 Desember 2019

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, 6 Desember 2019

A photograph of a handwritten signature in black ink on a light-colored background. The signature is stylized and appears to read 'Fathan Mubina'.

Fathan Mubina - 13518064