

Penerapan Algoritma A* dalam Penentuan Rumah Sakit dengan Jarak Terdekat dalam Kegawatdaruratan

Yudi Alfayat 13519051

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): yalfayat@gmail.com

Abstrak - Kejadian gawat darurat merupakan keadaan dimana seseorang memerlukan pertolongan segera karena apabila tidak mendapatkan pertolongan dengan segera maka bisa mengancam jiwanya atau menimbulkan kecacatan permanen. Dalam keadaan darurat diperlukan penanganan secepatnya, seperti perlunya membawa korban secepatnya ke rumah sakit terdekat. Semakin cepat mencapai rumah sakit maka akan semakin baik untuk keselamatan korban. Hal tersebut sangat terbantu ketika kita mengetahui rute yang paling pendek atau paling cepat menuju rumah sakit. Dengan banyaknya rumah sakit, dan rute untuk menuju rumah sakit tersebut akan menyulitkan saat pemilihan rumah sakit yang paling tepat. Algoritma A* atau A-star adalah algoritma optimal yang sering digunakan untuk pencarian jalan atau rute pada suatu graf. Algoritma A* menggunakan pendekatan algoritma Best First Search dan bobot minimal yang diberikan simpul awal ke simpul tujuan. Algoritma ini dapat menentukan rute pada graf dengan solusi paling optimal. Dengan menggunakan Algoritma A* ini, maka dapat ditentukan rute rumah sakit dengan jarak terdekat.

Kata Kunci : Algoritma A*, rute, rumah sakit, terpendek, tercepat.

I. PENDAHULUAN

Kegawatdaruratan merupakan suatu keadaan dimana dapat mengancam nyawa yang harus dilakukan pertolongan secepatnya untuk mencegah terjadinya cedera atau bahkan kematian. Pertolongan tersebut dapat berupa membawa korban secepatnya ke rumah sakit. Karena dalam keadaan darurat, maka setiap detik waktu akan sangat berharga, sehingga waktu yang dibutuhkan untuk mencapai rumah sakit akan menjadi faktor penentu terhadap keselamatan korban. Dengan banyaknya rumah sakit serta rute untuk menuju ke rumah sakit tersebut, sehingga akan menyulitkan ketika mencari rumah sakit yang tepat dengan jarak terdekat.

Algoritma A* atau A-star merupakan algoritma yang optimal yang sering digunakan untuk pencarian jalan atau rute pada suatu graf. Algoritma ini dapat menemukan rute pada graf dengan solusi paling optimal. Algoritma A* menggunakan pendekatan Best First Search dengan setiap simpulnya memiliki cost atau bobot tersendiri. A* akan membandingkan setiap cost dari simpul dan akan mencari simpul dengan cost paling kecil. Jika dihubungkan dengan suatu peta, maka cost tersebut adalah jarak dari suatu titik ke titik yang lain.

Algoritma A* dapat digunakan untuk mencari atau menentukan rute terpendek atau tercepat ke sebuah rumah sakit. Algoritma A* akan mencari rute terpendek untuk setiap rumah sakit yang ada. Setelah didapatkan semua rute tersebut, maka akan diambil rute dengan jarak paling dekat.

II. LANDASAN TEORI

A. Graf

1) Defenisi Graf

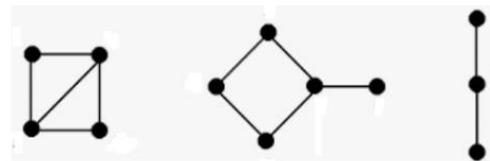
Graf merupakan struktur diskrit yang terdiri dari vertex/simpul dan edge/busur yang menghubungkan simpul-simpul tersebut. Graf dilambangkan sebagai berikut, Graf $G = (V, E)$, V merupakan himpunan simpul yang tidak kosong, dan E merupakan himpunan edge/busur, yang dimana setiap edge-nya menghubungkan dua buah simpul. Edge berbentuk (v_i, v_j) , yang artinya adalah edge yang menghubungkan simpul v_i dan simpul v_j .

2) Jenis-jenis Graf

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:

1. Graf Sederhana (*simple graph*)

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda.



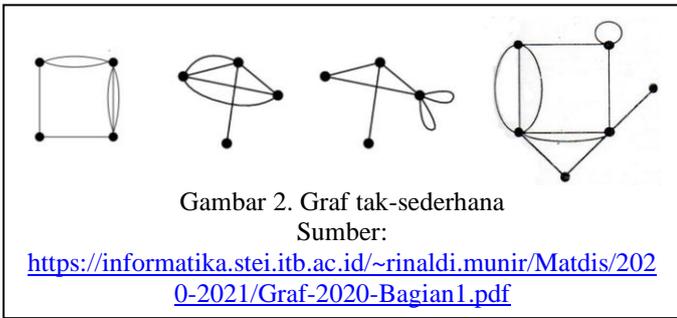
Gambar 1. Graf Sederhana

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

2. Graf tak-sederhana (*unsimple graph*)

Graf tak-sederhana adalah graf yang mengandung sisi gelang atau sisi ganda.

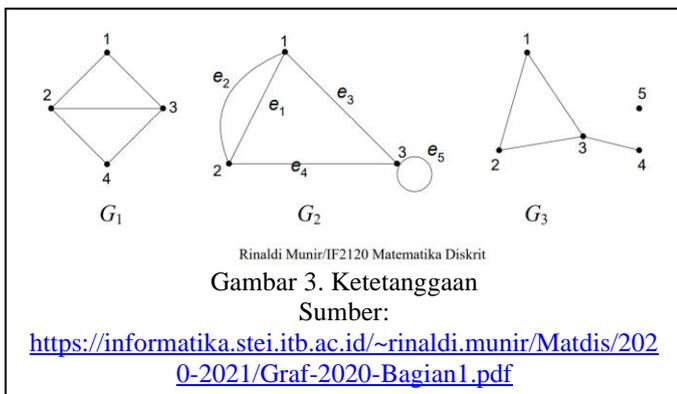


- Berdasarkan arah pada graf, dibedakan atas dua jenis:
1. Graf tak-berarah (*undirected graph*)
Graf yang sisinya tidak mempunyai orientasi arah disebut sebagai graf tak-berarah
 2. Graf berarah (*directed graph*)
Graf yang sisinya mempunyai orientasi arah disebut graf berarah.

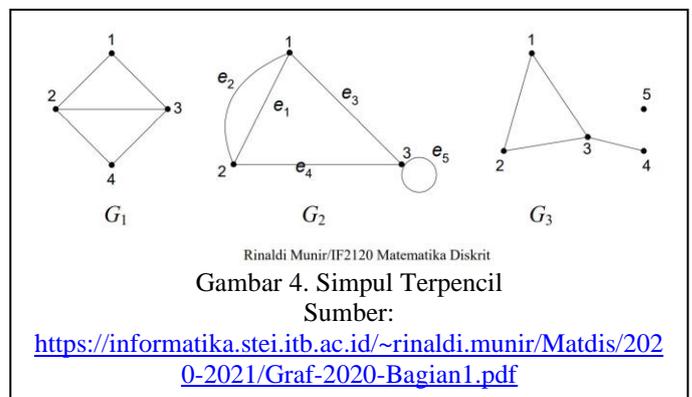
3) Terminologi Graf

1. Ketetanggaan (*Adjacent*)

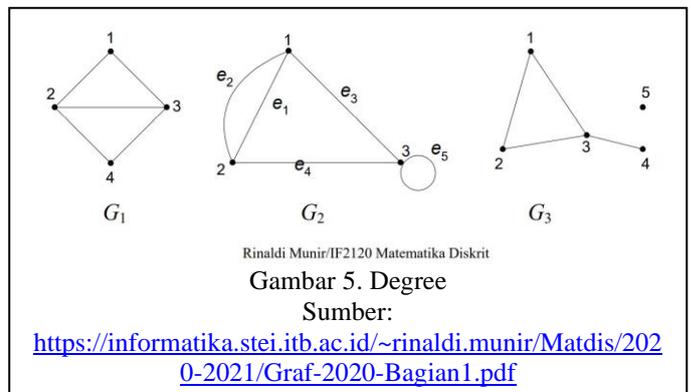
Dua *vertex* dikatakan bertetangga bila keduanya terhubung langsung. Diliat dari graf G_1 : simpul 1 bertetangga dengan simpul 2 dan 3, simpul 1 tidak bertetangga dengan simpul 4.



2. Bersisian (*Incidency*)
Jika ada sisi $e = (v_j, v_k)$, maka sisi e dikatakan bersisian dengan simpul v_j atau sisi e bersisian dengan simpul v_k .
3. Simpul Terpencil (*Isolated Vertex*)
Simpul terpencil merupakan simpul yang tidak mempunyai sisi yang bersisian dengannya. Seperti yang dapat dilihat pada gambar 4, simpul 5 merupakan simpul yang terpencil.

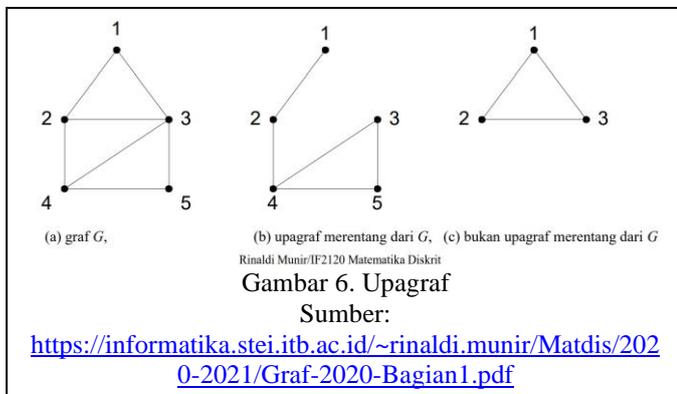


4. Graf Kosong (*empty graph*)
Graf kosong atau *empty graph* merupakan suatu graf yang tidak mempunyai sisi-sisi atau bisa disebut himpunan sisinya adalah himpunan kosong.
5. Derajat (*Degree*)
Derajat merupakan jumlah sisi yang bersisian dengan pada simpul. Notasi dari *degree* adalah $d(v)$. Seperti yang dapat dilihat pada gambar 5. Pada graf G_1 , $d(1) = d(4) = 2$ dan $d(2) = d(3) = 3$. Pada graf G_3 , $d(5) = 0$ karena merupakan simpul terpencil, dan $d(4) = 1$.

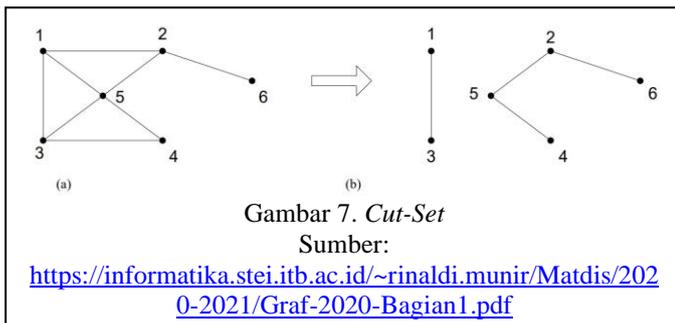


6. Lintasan (*Path*)
Jika suatu graf memiliki sisi seperti berikut, $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ..., $e_n = (v_{n-1}, v_n)$. Maka lintasan yang terbentuk adalah $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$.
7. Siklus (*cycle*) atau Sirkuit (*circuit*)
Siklus atau sirkuit merupakan suatu lintasan yang berawal dan berakhir pada simpul yang sama. Seperti yang dapat dilihat pada gambar 5, pada graf G_1 : 1, 2, 3, 1 merupakan sebuah sirkuit.

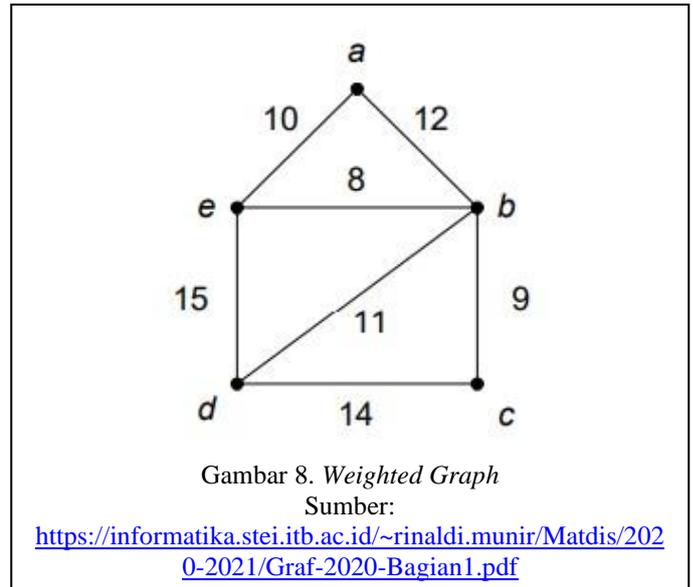
8. Keterhubungan (*Connected*)
Dua buah simpul v_1 dan v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2 . Suatu graf disebut **graf terhubung** (*connected graph*) jika terdapat lintasan untuk setiap pasangan simpul pada graf tersebut.
9. Upagraf Merentang (*Spanning Subgraph*)
Upagraf dari suatu *graph* disebut upagraf rentang jika semua simpul dari graf tersebut ada pada upagraf. Contoh jika upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$, maka $V_1 = V$.



10. *Cut-Set*
Pada graf yang terhubung, terdapat *cut-set* yang dimana merupakan himpunan sisi yang jika diputus akan membuat graf tersebut tidak terhubung lagi. *Cut-set* pada graf selalu menghasilkan dua buah komponen. Pada graf gambar dibawah ini, sisi (1, 2), (1, 5), (3, 5), dan (3, 4) merupakan sebuah *cut-set*.



11. Graf Berbobot (*Weighted Graph*)
Graf berbobot merupakan sebuah graf yang dimana setiap sisinya mempunyai harga atau bobot.



B. Algoritma A*

Algoritma A* atau A-Star adalah algoritma yang optimal yang sering digunakan untuk pencarian jalan atau rute pada suatu graf. Algoritma ini diciptakan oleh Peter Hart, Nils Nilsson, dan Bertram Raphael pada tahun 1968.

Secara konsep, Algoritma ini membagi graf menjadi dua bagian, yang pertama yaitu simpul-simpul yang dapat dilalui yang disebut dengan *Open List*. Sedangkan yang kedua adalah simpul-simpul yang tidak dapat dilalui atau disebut dengan *Close List*. *Close List* berfungsi sebagai pembatas agar ketika dilakukan iterasi, algoritma tidak akan mengecek lagi simpul-simpul yang ada di dalam *close list*, sehingga tidak akan terjadi pengecekan suatu simpul secara terus menerus. Algoritma ini akan berhenti ketika simpul-simpul yang ada pada *Open List* sudah habis atau ketika algoritma sudah mencapai tujuan.

Algoritma A* menggunakan pendekatan algoritma *Best First Search* dan bobot minimal yang diberikan simpul awal ke simpul tujuan. Algoritma ini akan membandingkan setiap simpul berdasarkan *cost*-nya, yang dimana akan mengambil simpul dengan *cost* yang paling minimum. Rumus untuk menghitung *cost* tersebut adalah sebagai berikut,

$$f(n) = g(n) + h(n)$$

yang dimana,

$f(n)$ = Biaya evaluasi

$g(n)$ = Biaya atau *cost* yang sebenarnya, yang dimana merupakan jarak dari simpul awal ke simpul n.

$h(n)$ = Biaya atau *cost* taksiran dari simpul n ke simpul tujuan.

$h(n)$ adalah nilai Heuristik yang didapatkan dari jarak secara garis lurus dari suatu simpul ke simpul tujuan. $h(n)$ dapat dihitung dengan menggunakan rumus jarak *Euclidean* :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

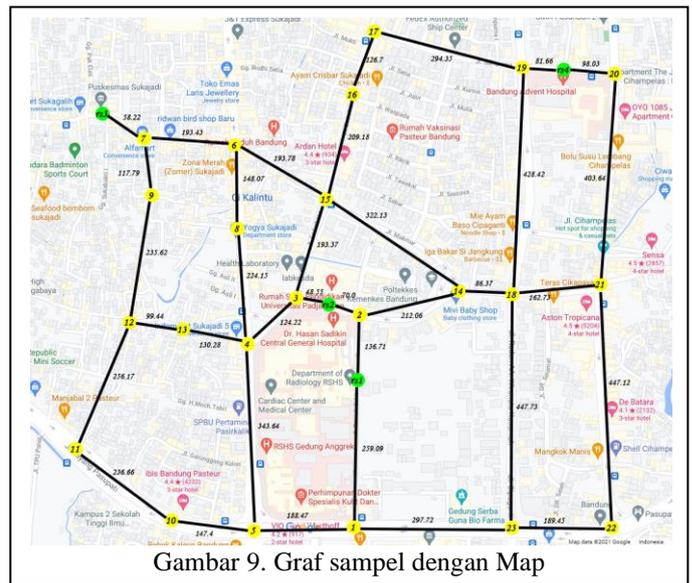
Langkah-langkah pencarian rute terpendek dengan menggunakan Algoritma A*:

1. Hitung semua nilai Heuristik setiap simpul ke simpul tujuan.
2. Inisialisasi simpul ekspan yang pertama adalah simpul awal.
3. Bangkitkan semua simpul hidup yang merupakan tetangga dari simpul ekspan beserta dengan nilai *cost*-nya yang dimana sesuai dengan rumus $f_n = g_n + h_n$. Jika tidak ada lagi simpul hidup baru yang bisa dibangkitkan dan simpul bukan merupakan simpul tujuan, maka iterasi dihentikan, dan rute yang dicari **tidak ditemukan**.
4. Ambil simpul hidup dengan nilai *cost* yang paling minimum dan set sebagai simpul ekspan berikutnya.
5. *Check* apakah simpul tersebut merupakan simpul tujuan atau tidak. Jika iya, maka iterasi akan diberhentikan dan rute yang dicari **sudah ditemukan**. Sedangkan jika tidak, kembalilah ke **langkah ke 3**.

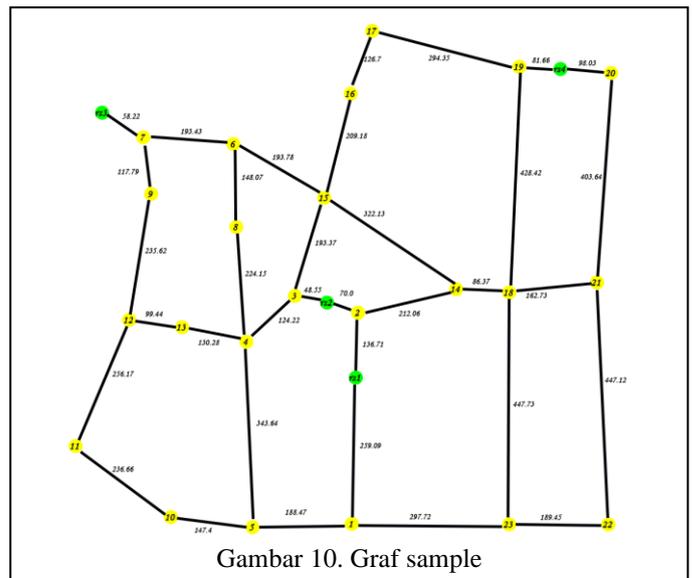
III. PEMBAHASAN

A. Pengambilan Graf dari Google Map

Pada makalah ini, graf yang digunakan adalah graf yang diambil dari *google map*, sehingga dapat memberikan gambaran pada lokasi aslinya. Setiap titik ditandai secara manual, lalu diambil titik koordinatnya. Untuk menghitung bobot sisinya digunakan rumus jarak *Euclidean*. Skala yang dipakai pada graf dibawah ini adalah 1:100000. Sehingga bobot dari setiap sisi dalam satuan meter (m).



Gambar 9. Graf sampel dengan Map



Gambar 10. Graf sample

Pada graf diatas, terdapat 4 *node*/simpul rumah sakit yang dimana *node* rs1 adalah Poliklinik RSHS, *node* rs2 adalah RS Unpad, *node* rs3 adalah Puskesmas Sukajadi, dan *node* rs4 adalah Rumah Sakit Advent Bandung. Baik itu rumah sakit atau pun puskesmas, pada graf ini tetap dianggap sebagai *node* yang sama yaitu rumah sakit. Pada graf juga terdapat 23 buah *node*/simpul yang bukan rumah sakit. Simpul tersebut merupakan persimpangan jalan pada *map*/peta. Berikut daftar simpul/*node* pada graf tersebut, beserta koordinatnya pada peta.

Daftar Node :

- Node 1 : (-6.900177, 107.599281)
- Poliklinik RSHS : (-6.897587, 107.59934)
- Node 2 : (-6.89622, 107.599383)
- RS Unpad : (-6.89605, 107.598702)
- Node 3 : (-6.895941, 107.598225)
- Node 4 : (-6.896772, 107.597308)
- Node 5 : (-6.900207, 107.597396)
- Node 6 : (-6.893059, 107.597044)
- Node 7 : (-6.892846, 107.595124)
- Puskesmas Sukajadi : (-6.8924, 107.594747)
- Node 8 : (-6.894535, 107.597161)
- Node 9 : (-6.893973, 107.595463)
- Node 10 : (-6.90014, 107.595924)
- Node 11 : (-6.898659, 107.594079)
- Node 12 : (-6.896294, 107.595066)
- Node 13 : (-6.896436, 107.596048)
- Node 14 : (-6.895749, 107.60145)
- Node 15 : (-6.894066, 107.598703)
- Node 16 : (-6.892065, 107.599309)
- Node 17 : (-6.890828, 107.599583)
- Node 18 : (-6.895802, 107.602308)
- Node 19 : (-6.89152, 107.602447)
- Rumah Sakit Advent Bandung : (-6.891499, 107.603262)
- Node 20 : (-6.891691, 107.604228)
- Node 21 : (-6.895717, 107.603938)
- Node 22 : (-6.900183, 107.604146)
- Node 23 : (-6.900279, 107.602258)

B. Pencarian Rute Terpendek untuk Setiap Rumah Sakit Menggunakan Algoritma A*

Pencarian rute pada pembahasan ini yaitu dari simpul/*node* **11** yang posisinya terletak pada koordinat **(-6.898659, 107.594079)** ke semua *node* rumah sakit yang ada. **Setelah mendapatkan semua rutenya, maka akan diambil rute ke rumah sakit dengan jarak yang paling minimum sebagai rekomendasi.**

Sebelum melakukan iterasi pada graf, nilai Heuristik dari setiap simpul harus dihitung terlebih dahulu. Dengan

menggunakan rumus *Euclidean* dan koordinat yang sudah didapatkan, berikut nilai Heuristik untuk setiap *node* rumah sakit,

- Heuristik Untuk Poliklinik RSHS (RS1)
 1. Node 1 : 259.09 m
 2. Poliklinik RSHS : 0.0 m
 3. Node 2 : 136.71 m
 4. RS Unpad : 166.51 m
 5. Node 3 : 198.75 m
 6. Node 4 : 219.4 m
 7. Node 5 : 326.35 m
 8. Node 6 : 507.71 m
 9. Node 7 : 634.62 m
 10. Puskesmas Sukajadi : 692.82 m
 11. Node 8 : 375.16 m
 12. Node 9 : 529.92 m
 13. Node 10 : 426.62 m
 14. Node 11 : 537.24 m
 15. Node 12 : 447.1 m
 16. Node 13 : 349.1 m
 17. Node 14 : 279.49 m
 18. Node 15 : 357.88 m
 19. Node 16 : 552.19 m
 20. Node 17 : 676.31 m
 21. Node 18 : 346.32 m
 22. Node 19 : 681.53 m
 23. Rumah Sakit Advent Bandung : 724.15 m
 24. Node 20 : 765.51 m
 25. Node 21 : 495.89 m
 26. Node 22 : 546.29 m
 27. Node 23 : 396.76 m
- Heuristik Untuk RS Unpad (RS2)
 1. Node 1 : 416.76 m
 2. Poliklinik RSHS : 166.51 m
 3. Node 2 : 70.0 m
 4. RS Unpad : 0.0 m
 5. Node 3 : 48.55 m
 6. Node 4 : 157.18 m
 7. Node 5 : 435.7 m
 8. Node 6 : 341.88 m
 9. Node 7 : 480.29 m
 10. Puskesmas Sukajadi : 537.98 m
 11. Node 8 : 216.1 m
 12. Node 9 : 384.43 m
 13. Node 10 : 494.36 m
 14. Node 11 : 530.89 m
 15. Node 12 : 364.74 m
 16. Node 13 : 268.3 m
 17. Node 14 : 276.3 m
 18. Node 15 : 198.41 m
 19. Node 16 : 403.14 m
 20. Node 17 : 529.63 m
 21. Node 18 : 361.72 m
 22. Node 19 : 587.82 m
 23. Rumah Sakit Advent Bandung : 644.41 m

- 24. Node 20 : 703.63 m
- 25. Node 21 : 524.43 m
- 26. Node 22 : 683.8 m
- 27. Node 23 : 552.5 m
- Heuristik Untuk Puskesmas Sukajadi (RS3)
 1. Node 1 : 900.13 m
 2. Poliklinik RSHS : 692.82 m
 3. Node 2 : 600.35 m
 4. RS Unpad : 537.98 m
 5. Node 3 : 496.41 m
 6. Node 4 : 506.44 m
 7. Node 5 : 824.39 m
 8. Node 6 : 238.91 m
 9. Node 7 : 58.22 m
 10. Puskesmas Sukajadi : 0.0 m
 11. Node 8 : 322.05 m
 12. Node 9 : 172.88 m
 13. Node 10 : 782.87 m
 14. Node 11 : 629.53 m
 15. Node 12 : 390.69 m
 16. Node 13 : 423.97 m
 17. Node 14 : 748.93 m
 18. Node 15 : 428.89 m
 19. Node 16 : 457.46 m
 20. Node 17 : 508.6 m
 21. Node 18 : 829.09 m
 22. Node 19 : 774.82 m
 23. Rumah Sakit Advent Bandung : 856.19 m
 24. Node 20 : 950.22 m
 25. Node 21 : 976.65 m
 26. Node 22 : 1220.37 m
 27. Node 23 : 1088.31 m

- Heuristik Untuk Rumah Sakit Advent Bandung (RS4)
 1. Node 1 : 954.82 m
 2. Poliklinik RSHS : 724.15 m
 3. Node 2 : 611.31 m
 4. RS Unpad : 644.41 m
 5. Node 3 : 671.45 m
 6. Node 4 : 795.66 m
 7. Node 5 : 1050.0 m
 8. Node 6 : 641.06 m
 9. Node 7 : 825.13 m
 10. Puskesmas Sukajadi : 856.19 m
 11. Node 8 : 681.69 m
 12. Node 9 : 818.03 m
 13. Node 10 : 1133.71 m
 14. Node 11 : 1164.68 m
 15. Node 12 : 950.05 m
 16. Node 13 : 874.45 m
 17. Node 14 : 462.17 m
 18. Node 15 : 523.49 m
 19. Node 16 : 399.24 m
 20. Node 17 : 373.81 m
 21. Node 18 : 440.74 m
 22. Node 19 : 81.66 m

- 23. Rumah Sakit Advent Bandung : 0.0 m
- 24. Node 20 : 98.03 m
- 25. Node 21 : 427.12 m
- 26. Node 22 : 872.91 m
- 27. Node 23 : 883.76 m

1) Rute dari Node 11 ke Poliklinik RSHS (node Rs1)

Iterasi : simpul ekspansi	Simpul hidup (rute) : $g + h = f$
Iterasi 1. 11 ()	<ul style="list-style-type: none"> • 10(11) : $236.66 + 426.62 = 663.28$ • 12(11) : $256.17 + 447.1 = 703.27$
Iterasi 2. 10(11)	<ul style="list-style-type: none"> • 12(11) : $256.17 + 447.1 = 703.27$ • 5(10,11) : $384.06 + 326.35 = 710.41$
Iterasi 3. 12(11)	<ul style="list-style-type: none"> • 5(10,11) : $384.06 + 326.35 = 710.41$ • 9(12,11) : $491.79 + 529.92 = 1021.71$ • 13(12,11) : $355.57 + 349.1 = 704.67$
Iterasi 4. 13(12, 11)	<ul style="list-style-type: none"> • 5(10,11) : $384.06 + 326.35 = 710.41$ • 9(12,11) : $491.79 + 529.92 = 1021.71$ • 4(13,12,11) : $485.85 + 219.4 = 705.25$
Iterasi 5. 4 (13, 12, 11)	<ul style="list-style-type: none"> • 5(10,11) : $384.06 + 326.35 = 710.41$ • 9(12,11) : $491.79 + 529.92 = 1021.71$ • 3(4,13,12,11) : $610.07 + 198.75 = 808.82$ • 8(4,13,12,11) : $710 + 375.16 = 1085.16$
Iterasi 6. 5 (10, 11)	<ul style="list-style-type: none"> • 9(12,11) : $491.79 + 529.92 = 1021.71$ • 3(4,13,12,11) : $610.07 + 198.75 = 808.82$ • 8(4,13,12,11) : $710 + 375.16 = 1085.16$ • 1(5,10,11) : $572.53 + 259.09 = 831.62$ • 4(5,10,11) : $727.7 + 219.4 = 947.1$
Iterasi 6. 3(4, 13, 12, 11)	<ul style="list-style-type: none"> • 9(12,11) : $491.79 + 529.92 = 1021.71$ • 8(4,13,12,11) : $710 + 375.16 = 1085.16$ • 1(5,10,11) : $572.53 + 259.09 = 831.62$ • 4(5,10,11) : $727.7 + 219.4 = 947.1$ • Rs2(3,4,13,12,11) : $658.66 + 166.51 = 825.13$ • 15(3,4,13,12,11) : $803.44 + 357.88 = 1161.32$
Iterasi 7. Rs2(3, 4, 13, 12, 11)	<ul style="list-style-type: none"> • 9(12,11) : $491.79 + 529.92 = 1021.71$ • 8(4,13,12,11) : $710 + 375.16 = 1085.16$ • 1(5,10,11) : $572.53 + 259.09 = 831.62$ • 4(5,10,11) : $727.7 + 219.4 = 947.1$ • 15(3,4,13,12,11) : $803.44 + 357.88 = 1161.32$ • 2(rs2,3,4,13,12,11) : $728.62 + 136.71$

	= 865.33
Iterasi 8. 1(5, 10, 11)	<ul style="list-style-type: none"> • 9(12,11) : 491.79 + 529.92 = 1021.71 • 8(4,13,12,11) : 710 + 375.16 = 1085.16 • 4(5,10,11) : 727.7 + 219.4 = 947.1 • 15(3,4,13,12,11) : 803.44 + 357.88 = 1161.32 • 2(rs2,3,4,13,12,11) : 728.62 + 136.71 = 865.33 • Rs1(1,5,10,11) : 831.62 + 0 = 831.62 • 23(1,5,10,11) : 870.25 + 396.76 = 1267.01
Iterasi 9 STOP. Rs1(1, 5, 10, 11)	
Jarak : 831.62 m	

Pada iterasi ke 9, simpul yang diekspan adalah simpul tujuan atau target sehingga iterasi dihentikan dan rutanya sudah ditemukan. Rute yang didapatkan adalah sebagai berikut,
Node 11 - Node 10 - Node 5 - Node 1 - Node Rs1 (Poliklinik RSHS),
 dengan jaraknya adalah sejauh 831.62m.

2) Rute dari **Node 11** ke **RS UNPAD (node Rs2)**

Dengan melakukan semua langkah-langkah yang sama dengan mencari rute pada nomor 1), maka rute minimum dari *node 11* ke RS UNPAD yang didapatkan adalah
Node 11 - Node 12 - Node 13 - Node 4 - Node 3 - Rs2 (RS UNPAD).
 Dengan jaraknya adalah sejauh 658.66m.

3) Rute dari **Node 11** ke **Puskesmas Sukajadi (Rs3)**

Dengan melakukan semua langkah-langkah yang sama dengan mencari rute pada nomor 1), maka rute minimum dari *node 11* ke Puskesmas Sukajadi yang didapatkan adalah
Node 11 - Node 12 - Node 9 - Node 7 - Rs3 (Puskesmas Sukajadi).
 Dengan jaraknya adalah sejauh 667.8 m.

4) Rute dari **Node 11** ke **Rumah Sakit Advent Bandung (R4)**

Dengan melakukan semua langkah-langkah yang sama dengan mencari rute pada nomor 1), maka rute minimum dari *node 11* ke Rumah sakit Advent Bandung yang didapatkan adalah
Node 11 - Node 12 - Node 13 - Node 4 - Node 3 - RS Unpad - Node 2 - Node 14 - Node 18 - Node 19 - Rs4 (Rumah Sakit Advent Bandung).

Dengan jaraknya adalah sejauh 1537.17 m.

C. Rumah Sakit yang direkomendasikan

Berdasarkan perhitungan dengan menggunakan Algoritma A* pada bagian B di atas, maka rute untuk setiap rumah sakit yang didapatkan adalah sebagai berikut.

- Poliklinik RSHS : 831.62 m
Node 11 - Node 10 - Node 5 - Node 1 - Node Rs1 (Poliklinik RSHS).
- RS UNPAD : 658.66 m
Node 11 - Node 12 - Node 13 - Node 4 - Node 3 - Rs2 (RS UNPAD).
- Puskesmas Sukajadi : 667.8 m
Node 11 - Node 12 - Node 9 - Node 7 - Rs3 (Puskesmas Sukajadi)
- Rumah Sakit Advent Bandung : 1537.17 m
Node 11 - Node 12 - Node 13 - Node 4 - Node 3 - RS Unpad - Node 2 - Node 14 - Node 18 - Node 19 - Rs4 (Rumah Sakit Advent Bandung)

Maka yang menjadi rekomendasi untuk posisi **Node 11** adalah RS UNPAD pada *node Rs2*, karena memiliki jarak yang paling minimum yaitu 658.66 m.

D. Pencarian rute rumah sakit yang direkomendasikan dengan menggunakan program

Berdasarkan gambar dibawah ini, hasil yang didapatkan dengan menggunakan program atau aplikasi sama dengan hasil yang didapatkan pada perhitungan dibagian B sebelumnya. Rumah sakit yang direkomendasikan adalah RS UNPAD dengan jarak sejauh 658.66 m dari *Node 11*.



Gambar 11. Rekomendasi Rute dengan menggunakan program atau aplikasi

IV. KESIMPULAN

Ketika dihadapkan pada suatu keadaan gawat darurat, baik itu karena kecelakaan, bencana dan hal lainnya. Sangat sering kita membutuhkan rumah sakit untuk menolong para korban dari peristiwa tersebut. Semakin cepat dapat mencapai rumah sakit maka akan semakin baik untuk keselamatan korban. Hal tersebut sangat akan terbantu ketika kita mengetahui rute yang paling pendek atau paling cepat untuk menuju rumah sakit.

Algoritma A* dapat digunakan untuk mencari rute paling pendek dengan solusi paling optimal. Dengan menggunakan algoritma ini, kita dapat menentukan rumah sakit terdekat dari posisi kejadian, dengan rute yang paling pendek untuk menuju rumah sakit tersebut.

Seperti yang sudah dibahas sebelumnya. Jika posisi kejadian berada pada *node* 11 di graf pada gambar 11 di atas, maka kita akan mendapatkan 4 rute untuk menuju 4 buah rumah sakit yang berbeda. Dari keempat rute tersebut, yang memiliki jarak paling pendek adalah menuju ke rumah sakit RS UNPAD dengan jarak 658.66 m, maka RS Unpad akan direkomendasikan untuk dituju.

VIDEO LINK AT YOUTUBE

<https://youtu.be/Njm9KMJqn1c>

UCAPAN TERIMAKASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa karena atas rahmat-Nya penulis dapat menyelesaikan penulisan makalah ini. Penulis juga berterima kasih kepada Tim Dosen Mata Kuliah IF211 Strategi Algoritma, atas ilmu yang telah diberikan selama satu semester ini. Serta ucapan terimakasih, juga penulis sampaikan kepada teman-teman prodi IF yang telah membantu penulis selama satu semester di Mata Kuliah Strategi Algoritma ini.

REFERENSI

- [1] <https://sinta.unud.ac.id/uploads/wisuda/1108605031-3-bab2.pdf>
Tanggal akses : 11 Mei 2021
- [2] <https://core.ac.uk/download/pdf/287227509.pdf>
Tanggal akses : 11 Mei 2021
- [3] R. Munir, Matematika Diskrit. Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2003.

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.

Mungo, 11 Mei 2021



Yudi Alfayat 13519051