

Aplikasi Pewarnaan Graf Untuk Menyelesaikan Teka-Teki Sudoku

Dandy Arif Rahman 13516086
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
dandyarifrahman@gmail.com

Ilmu tentang teori graf sangat luas, dari mulai teori tentang titik, sisi, upagrah, dan lain sebagainya. Terkadang saat mempelajarinya tidak puas jika hanya memahami teorinya saja, padahal banyak aplikasi dari teori graf ini. Salah satu aplikasi dari teori graf adalah pewarnaan graf. Pewarnaan graf banyak digunakan di bidang pemetaan, metode ini berguna memetakan negara-negara di dunia dengan cara membedakan warna dari wilayah dua negara yang bersisian, sehingga mudah untuk dikenali batas-batas antar negara.

Kata Kunci—Angka kromatik, pewarnaan graf, K-colourable, sudoku.

I. PENGANTAR

Teori graf merupakan pembahasan yang muncul pertama kali pada tahun 1736, yakni ketika Leonhard Euler mencoba untuk mencari solusi dari permasalahan klasik yang sangat terkenal yaitu Jembatan Konigsberg. Konigsberg merupakan kota di bagian utara Jerman, sekarang bernama Kalinigrad, disana terdapat sungai Pregal yang mengalir mengitari pulau Kneiphof lalu bercabang menjadi dua buah anak sungai.

Konigsberg menjadi saksi dari sejarah teori graf yang berguna sampai saat ini. Sungai pregel yang melalui Konigsberg membagi wilayah daratan pada kota tersebut menjadi empat bagian. Tujuh buah jembatan dibangun di atas sungai tersebut pada bagian yang memungkinkan untuk bepergian antar keempat wilayah tersebut. Pada abad ke-17, warga Konigsberg gemar berjalan di tepi sungai, hingga akhirnya beberapa dari mereka memikirkan apakah mungkin untuk berjalan di Konigsberg dan melalui setiap jembatan hanya sekali. Hal inilah yang kemudian disebut Teka-Teki Jembatan Konigsberg yang tidak dapat terselesaikan untuk waktu yang cukup lama dan menjadi terkenal di seluruh negeri.

Teka teki tersebut menarik perhatian Euler. Ia kemudian meneliti bahwa kasus tersebut dapat direpresentasikan ke dalam diagram. Setelah sekian banyak kegagalan warga Konigsberg untuk menemukan cara melalui seluruh jembatan hanya sekali, hingga akhirnya pada tahun 1736 masalah tersebut dijadikan sebuah kasus matematika dan kemustahilan, untuk menyelesaikan teka-teki tersebut terbukti. Pada tahun

tersebut, seorang pakar matematika ternama, Leonard Euler, menulis sebuah artikel yang membahas tidak hanya solusi atas teka-teki Konigsberg semata, akan tetapi juga dilengkapi dengan metode umum untuk persoalan serupa lainnya.

Seiring berganti tahun, ilmu pengetahuan berkembang begitu juga dengan ilmu tentang graf. Banyak aplikasi tentang graf bermunculan, salah satunya adalah pewarnaan graf, teknik ini dilakukan untuk merepresentasikan sebuah peta yang terdiri atas wilayah-wilayah yang berbeda. Wilayah ini dibedakan dengan warna tertentu, warna dari suatu wilayah tidak boleh sama dengan wilayah lain yang berdampingan. Teknik ini kemudian yang diadopsi untuk pemetaan peta dunia, pengaturan jadwal, pengaturan *table planner*, penyelesaian teka-teki sudoku, dan lainnya.

II. DEFINISI

- 1) **Graf** : Representasi dari himpunan objek yang berpasang-pasangan. Objek tersebut direpresentasikan sebagai titik dan hubungan antar objek direpresentasikan sebagai sisi_[7].
- 2) **Pewarnaan Graf** : Pewarnaan pada titik-titik yang ada di graf sehingga tidak ada dua titik yang berdampingan memiliki warna yang sama_[7].
- 3) **K-colouring** : Pewarnaan titik pada graf dengan hanya memakai K jumlah warna_[1].
- 4) **K-colourable** : Graf yang mempunyai K-colouring_[1].
- 5) **Angka Kromatik** : Angka kromatik dari suatu graf G adalah jumlah warna minimum yang diperlukan untuk melakukan pewarnaan graf. Angka kromatik di dinotasikan dengan $\chi(G)$ _[6].
- 6) **Independent Set** : himpunan titik-titik yang tidak berdampingan satu sama lain_[1].

7) **Colour classes** : himpunan titik-titik yang mempunyai warna yang sama_[1].

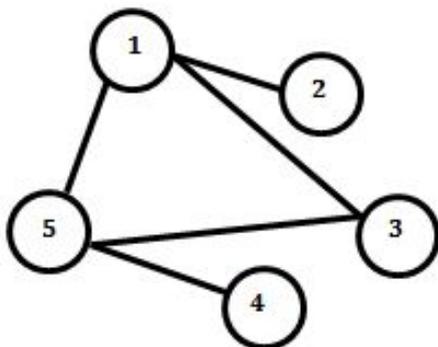
III. TEORI

- 1) Jika H adalah sebuah upagraf dari G maka $\chi(H) \leq \chi(G)$ _[7].
- 2) Jika G mempunyai n titik maka $\chi(G) \leq n$ _[7].
- 3) $\chi(G) = 1$ jika dan hanya jika G tidak mempunyai sisi_[7].
- 4) Graf *cycle* G yang mempunyai titik berjumlah genap mempunyai $\chi(G) = 2$ _[7].
- 5) Graf *cycle* G yang mempunyai titik berjumlah ganjil mempunyai $\chi(G) = 3$ _[7].
- 6) Graf komplit G berorde n mempunyai $\chi(G) = n$ _[7].

IV. MEMAHAMI LEBIH DALAM TENTANG PEWARNAAN GRAF

Berbicara mengenai teori tentang graf, teori graf telah melahirkan bidang-bidang baru yang mempermudah banyak persoalan. Graf membantu merepresentasikan suatu permasalahan kedalam bentuk matematika yang membuatnya lebih mudah untuk dikomputasikan.

Salah satu bidang dari graf yang menarik banyak perhatian adalah *graf coloring* atau pewarnaan graf. Seperti yang sempat disinggung diatas definisi dari pewarnaan graf adalah pewarnaan pada objek-objek yang di representasikan sebagai titik-titik yang ada di graf sehingga tidak ada dua titik yang berdampingan memiliki warna yang sama. Sebagai contoh, gambar 4.1 merepresentasikan graf yang mempunyai 5 titik dan 5 sisi.

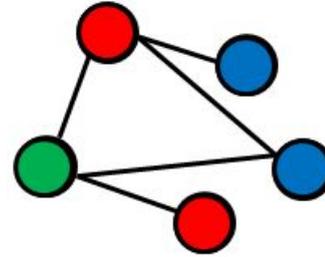


Gambar 4.1. Graf yang mempunyai 5 titik dan 5 sisi

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27November 2017 Pk. 15.40)

Jika graf pada gambar 4.1 kita beri warna maka grafnya akan menjadi seperti pada gambar 4.2



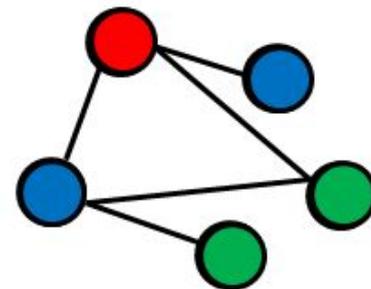
Gambar 4.2. Pewarnaan Graf pada gambar 4.1

Sumber :

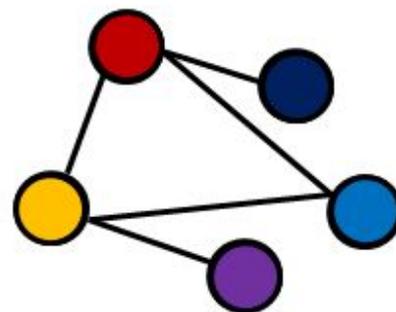
<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

Pada graf diatas, titik 1 dan 2 tidak boleh memiliki warna yang sama karena kedua titik tersebut terhubung oleh suatu sisi, sedangkan titik 2 dan 3 boleh memiliki warna yang sama karena kedua titik tersebut tidak terhubung oleh suatu sisi. Begitu juga dengan titik-titik lainnya, dua titik yang terhubung oleh suatu sisi tidak boleh memiliki warna yang sama.

Suatu graf memungkinkan untuk memiliki lebih dari satu kombinasi pewarnaan graf. Dibawah ini merupakan kombinasi pewarnaan graf lainnya.



(1)



(2)

Gambar 4.3. (1) dan (2) Kombinasi pewarnaan graf pada Gambar 4.1 lainnya

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

Sekarang sudah jelas bahwa suatu graf memungkinkan untuk memiliki lebih dari satu kombinasi pewarnaan, Sekarang

tujuan kita adalah untuk mencari warna minimum yang dibutuhkan untuk pewarnaan suatu graf.

Pada gambar 4.3 (1) kita dapat bahwa pewarnaan graf tersebut menggunakan tiga macam warna, sedangkan pada gambar 4.3 (2) terdapat 5 macam warna. Kalau kita lihat dari banyaknya warna, tentu gambar 4.3 (1) lebih hemat warna dibanding dengan gambar 4.3 (2). Sebenarnya mengapa kita membutuhkan sesedikit mungkin warna? ini disebabkan karena dalam banyak kasus warna merepresentasikan jumlah cara. Karena alasan itu agar tercapai efisiensi maka cara yang digunakan harus minimum. Warna minimum yang dibutuhkan untuk melakukan pewarnaan pada suatu graf disebut dengan *chromatic number*.

V. ALGORITMA *CONTRACTION* DAN *GREEDY COLORING*

Masalah pewarnaan graf sampai saat ini tergolong ke *computationally hard*, banyak riset yang membahas tentang ini, tetapi belum ada algoritma pewarnaan yang sangat efisien. Meskipun demikian algoritma yang ada sekarang cukup memuaskan, secara umum algoritma yang membahas tentang pewarnaan terbagi menjadi dua kategori, yaitu *contraction* dan *greedy*.

1) Algoritma Greedy

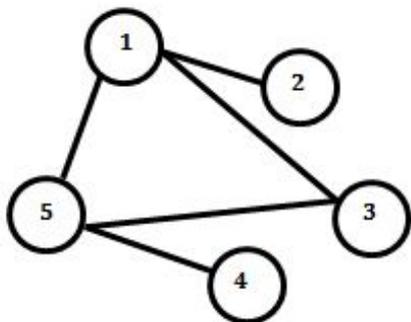
Algoritma ini fokus terhadap pemilihan titik selanjutnya yang akan diwarnai. Dalam algoritma ini, jika sebuah titik telah diwarnai, titik tersebut tidak bisa berubah warna. Salah satu algoritma *greedy* yang terkenal adalah *Welsh-Powell Algorithm*.

Cara kerja dari algoritma *Welsh-Powell* adalah sebagai berikut:

1. Cari derajat setiap titik
2. Urutkan titik-titik secara terurut mengecil berdasarkan jumlah derajatnya.
3. Warnai titik yang mempunyai derajat paling tinggi.
4. Warnai titik-titik lainnya yang tidak terhubung dengan titik yang telah diwarnai diatas dengan warna yang sama.
5. Hilangkan titik-titik yang sudah diwarnai dari list tersebut
6. Ulangi langkah 3 sampai semua titik berhasil diwarnai.

Berikut ilustrasinya :

Misalkan kita mempunyai graf seperti gambar 5.1



Gambar 5.1 (a) Graf yang mempunyai 5 titik dan 5

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

Langkah 1 :

Tabel 5.1. Tabel derajat tiap titik dari graf di gambar 5.1

Vertex	Degree
1	3
2	1
3	2
4	1
5	3

Langkah 2 :

Urutan titiknya adalah 1, 5, 3, 2, 4

Langkah 3 :

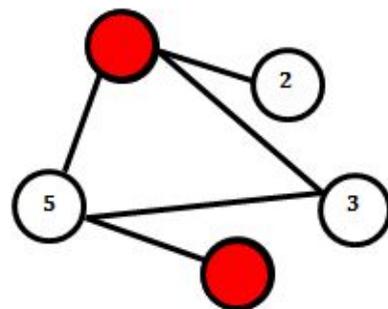
Warnai titik 1 dengan warna merah

Langkah 4 :

Warnai titik 4 dengan warna merah, karena 4 tidak berhubungan dengan titik 1.

Langkah 5 :

List yang baru menjadi 5,3,2



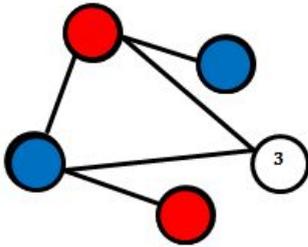
Gambar 5.1 (b) Graf yang mempunyai 5 titik dan 5 sisi setelah di beri warna di kedua titiknya

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

Langkah 6 :

Ulangi warnai titik yang berderajat paling tinggi dengan warna yang berbeda, yaitu titik 5 dengan warna biru. Warnai titik 2 juga dengan warna yang sama karena titik dua tidak terhubung langsung dengan titik 5.

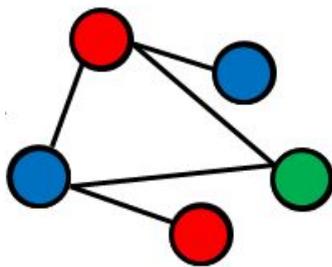


Gambar 5.1 (c) Graf yang mempunyai 5 titik dan 5 sisi setelah diberi warna di keempat titiknya

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

Kemudian warnai titik yang tersisa dengan warna yang lain, sehingga terbentuk seperti gambar 5.1 (d)



Gambar 5.1 (d) Graf berwarna yang mempunyai 5 titik dan 5 sisi

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

2) Algoritma Contraction

Algoritma yang kedua adalah algoritma *contraction*. Prinsip kerja dari algoritma ini adalah dengan mencari dua buah titik yang terhubung, kemudian hapus sisi antara dua titik tersebut. Lalu buat satu buah titik baru yang menerima semua titik yang terhubung dengan kedua titik sebelumnya.

Cara kerja dari algoritma *Contraction* adalah sebagai berikut:

1. Pilih titik V yang berderajat paling tinggi.
2. Cari himpunan titik yang tidak bersisian dengan titik V.
3. Dari himpunan titik diatas, cari titik Y yang mempunyai keterhubungan titik paling mirip dengan titik V.
4. Gabungkan titik Y dengan V menjadi titik T,V.
5. Hapus titik Y dari himpunan titik, kemudian lakukan langkah 3 sampai 5 hingga himpunan titik tersebut kosong.
6. Hapus titik V dari graf.
7. Ulangi langkah 1-6 hingga graf kosong dan terbentuk himpunan dari grup-grup titik.

Berikut ilustrasinya :

Kita gunakan contoh dari graf yang berada pada gambar 5.1(a).

Langkah 1 :

Dari graf tersebut titik yang berderajat maksimum adalah titik 1 dan titik yang tidak bersisian dengannya adalah titik 4. Gabungkan titik tersebut

Langkah 2 :

Titik yang tidak bersisian dengan titik 1 adalah titik 4.

Langkah 3 :

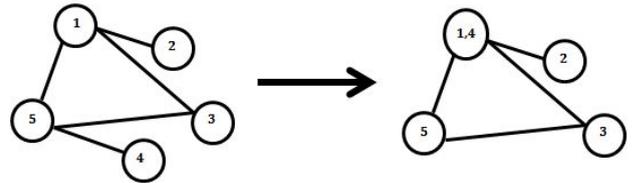
Karena hanya terdapat 1 titik maka lanjutkan ke langkah berikutnya.

Langkah 4 :

Gabungkan titik 4 ke titik 1.

Langkah 5 :

Hapus titik 4 dari himpunan titik, sehingga himpunan kosong.

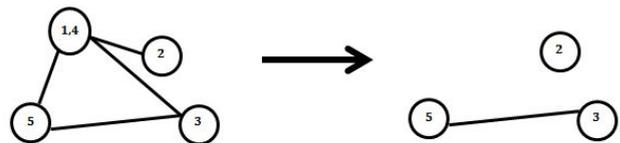


Gambar 5.2 (a) Langkah pewarnaan graf menggunakan *contraction algorithm*
Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk. 15.40)

Langkah 6 :

Hapus gabungan titik 1 dan 4 dari graf.



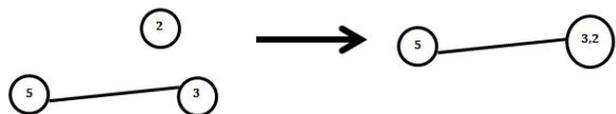
Gambar 5.2 (b) Langkah pewarnaan graf menggunakan *contraction algorithm*

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk.15.55)

Langkah 7 :

Ulangi langkah 1-6, hingga graf hasil #####. Sekarang titik yang mempunyai derajat paling tinggi adalah titik 5 dan titik 3, kita pilih titik 3. Titik yang tidak bersisian dengan titik 3 adalah titik 2.



Gambar 5.2 (c) Langkah pewarnaan graf menggunakan *contraction algorithm*

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk.15.55)

Kemudian gabungkan titik 2 ke titik 3

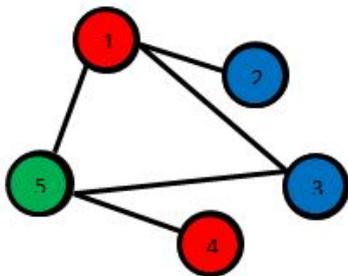


Gambar 5.2 (d) Langkah pewarnaan graf menggunakan *contraction algorithm*

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk.15.55)

Hapus titik 2 yang lama dari himpunan titik, sehingga menyebabkan himpunan list kosong. Karena himpunan list kosong maka hapus titik 3,2. Sekarang hanya tersisa 1 titik yaitu titik 5, itu berarti telah selesai. Dari algoritma *contraction* diperoleh 3 grup titik yaitu {1,4}, {2,3}, dan {5} tiap titik kita beri warna berbeda.



Gambar 5.2 (c) Graf hasil berwarna menggunakan *contraction algorithm*

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 Diakses November 2017 Pk. 15.40)

VI. TEKA-TEKI SUDOKU

Kata sudoku atau 数独 berasal dari Bahasa Jepang, karakter Su berarti angka dan Doku berarti tunggal, meskipun demikian teka-teki ini bukan permainan yang berasal dari Jepang melainkan dari Swiss yang kemudian dibawa ke Jepang.

Dibalik permainan yang menarik perhatian banyak orang ini terdapat seorang matematikawan jenius yaitu Leonhard Euler dalam pembuat teka-teki tersebut. Euler merupakan kelahiran Basel, Swiss.

Teka-teki sudoku merupakan permainan yang dimainkan oleh 1 orang. Teka-teki itu sendiri berbentuk tabel persegi yang berisi tabel-tabel persegi kecil atau biasa disebut blok. Biasanya tabel tersebut berukuran 9x9 dan sudah terisi sebagian selnya dengan angka. Pemain harus menebak angka-angka untuk sel yang masih kosong, dengan ketentuan angka di sel tersebut tidak boleh terdapat di baris, kolom, dan blok tempat sel tersebut.

		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 6.1 Teka-teki sudoku (Origin of Sudoku)

Sumber :

<http://www.sudoku-dragon.com/sudokuhistory.htm> (Diakses 28 November 2017 Pk. 06.30)

Jika ditinjau dari sisi teori graf, sebuah teka-teki sudoku berukuran 9x9 bisa dilihat sebagai 81 titik yang tiap titiknya merepresentasikan tiap sel. Kemudian dua titik disebut bersisian jika titik tersebut tidak bisa diisi dengan angka yang sama. Sebagai contoh, semua sel di baris, kolom, atau blok yang sama akan mempunyai sisi antara titik-titik tersebut. Jadi untuk tabel berukuran NxN terdapat N buah warna. Warna tersebut merepresentasikan angka yang akan diisi ke sel. Warna berbeda merepresentasikan angka yang berbeda pula.

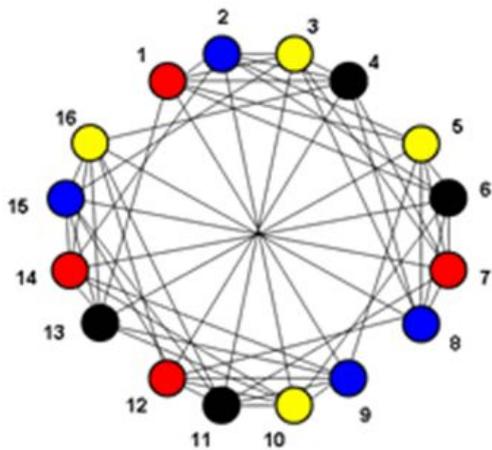
Sebagai contoh, kita akan gunakan sudoku berukuran 4x4 untuk memudahkan perrepresentasian grafnya.

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

Gambar 6.2(a) Teka-teki sudoku berukuran 4x4

Sumber :

<https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (Diakses 27 November 2017 Pk.15.55)



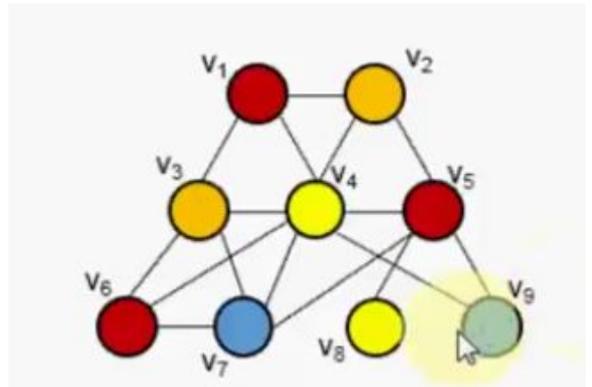
Gambar 6.2(b) Representasi teka-teki sudoku berukuran 4x4 dalam bentuk graf berwarna
 Sumber :
<https://www.youtube.com/watch?v=v4RAYOjKb5Y> (Diakses 26 November 2017 Pk. 14.00)

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

Gambar 6.2(c) Teka-teki sudoku berukuran 4x4 berwarna
 Sumber :
<https://www.youtube.com/watch?v=v4RAYOjKb5Y> (Diakses 26 November 2017 Pk. 14.00)

<https://www.youtube.com/watch?v=v4RAYOjKb5Y> (26 November 2017 Pk. 14.00)

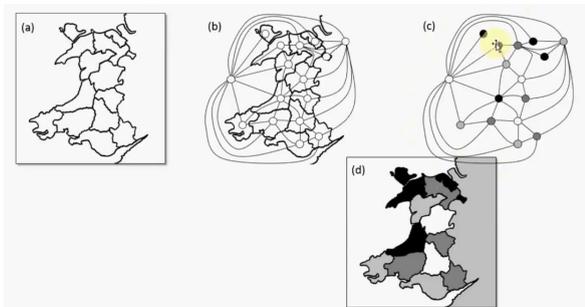
Lecture Name	Clashes with...
1) Algebra	2, 3, 4
2) Probability Theory	1, 4, 5
3) Graph Theory	1, 4, 6, 7
4) French Literature	1, 2, 3, 5, 6, 7, 9
5) English Literature	2, 4, 7, 8, 9
6) Welsh Literature	3, 4, 7
7) European Economics	3, 4, 5, 6
8) Algorithms	5
9) C++ Programming	4, 5



Gambar 4.2. Representasi tabel 4.1 dalam bentuk graf berwarna
 Sumber :
<https://www.youtube.com/watch?v=v4RAYOjKb5Y> (Diakses 26 November 2017 Pk. 14.00)

V. APLIKASI PEWARNAAN GRAF LAINNYA

1) Pembuatan peta dunia.



Gambar 4.1. Representasi peta dunia dalam bentuk graph berwarna
 Sumber :
<https://www.youtube.com/watch?v=v4RAYOjKb5Y> (Diakses 26 November 2017 Pk. 14.00)

2) Merancang jadwal yang waktunya berisikan

Tabel 4.1. Jadwal kuliah
 Sumber :

Tabel 4.2. Jadwal waktu kuliah
 Sumber :
<https://www.youtube.com/watch?v=v4RAYOjKb5Y> (Diakses 26 November 2017 Pk. 14.00)

Timeslot 1	2	3	4
Lecture 1 Algebra	Lecture 2 Probability Theory	Lecture 7 European Economics	Lecture 4 French Literature
Lecture 5 English Literature	Lecture 3 Graph Theory	Lecture 9 C++ Programming	Lecture 8 Algorithms
Lecture 6 Welsh Literature			

REFERENCES

- [1] Guichard, D., Graph Theory, *Graph Coloring* https://www.whitman.edu/mathematics/cgt_online/book/section05.08.html (November 26, 2017)
- [2] Lewis, R., 2015, Applications of Graph Colouring <https://www.youtube.com/watch?v=v4RAYOjKb5Y> (November 26, 2017)
- [3] Herke, S., 2015, Graph Theory, *Vertex Coloring* https://www.youtube.com/watch?v=4FE79v_JkCE (November 26, 2017)
- [4] Eissa, M., 2014, A Sudoku Solver using Graph Coloring, <https://www.codeproject.com/Articles/801268/A-Sudoku-Solver-using-Graph-Coloring> (November 27, 2017)
- [5] Origin of Sudoku, <http://www.sudoku-dragon.com/sudokuhistory.htm> (November 28, 2017)
- [6] Munir, R., 2006, Matematika Diskrit, *Graf*, (November 26, 2017)
- [7] Rosen, Kenneth H, 2012, Discrete Mathematics and Its Applications, New York:McGraw-Hill, (November 26, 2017)

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

A handwritten signature in black ink, appearing to read 'Dandy Arif Rahman', with a stylized, cursive script.

Dandy Arif Rahman
13516086