

# Aplikasi Graf pada Penentuan Jadwal dan Jalur Penerbangan

Hishshah Ghassani - 13514056  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
hishshah@students.itb.ac.id

**Abstrak**— Pesawat terbang merupakan salah satu transportasi yang sering digunakan oleh masyarakat Indonesia, terutama mereka yang sering berpergian ke luar kota dengan jarak yang sulit ditempuh dengan transportasi darat. Untuk meminimumkan kecelakaan di udara, maka harus penentuan jadwal dan jalur penerbangan harus diatur sedemikian sehingga tidak ada pesawat yang bertabrakan. Makalah ini menawarkan salah satu ide cara untuk menentukan jadwal dan jalur penerbangan, dengan menggunakan teori graf. Dengan metode pewarnaan graf, dapat ditentukan jalur dan jadwal yang bisa saja mengurangi jumlah kecelakaan. Selain pewarnaan graf, dibutuhkan juga algoritma-algoritma pendukung seperti algoritma Greedy dan algoritma Dijkstra untuk menentukan jalur terpendek dan efisien.

**Kata Kunci**— graf, jadwal, jalur, pewarnaan graf, algoritma Greedy, algoritma Dijkstra.

## I. PENDAHULUAN

Banyaknya maskapai penerbangan yang beroperasi setiap waktunya menyebabkan pengaturan jadwal dan jalur penerbangan menjadi tidak mudah. Menurut Wikipedia, jumlah maskapai penerbangan di Asia saja sudah mencapai lebih dari 100 maskapai. Banyaknya jumlah maskapai tersebut menyebabkan banyak pesawat yang beroperasi pada satu waktu, sehingga kemungkinan terjadinya kecelakaan di udara cukup besar apabila penentuan jadwal dan jalur penerbangan tidak diatur dengan baik. Oleh karena itu, dibutuhkan suatu metode agar jadwal dan jalur penerbangan dapat diatur sedemikian sehingga tidak terjadi kecelakaan di udara.



Gambar 1. Lalu lintas udara di daerah Asia Tenggara

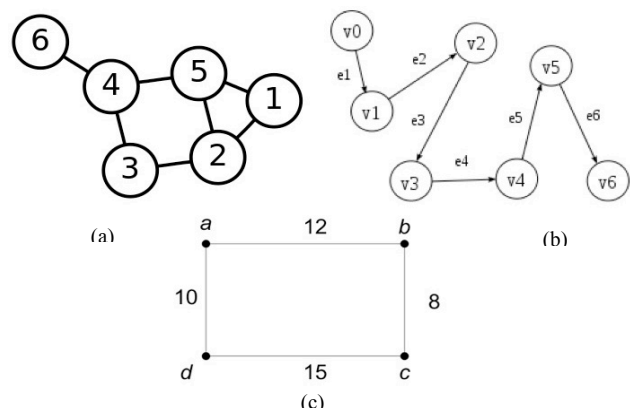
Jadwal dan jalur penerbangan harus disusun dengan baik oleh maskapai penerbangan, selain untuk meminimalkan kecelakaan, juga karena mempengaruhi efisiensi dan keuntungan bagi maskapai tersebut. Salah satu metode yang dapat digunakan dalam menentukan jadwal dan jalur penerbangan adalah dengan teori graf. Graf adalah kumpulan simpul-simpul (*vertex*) yang dihubungkan dengan sisi (*edge*). Satu sisi dapat menghubungkan satu simpul yang sama. Sisi yang seperti itu disebut gelang (*loop*). Dua sisi atau lebih juga dapat menghubungkan dua simpul yang sama. Sisi-sisi tersebut disebut sisi ganda.

Penerapan graf dalam kehidupan sangat banyak. Selain penentuan jadwal dan jalur transportasi, graf juga dapat digunakan dalam bidang jaringan komunikasi, rangkaian listrik, turnamen *round-robin*, permodelan *vending machine*, dan lain-lain.

## II. TEORI DASAR

### A. Teori Graf

Graf biasanya digunakan untuk merepresentasikan objek diskrit dan menghubungkan objek-objek tersebut. Graf terdiri dari simpul dan sisi. Suatu graf dapat  $G = (V, E)$ . Graf dinyatakan sebagai graf kosong apabila tidak ada sisi di antara simpul-simpul yang ada. Graf mungkin memiliki arah dan mungkin juga berbobot. Graf berarah adalah graf yang tiap sisinya memiliki orientasi arah, sedangkan graf berbobot adalah graf yang tiap sisinya memiliki bobot tertentu.



Gambar 2. (a) Graf Sederhana. (b) Graf Berarah. (c) Graf Berbobot

Terdapat beberapa terminologi graf, di antaranya:

1. Ketetanggaan (*Adjacent*)

Dua simpul yang bertetangga adalah simpul yang dihubungkan dengan satu sisi yang sama.

2. Bersisian (*Incidency*)

Suatu sisi bersisian dengan simpul-simpul yang dihubungkannya.

3. Simpul Terpencil (*Isolated Vertex*)

Simpul terpencil adalah simpul yang tidak memiliki sisi yang bersisian dengannya.

4. Derajat (*Degree*)

Derajat adalah jumlah sisi yang bersisian dengan suatu simpul.

5. Lintasan (*Path*)

Lintasan adalah sisi yang ditempuh untuk mencapai dari satu simpul ke simpul lainnya.

6. Sirkuit

Sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.

7. Terhubung (*Connected*)

Dua simpul terhubung apabila ada lintasan yang menghubungkan dua simpul tersebut. Graf terhubung adalah graf yang semua simpulnya terhubung.

8. *Cut-set*

*Cut-set* adalah himpunan sisi yang menyebabkan suatu graf terhubung. Dengan kata lain, apabila sisi tersebut dihilangkan, akan menyebabkan graf menjadi tidak terhubung.

Ada beberapa graf khusus, di antaranya:

1. Graf Lengkap

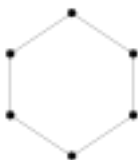
Graf lengkap adalah graf sederhana (tidak memiliki sisi ganda maupun gelang), yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf lengkap dengan  $n$  simpul memiliki jumlah sisi  $n(n-1)/2$ .



Gambar 3. Graf lengkap

2. Graf Lingkaran

Graf lingkaran adalah graf yang setiap simpulnya memiliki dua sisi (berderajat dua).



Gambar 4. Graf lingkaran

3. Graf Teratur

Graf teratur adalah graf yang setiap simpulnya berderajat sama. Jumlah sisi pada graf teratur adalah  $nr/2$ , dengan  $n$  adalah jumlah simpul dan  $r$  adalah derajat tiap

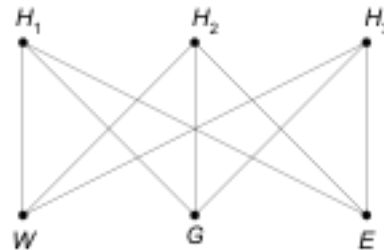
simpulnya.



Gambar 5. Graf lengkap

4. Graf Bipartit

Graf bipartit adalah graf yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian, sedemikian sehingga setiap sisi pada graf tersebut menghubungkan sebuah simpul di himpunan bagian pertama ke sebuah simpul di himpunan bagian kedua.



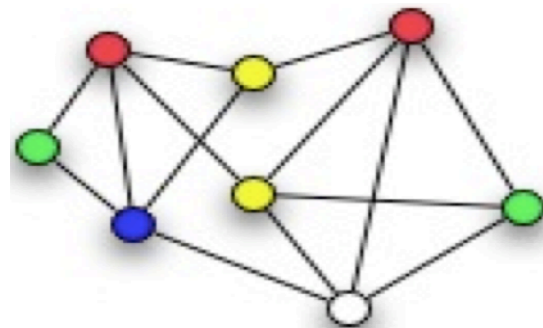
Gambar 6. Graf bipartit

B. Pewarnaan Graf

Pewarnaan graf adalah proses untuk menandai simpul atau sisi suatu graf. Ada tiga macam pewarnaan graf:

1. Pewarnaan Simpul (*Vertex Coloring*)

Pewarnaan simpul adalah memberikan warna pada simpul-simpul sedemikian sehingga simpul yang bertetangga tidak memiliki warna yang sama. Dalam pewarnaan graf, kita tidak hanya mencari warna yang berbeda, tetapi juga mencari cara agar warna yang digunakan sesedikit mungkin. Jumlah warna minimum yang dibutuhkan untuk mewarnai graf disebut bilangan kromatik (simbol:  $\chi(G)$ ). Graf kosong memiliki bilangan kromatik samadengan satu, sedangkan graf lengkap memiliki bilangan kromatik samadengan jumlah simpul. Untuk mewarnai graf planar, dapat digunakan maksimal empat warna.

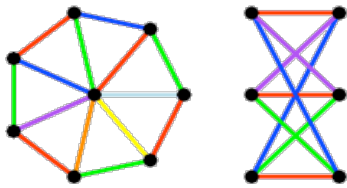


Gambar 7. Pewarnaan simpul

2. Pewarnaan Sisi (*Edge Coloring*)

Pewarnaan sisi adalah memberikan warna pada sisi-sisi dalam graf yang tidak memiliki sisi gelang, sedemikian

sehingga sisi yang berada pada satu simpul tidak berwarna sama. Jumlah warna minimum yang dibutuhkan untuk mewarnai sisi graf disebut indeks kromatik (simbol:  $\chi'(G)$ ). Indeks kromatik pada suatu graf selalu sama dengan derajat maksimumnya atau derajat maksimum tambah satu.



Gambar 8. Pewarnaan sisi

### C. Algoritma Greedy

Algoritma Greedy merupakan algoritma yang sering digunakan dalam implementasi sistem yang berhubungan dengan optimasi. Prinsip algoritma Greedy adalah “take what you can get now”.

Algoritma Greedy menggunakan solusi langkah per langkah. Banyaknya pilihan yang perlu dieksplorasi setiap langkahnya menyebabkan harus membuat keputusan terbaik setiap langkah dalam menentukan pilihan. Keputusan yang telah diambil tidak dapat diubah lagi pada langkah selanjutnya.

Solusi optimum adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. Solusi yang memenuhi semua kendala disebut solusi layak. Solusi layak yang mengoptimalkan fungsi optimasi disebut fungsi optimum.

```

Berikut adalah pseudocode algoritma Greedy[7]:
set Greedy (Set Candidate){
  solution= new Set( );
  while (Candidate.isNotEmpty()) {
    next = Candidate.select(); //use
    selection criteria,
    //remove from candidate and
    return value
    if (solution.isFeasible(next))
  //constraints satisfied
    solution.union(next);
    if (solution.solves()) return
  solution}
  //No more candidates and no solution
  return null
}

```

### E. Algoritma Dijkstra

Algoritma Dijkstra merupakan salah satu algoritma untuk mencari jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik yang lain. Langkah pertama dalam algoritma Dijkstra adalah menentukan titik mana yang dijadikan titik awal, lalu beri bobot yang merepresentasikan jarak dari titik awal tersebut ke titik terdekatnya satu per satu. Algoritma Dijkstra akan melakukan pencarian dari satu titik ke titik lain dan ke titik selanjutnya langkah demi langkah.

```

Berikut adalah pseudocode algoritma Dijkstra[8]:
function Dijkstra(Graph, source):
  for each vertex v in Graph:
    dist[v] := infinity;

```

```

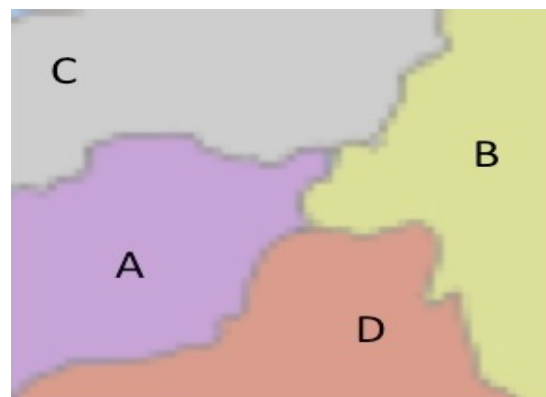
    previous[v] := undefined;
  end for
  dist[source] := 0;
  u := vertex in Q with smallest
  distance in dist[];
  remove u from Q;
  if dist[u] = infinity:
    break;
  end if
  for each neighbor v of u:
    alt := dist[u]+dist_between(u,
v);
    if alt < dist[v]:
      dist[v] := alt;
      previous[v] := u;
      decrease-key v in Q;
    end if
  end for
end while
return dist;

```

### III. PENENTUAN JALUR PENERBANGAN

Pada kasus penentuan jalur dan jadwal penerbangan, ada beberapa hal yang perlu dipertimbangkan, yaitu jarak antarkota, waktu tempuh, dan waktu keberangkatan pesawat.

Sebelum menentukan jadwal penerbangan, sebaiknya ditentukan terlebih dahulu jalur penerbangannya. Untuk memodelkan penentuan jalur penerbangan, penulis membuat contoh kasus. Misalkan ada kota A, B, C, dan D. Terdapat pesawat M dan N yang akan berangkat dari kota A dan B. Kemudian terdapat pesawat O dan P yang akan berangkat dari kota C ke kota D. Lalu, terdapat pesawat Q dan R yang akan berangkat dari kota B ke kota C. Berikut posisi kota-kota tersebut:



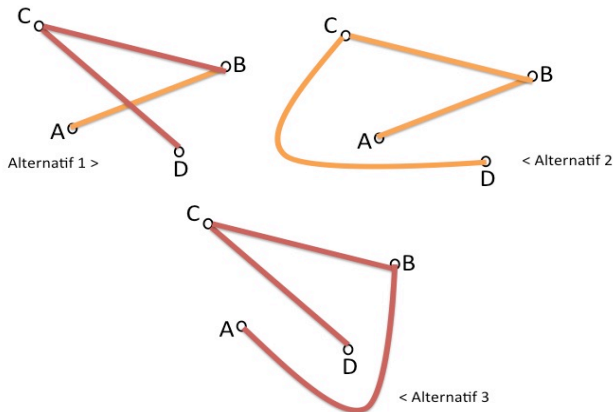
Gambar 9. Posisi kota A, B, C, D

Pada penentuan jalur penerbangan, simpul pada graf merepresentasikan kota, sedangkan sisi pada graf merepresentasikan jalur pesawat. Warna pada sisi-sisi tersebut merepresentasikan level ketinggian pesawat. Jadi, mungkin pewarnaan graf yang dilakukan di sini berbeda dengan pewarnaan sisi graf karena sisi yang bersatu pada satu simpul mungkin memiliki warna yang sama.

Sisi yang berbeda warna merepresentasikan level ketinggian terbang yang berbeda juga. Sebenarnya, level ketinggian pesawat saat terbang tidak mungkin selalu sama karena akan ada halangan-halangan di atas seperti bangunan tinggi, awan, dan sebagainya. Maksud penulis

untuk membedakan ketinggian adalah saat dua pesawat mungkin berpapasan, sehingga tidak menyebabkan tabrakan.

Misalkan warna merah pada sisi merepresentasikan pesawat terbang lebih rendah dari pesawat lainnya, dan warna kuning merepresentasikan pesawat terbang lebih tinggi dari pesawat lainnya. Berikut adalah hasil pewarnaan sisi graf berdasarkan level ketinggian tersebut:



Gambar 10. Beberapa alternatif jalur penerbangan

Karena adanya beberapa alternatif jalur, digunakan algoritma Greedy dan algoritma Dijkstra untuk mendapatkan jalur yang paling pendek dan efisien.

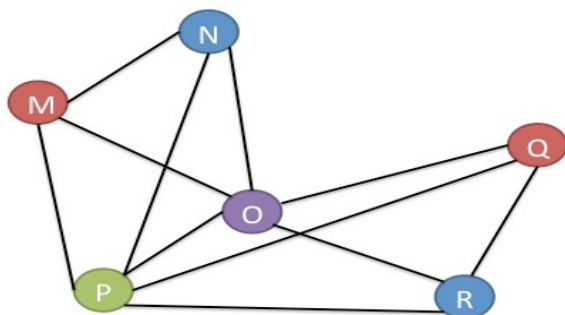
#### IV. PENENTUAN JADWAL PENERBANGAN

Setelah jalur penerbangan didapatkan, kemudian ditentukan jadwal penerbangan. Menentukan jadwal penerbangan harus mempertimbangkan pesawat lain yang akan lepas landas dan mendarat.

Pada penentuan jadwal penerbangan, simpul pada graf merepresentasikan pesawat, sedangkan sisi merepresentasikan hubungan pesawat yang mungkin bertabrakan apabila berangkat bersamaan.

Kemudian, dilakukan pewarnaan simpul sedemikian sehingga pesawat yang berhubungan memiliki warna yang tidak sama. Simpul-simpul berwarna sama merepresentasikan bahwa dua simpul (pesawat) tersebut dapat berangkat bersamaan.

Berikut adalah hasil pewarnaan simpul graf tersebut:



Gambar 11. Hasil pewarnaan simpul

Hasil pewarnaan simpul tersebut berarti pesawat M hanya mungkin terbang bersamaan dengan pesawat Q, pesawat N hanya mungkin terbang bersamaan dengan

pesawat R. Sedangkan yang lainnya harus terbang dengan waktu yang berbeda.

#### V. PENENTUAN JADWAL DAN JALUR PENERBANGAN

Pada dunia nyata, banyak hal-hal lain yang perlu diperhatikan dalam menentukan jadwal dan jalur penerbangan, yaitu:

1. Banyaknya jumlah permintaan

Dalam menentukan jalur dan jadwal, harus diperhatikan banyaknya permintaan penumpang untuk suatu jalur. Misalnya, penumpang dari kota A ke kota B lebih banyak daripada kota C ke kota D, maka jadwal dari kota A ke kota B seharusnya lebih banyak daripada dari kota C ke kota D.

2. Kapasitas penumpang

Berhubungan dengan nomor 1, penentuan jadwal juga bergantung pada kapasitas pesawat dalam menampung penumpang. Jalur yang lebih banyak penumpangnya harus disediakan dari satu jadwal atau satu pesawat.

3. Jenis pesawat

Jenis pesawat juga perlu diperhatikan, apakah pesawat tersebut feasible untuk melewati suatu jalur atau tidak. Selain itu, pesawat dengan kapasitas besar juga akan dialokasikan untuk menempuh rute yang jumlah permintaannya besar.

Jadi, penentuan jalur dan jadwal penerbangan dalam kehidupan nyata tidak akan sesederhana yang penulis contohkan dalam makalah ini, namun tetap bisa dipertimbangkan dengan teori graf.



Gambar 12. Contoh jalur pesawat di kehidupan nyata

#### VI. KESIMPULAN

Kesimpulan yang dapat diambil dari pemaparan di atas antara lain:

- Teori graf sangat bermanfaat dalam kehidupan sehari-hari dan dalam berbagai bidang, salah satunya adalah bidang penerbangan.
- Pewarnaan graf dapat membantu dalam penentuan jadwal dan jalur penerbangan.
- Jalur penerbangan dapat ditentukan dengan mewarnai sisi-sisi pada graf, dengan perbedaan warna merepresentasikan perbedaan level

ketinggian.

- Banyaknya alternatif jalur yang didapat menyebabkan kita harus mencari jalur terpendek dan efisien. Untuk itu, digunakan algoritma pendukung seperti algoritma Greedy dan algoritma Dijkstra.
- Penentuan jadwal penerbangan dapat dilakukan dengan mewarnai simpul-simpul pada graf, dengan perbedaan warna merepresentasikan pesawat yang tidak boleh berangkat bersamaan.
- Contoh yang diberikan penulis pada makalah ini masih sangat sederhana. Pada kehidupan nyata, banyak hal-hal lain yang perlu diperhatikan. Namun, bukan berarti teori graf tidak tetap dapat membantu dalam menentukan jadwal dan jalur penerbangan.

## VII. UCAPAN TERIMA KASIH

Pada bagian ini, penulis ingin mengucapkan terima kasih kepada Allah SWT atas segala rahmat dan berkahnya sehingga penulis dapat menyelesaikan makalah ini. Ucapan terima kasih juga penulis sampaikan kepada orangtua penulis, yang selalu mendukung dan mendoakan penulis tanpa henti. Kemudian, terima kasih juga penulis ucapkan kepada Pak Rinaldi Munir, selaku dosen mata kuliah IF2120 Matematika Diskrit, atas bimbingan, dukungan, dan referensi-referensi yang sangat membantu dalam penyelesaian makalah ini. Terakhir, terima kasih kepada teman-teman dan sahabat atas dukungannya selama ini.

## DAFTAR REFERENSI

- [1] [https://id.wikipedia.org/wiki/Daftar\\_maskapai\\_penerbangan\\_di\\_Asia](https://id.wikipedia.org/wiki/Daftar_maskapai_penerbangan_di_Asia), diakses pada 5 Desember 2015, pukul 11:20
- [2] <https://www.flightradar24.com>, diakses pada 5 Desember 2015, pukul 11:35
- [3] [https://id.wikipedia.org/wiki/Teori\\_graf](https://id.wikipedia.org/wiki/Teori_graf), diakses pada 5 Desember 2015, pukul 12:10
- [4] Munir, Rinaldi. 2003. Matematika Diskrit. Bandung.
- [5] [http://www.academia.edu/3682167/PEWARNAAN\\_graph](http://www.academia.edu/3682167/PEWARNAAN_graph), diakses pada 9 Desember 2015 pukul 11:33
- [6] <http://www.it-jurnal.com/2015/09/pengertian-algoritma-greedy.html>, diakses pada 9 Desember 2015 pukul 12:01
- [7] [http://www.slideshare.net/Krish\\_ver2/51-greedy-02](http://www.slideshare.net/Krish_ver2/51-greedy-02), diakses pada 9 Desember 2015 pukul 13:00
- [8] <https://wirasetiawan29.wordpress.com/2015/04/02/tentang-algoritma-dijkstra/>, diakses pada 9 Desember 2015 pukul 15:21
- [9] <http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2008-1-00264-MTIF-Bab%202.pdf>, diakses pada 10 Desember 2015 pukul 14:07

## 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 2015



Hishshah Ghassani - 13514056