Aplikasi Rekursifitas dan Teori Graf dalam Algoritma Penyelesaian Teka-Teki Sudoku

Michelle Eliza Gananjaya - 13516015
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516015@std.stei.itb.ac.id

***Abstract*—Sudoku adalah sebuah teka-teki logika yang mulai populer sejak tahun 1986 di Jepang karena menarik dan dapat dimainkan oleh orang-orang dari berbagai kalangan dan usia. Untuk orang yang menginginkan tantangan, sudoku juga memiliki berbagai tingkat kesulitan. Kebanyakan orang mencari solusi dari teka-teki ini secara manual, namun untuk mempermudah, teka-teki ini dapat diselesaikan dengan berbagai cara lain, di antaranya dengan menggunakan algoritma yang berbasis pada kombinatorial dan pewarnaan graf. Pewarnaan graf sendiri merupakan pemberian warna terhadap simpul-simpul graf sehingga setiap simpul yang bersebelahan atau berdampingan memiliki warna yang berbeda.**

***Keywords*—sudoku, graf, pewarnaan graf, algoritma, *backtracking*.**

I. PENDAHULUAN

Sudoku adalah teka-teki yang berbentuk matriks berukuran 9x9 yang kemudian terbagi lagi menjadi upamatriks berukuran 3x3. Setiap elemen di matriks tersebut harus diisi dengan angka satu sampai sembilan, dan setiap elemen dalam setiap upamatriks harus berbeda satu sama lain, begitu pula dengan setiap elemen dalam setiap baris dan setiap kolom. Pada kondisi awal teka-teki, terdapat beberapa angka yang sudah diisikan dalam matriks. Jumlah dan posisi angka-angka yang telah diisi ini dapat menentukan tingkat kesusahan dari teka-teki sudoku yang dimainkan.

		6		5	4	9		
1				6			4	2
7				8	9			
	7				5		8	1
	5		3	4		6		
4		2						
	3	4				1		
9			8				5	
			4			3		7

Gambar 1.1
Contoh Teka-Teki Sudoku
Sumber: https://www.learn-sudoku.com/images/sample_puzzle_half.gif
(Akses: 24 November 2017)

Setiap teka-teki sudoku yang sudah selesai akan membentuk suatu *Latin Square*, yaitu persegi berukuran *n*x*n* yang secara spesifik berisi angka dari 1 sampai *n* yang tersusun sehingga tidak ada baris atau kolom yang mengandung dua buah angka yang sama. [1]

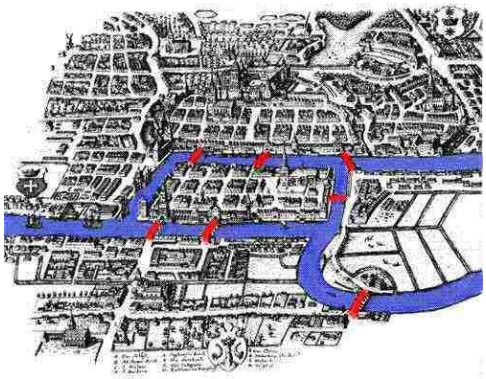
Sudoku (数独) sendiri memiliki arti *single-number*, yang cocok dengan definisi dari *latin square* di atas. Meskipun teka-teki ini mulai popular di Jepang, tetapi sudoku tidak berasal dari Jepang. Sudoku sesungguhnya berasal adalah Swiss, yang lalu mencapai Jepang melalui Amerika Serikat.

Orang yang sering disebut sebagai pencipta sudoku adalah Leonhard Euler yang berasal dari Swiss. Euler adalah pencipta dari istilah *Latin Square*, yang pada dasarnya adalah teka-teki permutasi, yang kemudian menjadi dasar untuk peraturan dari teka-teki sudoku. [2]

II. LANDASAN TEORI

A. Definisi Graf

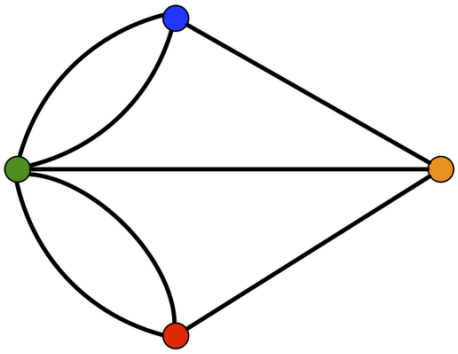
Konsep graf pertama kali ditemukan oleh Leonhard Euler sebagai solusi dari masalah jembatan Königsberg pada tahun 1736. Königsberg dulunya adalah nama sebuah kota di timur negara bagian Prussia, Jerman. Di kota tersebut terdapat sebuah sungai, yaitu sungai Pregal yang mengalir mengitari Pulau Kneiphof lalu terpecah menjadi dua buah anak sungai.



Gambar 2.1
Masalah Jembatan Königsberg
Sumber:
https://www.maa.org/sites/default/files/images/cms_upload/Konigsberg_colour37936.jpg
(Akses: 3 Desember 2017)

Seperti yang dapat dilihat pada gambar 2.1, terdapat tujuh buah jembatan yang menghubungkan daerah-daerah pada kota Königsberg. Masalah jembatan Königsberg adalah: dapatkah seseorang melalui tiap jembatan tepat sekali dan kembali ke tempat ia berada semula? Para penduduk Königsberg sepakat bahwa jawabannya adalah tidak, tetapi mereka tidak dapat menjelaskan penyebabnya.

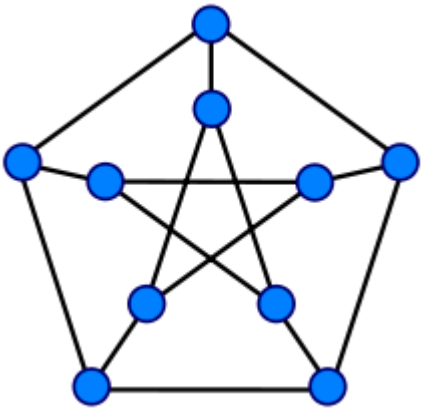
Lalu, pada tahun 1736, Leonhard Euler yang berasal dari Swiss menemukan pembuktian sederhana dari jawaban masalah tersebut, yaitu dengan merepresentasikan daratan sebagai simpul dan jembatan sebagai sisi dari suatu graf.



Gambar 2.2
Representasi Masalah Jembatan Königsberg Sebagai Graf
Sumber: http://rosalind.info/media/Konigsberg_graph.png
(Akses: 3 Desember 2017)

Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak-kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul.

Setiap simpul pada graf dapat diberi identitas, entah dengan huruf, seperti a, b, c, …, atau dengan bilangan asli 1, 2, 3, …, atau dengan gabungan dari keduanya. Sementara, sebuah sisi yang menghubungkan simpul a dengan simpul b dinyatakan dengan pasangan (a, b) atau dinyatakan dengan lambang $e_1, e_2,$ ….



Gambar 2.3 Contoh Graf
Sumber:
https://upload.wikimedia.org/wikipedia/commons/thumb/4/48/Petersen_graph_blue.svg/220px-Petersen_graph_blue.svg.png
(Akses: 3 Desember 2017)

B. Terminologi Graf

1) Bertetangga (*Adjacent*):
   Dua buah simpul pada graf tak-berarah G dikatakan bertetangga bila keduanya terhubung langsung oleh sebuah sisi. Dengan kata lain, u bertetangga dengan v jika (u, v) adalah sebuah sisi pada graf G.
2) Bersisian (*Incident*):
   Untuk sembarang sisi $e = (u, v)$, sisi e dikatakan bersisian dengan simpul u dan simpul v.
3) Simpul Terpencil (*Isolated Vertex*):
   Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya. Atau dapat juga dikatakan bahwa simpul terpencil adalah simpul yang tidak bertetangga dengan satupun simpul lainnya.
4) Graf Kosong (*Null Graph* atau *Empty Graph*):
   Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong.
5) Derajat (*Degree*):
   Derajat suatu simpul pada graf tak-berarah adalah jumlah sisi yang bersisian dengan simpul tersebut.
6) Lintasan (*Path*):
   Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, …, $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G.
7) Siklus (*Cycle*) atau Sirkuit (*Circuit*):
   Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.
8) Terhubung (*Connected*):
   Graf tak-berarah G disebut graf terhubung (*connected* graph) jika untuk setiap simpul u dan v di dalam himpunan V terdapat lintasan dari u ke v. Jika tidak, maka G disebut graf tak-terhubung (*disconnected graph*).
   Graf berarah G dikatakan terhubung jika graf tak-berarahnya terhubung (graf tak-berarah dari G diperoleh dengan menghilangkan arahnya).
9) Upagraf (*Subgraph*) dan Komplemen Upagraf:
   Misalkan $G = (V, E)$ adalah sebuah graf. $G_1 = (V_1, E_1)$ adalah upagraf dari G jika $V_1 \subseteq V$ dan $E_i \subseteq E$.
   Komplemen dari upagraf G_1 terhadap graf G adalah graf $G_2 = (V_2, E_2)$ sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang anggota-anggota E_2 bersisian dengannya.
10) Upagraf Merentang (*Spanning Subgraph*):
    Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan upagraf merentang jika $V_1 = V$ (yaitu G_1 mengandung semua simpul dari G).
11) *Cut-Set*:
    Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G menjadi tidak terhubung. Jadi, *cut-set* selalu menghasilkan dua buah komponen terhubung.
12) Graf Berbobot (*Weighted Graph*):
    Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga atau bobot.

C. Jenis-Jenis Graf

Graf dapat dikelompokkan menjadi berbagai kategori atau jenis tergantung dari sudut pandang pengelompokkannya.

- Berdasarkan ada atau tidaknya sisi ganda atau gelang pada graf:
  1) Graf sederhana (*simple graph*):
     Graf sederhana adalah graf yang tidak mengandung gelang ataupun sisi-ganda.
  2) Graf tak-sederhana (*unsimple-graph*):
     Graf tak-sederhana adalah graf yang mengandung sisi ganda atau gelang. Terdapat dua macam graf tak-sederhana, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung atau memiliki sisi ganda. Sedangkan, graf semu adalah graf yang mengandung atau memiliki gelang (*loop*).

- Berdasarkan orientasi arah pada sisi:
  1) Graf tak-berarah (*undirected graph*):
     Graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah.
  2) Graf berarah (*directed graph* atau *digraph*):
     Graf berarah adalah graf yang setiap sisinya memiliki orientasi arah. Sisi berarah lebih sering disebut sebagai busur (*arc*). Pada graf berarah, (u, v) dan (v, u) menyatakan dua busur yang berbeda. Pada (u, v), simpul u disebut simpul asal (*initial vertex*) dan simpul v disebut simpul terminal (*terminal vertex*). [3]

D. Peraturan Sudoku

Permainan sudoku klasik melibatkan sebuah matriks yang terdiri atas delapan puluh satu buah sel atau kotak. Matriks tersebut kemudian dibagi menjadi sembilan blok yang masing-masing mengandung sembilan buah kotak.

Peraturan dari teka-teki ini cukup sederhana, yaitu setiap dari sembilan blok tersebut harus mengandung angka satu sampai sembilan dalam kotak-kotaknya. Setiap angka hanya dapat muncul sekali. Yang membuat teka-teki ini susah adalah dalam setiap kolom dan baris dalam matriks besar teka-teki ini juga harus mengandung angka satu sampai sembilan tanpa pengulangan ataupun penghilangan. Setiap teka-teki hanya memiliki satu jawaban yang benar. [4]

3	8	2	6	7	1	4	9	5
5	6	7	9	4	8	2	3	1
1	9	4	2	3	5	7	6	8
7	2	8	5	6	4	9	1	3
6	1	3	7	2	9	5	8	4
9	4	5	1	8	3	6	2	7
4	7	1	8	9	6	3	5	2
8	3	6	4	5	2	1	7	9
2	5	9	3	1	7	8	4	6

Gambar 2.4
Contoh Teka-Teki Sudoku yang Sudah Selesai
Sumber: http://www.faszold.com/sudoku/
(Akses: 24 November 2017)

E. Algoritma Penyelesaian Sudoku

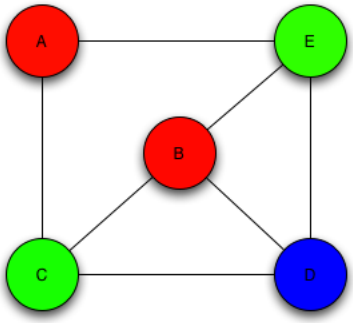
Selain dengan cara manual yaitu menggunakan kertas dan pensil, terdapat berbagai algoritma untuk menyelesaikan teka-teki sudoku secara otomatis, yaitu:

- *Backtracking* atau *Bruteforce Search*
- *Stochastic Search* atau *Optimization Method*
- *Constraint Programming*
- *Exact Cover*

III. Pewarnaan Graf

A. Definisi

Pewarnaan graf adalah pemberian warna kepada elemen tertentu dari graf dengan syarat-syarat yang telah ditentukan sebelumnya. Pewarnaan simpul adalah bentuk pewarnaan graf yang paling sering ditemukan. Dalam hal ini, tidak boleh ada dua simpul bersebelahan yang diwarnai dengan warna yang sama. Masalah pewarnaan graf lain seperti pewarnaan sisi dan pewarnaan muka dapat diubah menjadi pewarnaan simpul.



Gambar 3.1 Contoh Pewarnaan Simpul
Sumber: http://www.geeksforgeeks.org/wp-content/uploads/vertex_coloring.png
(Akses: 3 Desember 2017)

Jumlah warna terkecil yang dapat digunakan untuk mewarnai graf G disebut bilangan kromatik atau *chromatic number* dari graf G. Contohnya, pada gambar 3.1, bilangan kromatik dari graf tersebut adalah tiga.

B. Aplikasi

Terdapat banyak aplikasi dari pewarnaan graf, terutama pewarnaan simpul, yang meliputi:

1) Tabel Waktu atau Penjadwalan
   Pewarnaan graf dapat digunakan untuk menentukan jadwal ujian supaya tidak ada dua ujian mata kuliah dengan mahasiswa yang sama yang dijadwalkan pada waktu yang sama, dan untuk mencari tahu berapa slot waktu yang dibutuhkan untuk melaksanakan ujian. Masalah ini dapat dipecahkan dengan merepresentasikan suatu mata kuliah sebagai simpul dan sisi di antara dua simpul berarti ada mahasiswa yang sama yang mengambil kedua mata kuliah tersebut. Jadi, slot waktu minimum dapat diketahui dengan mencari bilangan kromatik dari graf tersebut.

2) Penentuan Frekuensi Radio

Frekuensi pada tiap menara radio di lokasi yang sama haruslah berbeda. Pada masalah ini, tiap menara dapat direpresentasikan sebagai simpul dan adanya sisi di antara dua simpul berarti dua menara tersebut berada pada lokasi yang sama.

3) Sudoku

Sudoku juga merupakan suatu variasi dari masalah pewarnaan graf, dalam hal ini setiap sel matriks merepresentasikan sebuah simpul dan terdapat sisi di antara dua simpul jika dua sel tersebut terletak pada baris, kolom, atau blok yang sama.

4) Alokasi *Register*

Dalam optimisasi *compiler*, alokasi *register* adalah proses penentuan variabel dalam program dalam jumlah yang banyak ke *register* dalam CPU yang hanya ada sedikit. Masalah ini juga merupakan masalah pewarnaan graf.

5) Graf Bipartit

Suatu graf dapat dikatakan bipartit jika graf tersebut dapat diwarnai dengan dua warna saja atau dengan kata lain memiliki bilangan kromatik dua. Jika bilangan kromatik dari graf tersebut tidak sama dengan dua, maka graf tersebut bukanlah graf bipartit.

6) Pewarnaan Peta

Pewarnaan graf dapat menyelesaikan masalah di mana peta geografis dari suatu negara, negara bagian, atau provinsi yang setiap kota bersebelahannya tidak boleh diwarna dengan warna yang sama. Pada masalah ini, hanya dibutuhkan maksimal empat warna menurut Teorema Empat Warna. [5]

IV. ALGORITMA RUNUT-BALIK (*BACKTRACKING)*

Algoritma runut-balik atau *backtracking* adalah salah satu algoritma yang menggunakan prinsip rekursifitas dalam pengimplementasiannya. Algoritma *backtracking* mencoba untuk solusi untuk suatu permasalahan komputasional secara bertahap. Ketika perlu memutuskan antara beberapa alternatif komponen selanjutnya dari solusi, algoritma ini akan mencoba segala solusi yang memungkinkan secara rekursif. [6]

Percobaan untuk menemukan solusi dimulai dengan mencoba salah satu dari banyak kemungkinan. Jika ditemukan solusi dengan kemungkinan-kemungkinan ada, solusi tersebut akan ditampilkan. Jika tidak ada kemungkinan yang berhasil menyelesaikan masalah, maka masalah tersebut dapat disebut tidak memiliki solusi atau penyelesaian. [7]

Berikut adalah algoritma umum dari algoritma *backtracking:*

```
Pick a starting point.
while(Problem is not solved)
    For each path from the
starting point.
        check if selected path
is safe, if yes select it
                and make
recursive call to rest of the
problem
        If recursive calls
returns true, then return true.
        else undo the current
move and return false.
    End For
    If none of the move works
out, return false, NO SOLUTION.
```

V. APLIKASI ALGORITMA *BACKTRACKING* DALAM PEWARNAAN GRAF

Algoritma runut-balik atau *backtracking* dapat digunakan untuk menyelesaikan masalah pewarnaan graf dengan merepresentasikan graf sebagai matriks ketetanggaan atau *incidency matrix* GRAPH[1..n][1..n]. Direkomendasikan menggunakan *boolean* sebagai elemen dari matriks karena algoritma mementingkan apakah kedua simpul bertetangga atau tidak. Dalam hal ini, GRAPH(i,j) bernilai *true* jika simpul *i* dan *j* bertetangga dan bernilai *false* jika sebaliknya.

Jenis warna atau bilangan kromatik dapat direpresentasikan dengan tipe bilangan 1, 2, …, *m* dan solusi dapat direpresentasikan sebagai *n-tuple* {X(1), X(2), …, X(n)} di mana X(i) bernilai bilangan yang merepresentasikan warna dari simpul *i.*

Berikut adalah algoritma pewarnaan graf menggunakan algoritma runut-balik atau *backtracking:*

```
procedure MCOLORING(k) //k adalah indeks
selanjutnya dari simpul yang akan diberi warna

global integer m,n,X(1..n)
boolean GRAPH[1..n][1..n]
integer k

loop //bangkitkan semua kemungkinanuntuk X(k)
     call NEXTVALUE(k) //set X(k) dengan
                       //warna yang
                       //memungkinkan

     if (X(k)=0) then exit //warna sudah habis
     endif

     if (k=n)
        then print(X)
        else call MCOLORING(k+1)
     endif
repeat
end MCOLORING
```

```
procedure NEXTVALUE(k)

global integer m,n,X(1..n)
boolean GRAPH[1..n][1..n]
integer j,k
loop
     X(k)-(X(k)+1) mod (m+1) //warna
berikutnya
     if (X(k)=0) then
     return endif //warna sudah habis
     for (j=1 to n ) do
        if (GRAPH(k,j) and (X(k)=X(j)))
         //jika (k,j) adalah sebuah sisi
         dengan   //simpul awal dan akhir
         berwarna sama
        then exit
        endif
        if j=n+1 then
        return endif
repeat
end NEXTVALUE
```

Algoritma ini dapat diimplementasikan untuk sembarang graf dengan *n* simpul dan *m* umlah warna. Algoritma ini juga dapat digunakan untuk menguji kebenaran dari Teorema Empat Warna, yaitu apakah benar suatu graf planar dapat diwarnai dengan maksimal empat warna saja. Algoritma ini cukup efektif untuk jumlah simpul yang tidak terlalu besar dan selalu dapat mencapai solusi yang unik. [8]

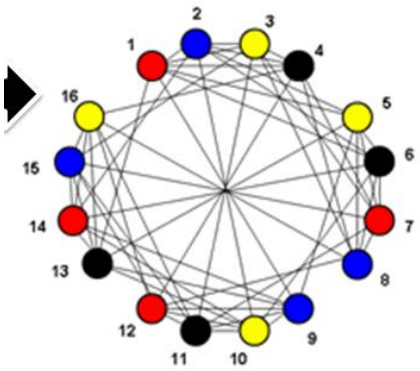
VI. Penyelesaian Sudoku Dengan Pewarnaan Graf

Untuk menyelesaikan sudoku dengan pewarnaan graf, pertama-tama yang harus dilakukan adalah mengubah matriks teka-teki menjadi sebuah graf. Untuk matriks 9x9, graf yang telah diubah dari matriks akan memiliki 81 simpul sebagai representasi dari tiap sel matriks. Dua simpul yang berbeda akan bertetangga jika terletak dalam baris, kolom, atau blok yang sama.

Setiap simpul pada graf yang telah dibuat akan memiliki derajat yang sama, yaitu 20, maka graf ini dapat dikatakan merupakan graf teratur. Setiap matriks sudoku yang terselesaikan berkoresponden dengan pewarnaan-*k* dari graf. [9]

1	2	3	4
3	4	1	2
2	3	4	1
4	1	2	3

Gambar 6.1
Contoh Matriks Sudoku 4x4 yang Sudah Selesai
Sumber:
https://www.codeproject.com/KB/recipes/801268/sudoku.PNG
(Akses: 3 Desember 2017)



Gambar 6.2
Graf Konversi dari Gambar 6.1
Sumber:
https://www.codeproject.com/KB/recipes/801268/sudoku.PNG
(Akses: 3 Desember 2017)

Setelah matriks diubah menjadi graf representasi, dapat dilakukan pewarnaan graf untuk menemukan solusi dari teka-teki sudoku. Algoritma pewarnaan graf untuk menyelesaikan sudoku secara manual dapat dibagi menjadi langkah-langkah berikut:

1) Simpul yang telah diwarnai dipilih dan segala sisi yang bersisian dengan simpul tersebut ditandai dengan warna yang sama sehingga simpul-simpul yang bertetangga dengan simpul tersebut tidak lagi dapat diwarnai dengan warna yang sama. Langkah ini diulangi untuk setiap simpul dengan angka yang telah diketahui sebelumnya.
2) Ditemukan simpul yang terhubung dengan sisi berwarna dalam jumlah terbanyak (kemungkinan hanya ada satu kandidat).
3) Dari simpul-simpul yang telah ditemukan, jika ada satu simpul yang dapat diwarnai dengan hanya satu warna, maka simpul tersebut diberi warna dan prosedur dilanjutkan kembali dari langkah pertama. Jika tidak ada simpul yang memenuhi kondisi di atas, prosedurkan dilanjutkan dengan langkah keempat.
4) Dari simpul-simpul yang telah ditemukan, simpul yang bertetangga dengan simpul tidak berwarna yang paling banyak dipilih dan diwarnai dengan warna yang paling sedikit digunakan hingga saat itu. Jika ada lebih dari satu simpul yang memenuhi, dipilih satu secara acak. Prosedur lalu dilanjutkan dengan langkah pertama. [9]

VII. Kesimpulan

Teori graf, terutama dalam hal ini pewarnaan graf, memiliki banyak sekali kegunaan yang dapat diaplikasikan dalam kehidupan manusia sehari-harinya. Salah satu permasalahan yang dapat diselesaikan dengan menggunakan pewarnaan graf adalah teka-teki sudoku yang populer di segala kalangan karena menarik dan menantang

Untuk mempermudah penyelesaian teka-teki sudoku, dapat digunakan algoritma runut-balik yang menggunakan prinsip rekursifitas supaya dapat menemukan solusi dari masalah pewarnaan graf dengan cepat dan tepat, dan dari situ dapat ditemukan solusi dari teka-teki sudoku dengan berbagai macam

tingkat kesusahan. Algoritma pun tidak akan terlalu lama dalam menemukan solusi karena jumlah input simpul yang tidak terlalu besar.

VIII. Ucapan Terima Kasih

Penulis mengucapkan terimakasih kepada Tuhan YME atas bimbingannya selama pengerjaan makalah, orang tua dan teman-teman dari penulis atas segala dukungan yang telah diterima, dan kepada Drs. Judhi Santoso, M.Sc, Dr. Ir. Rinaldi Munir, M.T., dan Dra. Harlili S., M.Sc selaku dosen mata kuliah IF2120 Matematika Diskrit atas segala pengajaran dan ilmu yang telah diterima penulis selama kegiatan belajar mengajar.

Referensi

[1] http://mathworld.wolfram.com/LatinSquare.html diakses pada 2 Desember 2017 pukul 16.30.
[2] http://www.sudokudragon.com/sudokuhistory.htm diakses pada 2 Desember 2017 pukul 17.00.
[3] Munir, Rinaldi. *Matematika Diskrit*. Bandung: Informatika. 2010. ch. 6
[4] http://www.counton.org/sudoku/rules-of-sudoku.php diakses pada 3 Desember 2017 pukul 19.00.
[5] http://www.geeksforgeeks.org/graph-coloring-applications/ diakses pada 3 Desember 2017 pukul 19.00.
[6] http://jeffe.cs.illinois.edu/teaching/algorithms/notes/03-backtracking.pdf diakses pada 3 Desember 2017 pukul 21.20
[7] http://algorithms.tutorialhorizon.com/introduction-to-backtracking-programming/ diakses pada 3 Desember 2017 pukul 21.20
[8] Sari, Deasy Ramadian. Widyasari, Wulan. Ria, Eunice Sherta. *Penerapan Algoritma Backtracking pada Pewarnaan Graf.* Published online, 2005.
[9] http://www.cs.kent.edu/~dragan/ST-Spring2016/SudokuGC.pdf diakses pada 3 Desember 2017 pukul 22.00.

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, 3 Desember 2017

Michelle Eliza Gananjaya
13516015