

Penerapan Graf dalam Menentukan Jalur Penerbangan Efektif Dengan Algoritma Graf yang Tepat

Danang Afnan Hudaya (13512056)

Program Studi Teknik Informatika Sekolah
Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512056@std.stei.itb.ac.id; afnanhudaya@gmail.com

Abstract

Akhir-akhir ini sering terjadi kecelekaan penerbangan yang salah satunya dikarenakan menabrak lempengan gunung. Hal ini karena kurangnya pilot dalam mengamati keadaan wilayah sekitar. Untuk itu diperlukan sebuah software yang berguna untuk menganalisa ketinggian pesawat dan ketinggian daratan yang akan dilalui pesawat tersebut beserta pemberian informasi mengenai keamanan jalur tersebut serta pencarian jalur tercepat untuk penerbangan.

I.PENDAHULUAN

1.1 Cara Kerja Software

Software ini menentukan ketinggian pesawat dan ketinggian lempeng yang ada di depan pesawat tersebut dengan memanfaatkan satelit. Pertama, pesawat tersebut mengirim sinyal ke satelit untuk mendapatkan gambar. Setelah itu software akan mengolah informasi yang didapatnya. Hasil dari pengolahan tersebut berupa peta daratan yang ada di sekitar pesawat dengan tambahan pewarnaan yang berbeda berdasar ketinggian. Ketinggian tersebut dibagi beberapa level berdasarkan aman atau tidaknya wilayah tersebut untuk penerbangan. Dalam hal ini software menggunakan sistem untuk menganalisis lebar pesawat dan jarak pesawat antara pesawat dengan lempeng tersebut untuk menentukan keamanannya. Jalur tercepat dalam software ini diperoleh dengan memberikan alternatif jalur aman yang terpendek dari sekian jalur teraman untuk mencapai tujuan.

1.2 Syarat Jalur Aman

Jalur tersebut dikatakan aman jika jalur tersebut ketika dilalui pesawat dengan ketinggian maksimum memiliki selisih yang cukup jauh. Dalam beberapa kasus ada daerah yang dilarang dilalui pesawat. Daerah tersebut

merupakan daerah yang mana disekitarnya banyak dikelilingi lempeng gunung yang memiliki ketinggian yang memenuhi syarat tidak aman dilalui. Meskipun ada jalur alternatif untuk melewatinya namun tingkat keselamatannya kurang dari 50%. Daerah ini pun perlu dihindarkan dari pemilihan alternatif penerbangan. Jalur aman berikutnya yang bisa dijadikan alternatif adalah jalur yang dekat dengan bandara. Karena jika terjadi turbulence pilot dapat mengamankan pesawatnya ke bandara. Yang terakhir adalah pencarian daerah yang mengambil daerah yang banyak memberikan tempat untuk pendaratan darurat. Tempat tersebut berupa dataran rendah yang rata tanpa adanya bangunan. Meski ini akan sulit dicari jika terbang melalui perkotaan atau daerah pegunungan tapi hal ini dapat berguna saat tempat tersebut ada dan dapat memberitahukan langsung kepada pilot untuk segera mengarahkan pesawatnya ke tempat tersebut.

1.3 Syarat Jalur Tercepat

Jalur tercepat sangat dibutuhkan dalam penerbangan. Karena bahan bakar pesawat terbatas sehingga pesawat tidak bisa berlama-lama di udara dan jadwal penerbangan sudah ditentukan sehingga tidak boleh terjadi keterlambatan penerbangan. Berbeda dengan jalur teraman. Jalur tercepat lebih mengutamakan jarak antara pesawat dengan bandara tujuan. Namun dalam hal ini jalur tercepat masih memungkinkan mengambil jalur yang belum memenuhi syarat aman untuk penerbangan. Jalur ini akan berguna jika digabungkan dengan jalur teraman untuk membentuk jalur alternatif. Pengambilan jalur terpendek dari bandara asal ke bandara tujuan dapat dilakukan dengan membuat grafik garis. Fungsi jalur tercepat akan terlihat jika pilot mnghendaki suatu bandara yang mana sebelum mencapai bandara harus melakukan take off ke suatu tempat sebanyak n kali. Atau

sebelum berangkat dari bandara X menuju bandara Y harus melalui bandara Z terlebih dahulu. Dalam kasus tersebut maka akan memunculkan jalur yang berbeda.

1.4 Jalur Alternatif

Jalur ini menggabungkan metode jalur teraman dan jalur tercepat. Cara menggabungkannya terbagi tiga tahap. Adapun dari tiga tahap itu yang pertama adalah menentukan jalur teraman yang perlu dilalui pesawat. Setelah muncul semua jalur teraman yang layak dilalui pesawat serta persen keamanannya kemudian menuju tahap ke 2. Dalam tahap ini berupa penyaringan antara tahap jalur tercepat dengan memberikan persen jalur teraman di atas 70%. Sehingga akan muncul jalur alternatif. Dari sekian jalur alternatif akan muncul pengurutan jalur alternatif I sampai jalur alternatif III. Yang mana jalur tersebut merupakan jalur terbaik berdasarkan bobot persen yang diberikan dari jalur teraman dan jalur tercepat sehingga penyaringan tersebut akan memberikan 3 solusi terbaik yang telah diurutkan sehingga pilot akan fokus pada 3 jalur tersebut. Dalam pelaksanaannya akan mendapatkan berbagai kasus seperti yang beragam yang akan dibahas di bab selanjutnya. Dengan terpilihnya 3 jalur tersebut akan memudahkan pilot dalam menentukan pilihan jalur mana yang harus dilalui dengan cepat dan tepat sehingga terjadinya kemungkinan terjadinya kecelakaan akibat salah pemilihan jalur penerbangan dapat berkurang.

1.5 Jalur utama

Metode ini adalah pengembangan dari jalur alternatif. Jalur utama merupakan jalur yang paling diutamakan untuk dilalui oleh pilot sesuai kebutuhan. Ada dua pilihan kebutuhan. *Safety* atau *Time*. Dua pilihan ini akan mempengaruhi proses kalkulasi dalam menentukan jalur alternatif. Karena jika mengutamakan keselamatan maka akan terpilihlah jalur paling aman yang tercepat namun dalam prosesnya lebih mengutamakan tingkat keamanan yang tinggi. Jika memilih mode tercepat maka akan memberikan jalur paling pendek yang memenuhi syarat aman meski prosentase keamanan lebih rendah jika kita memilih keamanan. Ini akan berdampak pada pengurutan jalur alternatif yang akan

muncul. Adapun lebih rincinya akan dijelaskan di bab selanjutnya.

II. DASAR TEORI

2.1 Pengertian Graf

Alternatif yang digunakan dalam menyelesaikan masalah jalur penerbangan tersebut adalah dengan menggunakan graf. Graf merupakan suatu alat bantu untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. *Graph* G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$. V (*node*) merupakan himpunan tidak kosong dari simpul. E (*edge*) merupakan himpunan sisi yang menghubungkan sepasang simpul. Dan jika suatu *edge* berasal dari suatu simpul dan ujungnya kembali ke simpul yang sama, maka *edge* tersebut dinamakan *loop*. Simpul pada graf umumnya dinomori dengan huruf atau angka. Sedangkan sisi pada *graph* umumnya dinamai dengan himpunan simpul yang dihubungkan oleh sisi tersebut [10].

graf dapat dikelompokkan berdasarkan ada tidaknya *edge*-nya yang paralel atau *loop*, jumlah *node*, berdasarkan ada tidaknya arah pada *edge*-nya, atau ada tidaknya bobot pada *edge*-nya.

Berdasarkan ada tidaknya *edge* yang paralel atau *loop* *graph* terdiri dari graf sederhana dan *graph* tak sederhana. Graf sederhana adalah *graph* yang tidak memiliki sisi ganda dan juga *loop*. Sisi ganda merupakan kondisi ketika dua buah simpul memiliki lebih dari satu sisi. Sedangkan *graph* tak sederhana adalah *graph* yang memiliki sisi ganda dan atau *loop*. Graf tak sederhana dapat dibagi dua, yaitu *graph* semu (*pseudograph*) dan *multiplegraph*. Graf semu adalah graf yang mempunyai *loop* dan *edge* ganda sedangkan *multiplegraph* adalah hanya mempunyai *edge* ganda. Untuk lebih jelasnya akan dijelaskan secara singkat dibawah ini.

Graf

1. Graf sederhana (simple graph).

Graf sederhana merupakan graf tak berarah yang tidak mengandung sisi-ganda dan gelang(*loop*)

2. Graf Ganda (multigraph).

Graf ganda merupakan graf tak berarah yang tidak mengandung gelang (*loop*).

3. Graf semu (Pseudo graph)

Graf semu merupakan graf yang mengandung gelang (loop).

4. Graf berarah (directed graph atau digraph).

Graf berarah merupakan graf yang setiap sisinya mempunyai arah dan diantara dua buah simpul tidak mempunyai dua sisi yang berlawanan.

5. Graf ganda berarah (directed multigraph). Graf ganda berarah merupakan graf berarah yang membolehkan adanya sisi ganda pada graf tersebut (boleh mempunyai dua sisi yang berlawanan antara dua buah simpul).

B. Terminologi Graf

1. Bertetangga (Adjacent)

Jika kedua simpul tersebut terhubung langsung oleh suatu sisi.

2. Bersisian (Incidency)

Suatu sisi e dikatakan bersisian dengan simpul v_1 dan simpul v_2 jika e menghubungkan kedua simpul tersebut.

3. Simpul Terpencil (Isolated Vertex)

Jika suatu simpul tidak mempunyai sisi yang bersisian dengan simpul itu sendiri.

4. Derajat (Degree)

Derajat suatu simpul merupakan jumlah sisi yang bersisian dengan simpul tersebut. Misalkan, suatu simpul v mempunyai 3 buah sisi yang bersisian dengannya maka dapat dikatakan simpul tersebut berderajat 3, atau dinotasikan oleh $d(v) = 3$.

Pada graf diatas :

$$d(P) = d(Q) = d(S) = 5, \text{ sedangkan } d(R) = 3.$$

Derajat sebuah simpul pada suatu graf berarah dijelaskan sebagai berikut :

- $d_{in}(v)$ merupakan jumlah busur yang masuk ke simpul v
- $d_{out}(v)$ merupakan jumlah busur yang keluar dari simpul v

Dengan demikian derajat pada simpul tersebut, diperoleh : $d(v) = d_{in}(v) + d_{out}(v)$.

6. Lintasan (Path)

Lintasan dari suatu simpul awal v_0 ke simpul tujuan v_T di dalam suatu graf G merupakan barisan sebuah sisi atau lebih $(x_0, x_1), (x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n)$ pada G , dimana $x_0 = v_0$ dan $x_n = v_T$. Lintasan ini dinotasikan oleh : $x_0, x_1, x_2, x_3, \dots, x_n$.

- Pada graf tersebut lintasan P, Q, R memiliki panjang

2. Sementara itu

lintasan P, Q, S, R memiliki panjang 3.

- Lintasan P, Q, R, S, P dinamakan siklus atau sirkuit dengan panjang 4.

- Antara simpul P dan U maupun T tidak dapat ditemukan lintasan.

7. Cut-Set

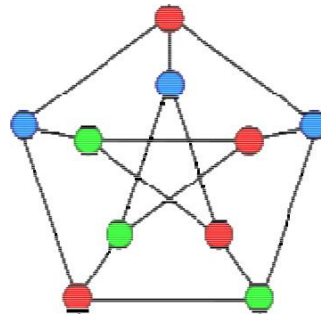
Cut-set dari suatu graf terhubung G adalah himpunan sisi yang jika di buang dari G menyebabkan G tidak terhubung. Jadi, cut-set selalu menghasilkan dua buah subgraf. Pada graf di bawah, $\{(1,4), (1,5), (2,3), (2,4)\}$ adalah cut-set. Terdapat banyak cut-set pada sebuah graf terhubung. Himpunan $\{(1,5), (4,5)\}$ juga adalah cut-set, $\{(1,4), (1,5), (1,2)\}$ adalah cut-set, $\{(5,6)\}$ juga cut-set, tetapi $\{(1,4), (1,5), (4,5)\}$ bukan cut-set sebab himpunan bagiannya, $\{(1,5), (4,5)\}$ adalah cut-set.

2.2 Pewarnaan Graf.

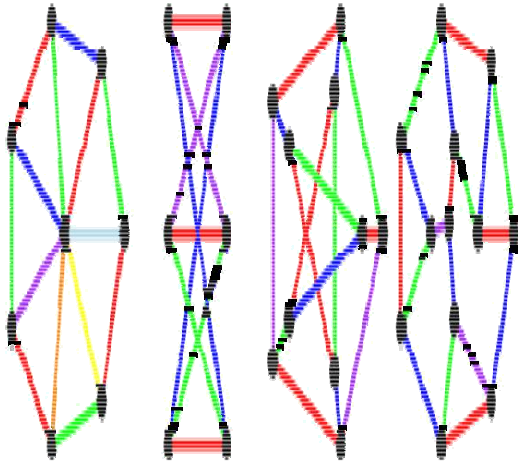
Untuk menentukan jalur teraman maka digunakan metode pewarnaan graf. Yang mana pewarnaan tersebut berdasarkan perbedaan level ketinggian. Sehingga akan lebih mudah dalam menentukan jalurnya dan semakin mudah untuk dilihat jalur mana yang akan memberikan alternatif terbaik.

Ada tiga macam pewarnaan graf.

Pertama, pewarnaan titik yaitu memberikan warna berbeda pada setiap titik yang bertetangga sehingga tidak ada dua titik yang bertetangga dengan warna yang sama.



Kedua, pewarnaan sisi (edge coloring), yaitu memberikan warna berbeda pada sisi yang bertetangga sehingga tidak ada dua sisi yang bertetangga mempunyai warna yang sama.



Ketiga, pewarnaan bidang, yaitu memberikan warna pada bidang sehingga tidak ada bidang yang bertetangga mempunyai warna yang sama.



Metode yang digunakan adalah metode ketiga yaitu menerapkan Pewarnaan daerah pada suatu graf merupakan pemetaan sekumpulan warna ke beberapa daerah yang berada pada graf planar tersebut sedemikian sehingga daerah yang bertetangga dengan ketinggian

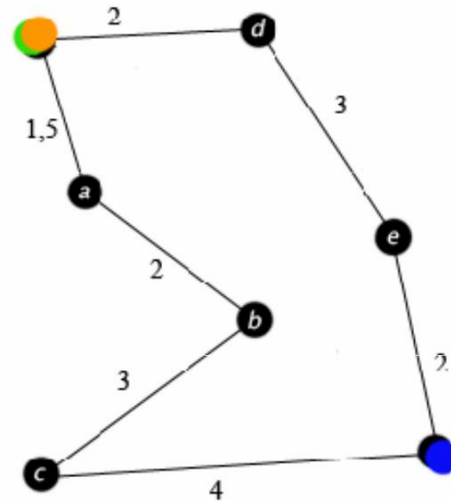
yang berbeda tidak memiliki warna yang sama.

2.3 Graf Jalur Terpendek

Ada beberapa teori antara lain:

1. Algoritma Greedy

Algoritma greedy memiliki kecepatan pencarian yang lebih singkat dibandingkan algoritma a* karena ia hanya membandingkan jarak dari tempat asal dengan tempat lain.



Tetapi algoritma greedy ini memiliki optimasi pencarian lintasan terpendek yang kurang baik karena hal tersebut. Jika ditengah pencarian dia hanya menemukan suatu simpul dan padahal simpul tersebut tersebut jaraknya sangat besar , algoritma greedy tetap memilih jalur tersebut karena tidak dapat kembali lagi ke awal dan tidak ada pilihan lain. Sehingga pencarian tidak optimal, hal yang lebih buruknya adalah gagal dalam pencarian.

Adapun pseudocodenya seperti dibawah ini: 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
}

```

2. Algoritma A*

Berbeda dengan algoritma greedy. Algoritma ini

Menghitung semua kemungkinan dan menyimpannya

sehingga jika setiap memilih jalan. Ia juga membandingkan dengan jalan lain yang disimpan .

Sehingga hasil pencarian lintasan tercapat dengan

menggunakan algoritma ini akan menghasilkan hasil yang optimum.

Namun karena ia terus membandingkan algoritma ini memakan waktu yang cukup lama. Sehingga jika simpulnya sangat banyak akan memakan waktu yang sangat lama.

Adapun Pseudocodenya adalah sebagai berikut:

```

function A*(start,goal)

```

```

closedset := the empty set % The set of nodes already evaluated.

```

```

openset := set containing the initial node % The set of

```

```

tentative nodes to be evaluated. g_score[start] :=
0 % Distance from start along optimal path.

```

```

h_score[start] :=
heuristic_estimate_of_distance(start, goal)

```

```

f_score[start] := h_score[start] % Estimated total

```

```

distance from start to goal through y. while
openset is not empty

```

```

x := the node in openset having the lowest
f_score[]
value

```

```

if x = goal

```

```

return reconstruct_path(came_from,goal) remove
x from openset
add x to closedset

```

```

foreach y in neighbor_nodes(x) if y in
closedset
continue

```

```

tentative_g_score := g_score[x] +
dist_between(x,y)

```

```

if y not in openset
add y to openset

```

```

tentative_is_better := true

```

```

elseif tentative_g_score < g_score[y]

```

```

tentative_is_better := true

```

```

else

```

```

tentative_is_better := false

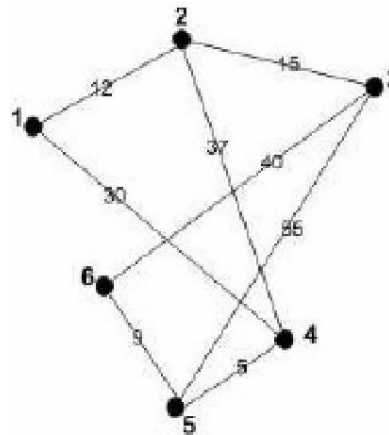
```

```

if tentative_is_better = true
came_from[y] := x
g_score[y] := tentative_g_score
h_score[y] :=
heuristic_estimate_of_distance(y,
goal)
f_score[y] := g_score[y] + h_score[y] return
failure
function reconstruct_path(came

```

Dapat terbaca bahwa pseudocodenya bahwa semua dicari satu persatu. Meski lama dalam pencariannya namun metode ini sangat cocok jika memilih jalur yang mengutamakan keselamatan sedangkan jalur yang mengutamakan cepat sampai dengan metode algoritma greedy. Adapun contoh gambar grafik seperti gambar di bawah ini:



III APLIKASI

Pencarian jalur yang tepat dalam penerbangan akan terbagi dalam beberapa tahap. Pertama, melakukan pewarnaan graf berdasar perbedaan level ketinggian yang terpantau dari satelit dengan ketentuan warna hijau untuk ketinggian rendah dan juga aman untuk dilalui, warna kuning untuk daerah yang memiliki ketinggian sedang, dan merah merupakan daerah yang memiliki ketinggian yang sangat tinggi dan berbahaya untuk dilewati dan sebaiknya harus dihindari. Pada tahap ini tentu akan menghasilkan beberapa pilihan jalur yang berbeda terutama jalur untuk daratan yang berwarna hijau memiliki jalur yang bisa memberi banyak alternatif dengan memberikan prosentase keamanan dengan rumus: $100\% - [(ketinggian\ yang\ diijinkan\ pesawat - ketinggian\ daerah) / ketinggian\ maksimum\ pesawat \times 5\%]$. Kemudian pada

tahap selanjutnya dilakukan pencarian jalur terpendek menuju bandara selanjutnya dengan menggunakan algoritma A* yang akan memberikan alternatif jalur terpendek dengan memperhatikan prosentase keamanan yang telah diberikan dari tahap pertama. Untuk memberikan hasil yang nyata dan dapat diukur maka dalam program tersebut perlu diberikan rumus yang mengkonvert segala kemungkinan menjadi prosentase yang mana pilot dapat menganalisis jalur yang akan dipakai dengan melihat angka kemungkinan yang keluar. Adapun rumus untuk jalur tercepat adalah:

$(\text{jarak/kecepatan-waktu sekarang}) / (\text{waktu sampai-waktu sekarang}) \times 100\%$

Dengan memanfaatkan prosentase akan memudahkan pilot memilih jalur tercepat. Setelah itu menuju tahap ketiga yaitu tahap semua informasi probabilitas pada suatu jalur digabungkan dengan rumus: jalur tercepat+jalur teraman/2. Yang mana setiap jalur akan memberikan prosentase yang berbeda beda. Setelah itu akan dilakukan sorting dan yang akan dimunculkan pertama kali adalah nilai prosentase tertinggi sehingga akan memudahkan pilot dalam mengambil keputusan dengan cepat dan tepat..

IV KESIMPULAN

Untuk mendapatkan sebuah jalur alternatif yang tepat dan akurat diperlukan 2 pertimbangan yaitu jalur tercepat dan teraman yang mana keduanya bisa digabungkan menjadi alternatif utama dan dalam menentukannya perlu rumus agar sebuah perhitungan bisa diukur dan akan memudahkan penentuan tersebut.

V. ACKNOWLEDGEMENT

Ucapan terima kasih Saya ucapkan kepada Pak Rinaldi Munir atas dukungan, bimbingan dan berbagai referensi-referensi yang dapat digunakan untuk penyelesaian makalah ini. Terima kasih saya ucapkan kepada teman-teman prodi IF dan STI yang telah bersedia untuk berdiskusi terhadap berbagai permasalahan dalam makalah ini.

VI. REFERENSI

1. H. Rossen, Kenneth, "Discrete Mathematics and Its Application", McGrawHill, 2013
2. http://id.wikipedia.org/wiki/Teori_graf
Tanggal akses 8 Desember 19.00
3. http://www.airport-technology.com/contractors/consult/qu_intiq/
Tanggal akses 8 Desember 19.20
4. <http://www.iata.org/whatwedo/passenger/scheduling/Pages/index.aspx>
Tanggal akses 8 Desember 19.30

Bandung, 8 Desember 2016



Danang Afnan Hudaya
13512056