

Aplikasi Graph dan Algoritma Dijkstra untuk Harga Tiket Pesawat Termurah Antar Kota

Arief Darmawan Tantriady - 13518015

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13518015@std.stei.itb.ac.id

Abstract— Graf adalah salah satu representasi data yang cukup terkenal dan berdampak pada kehidupan sehari-hari. Salah satu dampak yang akan dibahas pada makalah ini yaitu untuk mencari harga tiket pesawat termurah dari kota A ke kota B. Dengan memanfaatkan graf dan algoritma Dijkstra dapat dicari harga tiket pesawat termurah dari kota A ke kota B.

Kata Kunci—Graf, Dijkstra

I. PENDAHULUAN

Graf adalah salah satu representasi data yang memiliki banyak implementasi dalam kehidupan sehari-hari. Banyak algoritma yang telah ditemukan untuk pengaplikasian graf, misalnya algoritma djistra, algoritma *breadth-first search*, algoritma *depth-first search*, dll.

Pada makalah ini, untuk membantu efisiensi perairan pada sawah akan digunakan algoritma djikstra. Algoritma Dijkstra ini merupakan algoritma yang biasa digunakan untuk mencari *shortest-path* pada sebuah graph. Graf adalah suatu representasi data yang memiliki komponen berupa simpul (*vertice*) dan sisi (*edge*). Graf ada 2 jenis berdasarkan berbobot atau tidaknya, yaitu *weighted graph* dan *unweighted graph*. Pada bagian akan dibahas bagaimana pengaplikasian algoritma Dijkstra pada graph berbobot yang dapat menggambarkan kondisi perairan pada sawah. Semua daerah yang ingin dialiri akan digambarkan sebagai simpul (*vertice*) dan pipa yang mengalir air di sawah akan digambarkan sebagai *sisi (edge)* yang memiliki bobotnya masing-masing.

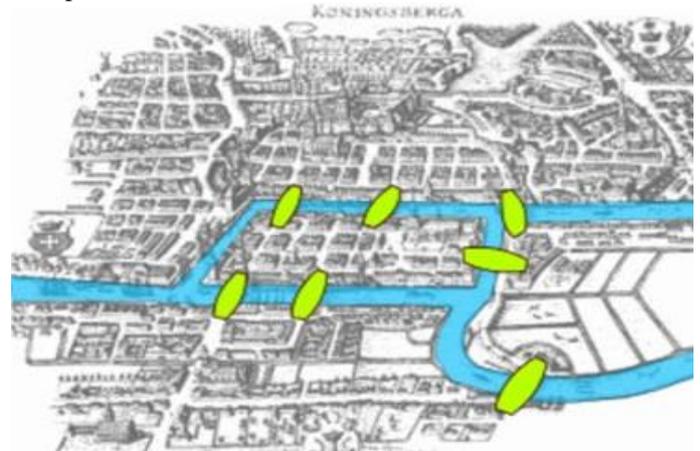
II. LANDASAN TEORI

A. Graf

Graf adalah sebuah representasi data yang mengandung simpul-simpul dan sisi-sisi yang menghubungkan antarsimpul tersebut. Umumnya, graf digunakan untuk menggambarkan hubungan antara objek-objek, dengan objek dinyatakan sebagai simpul dan hubungan antarobjek dinyatakan sebagai sisi.

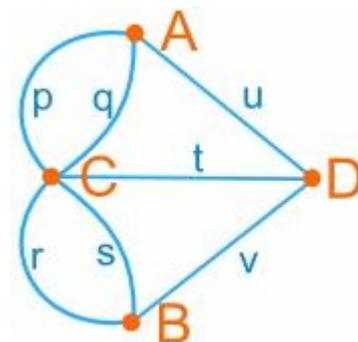
Teori graf pertama kali ditemukan pada tahun 1736 yaitu pada kasus Jembatan Konigsberg. Di Kota Konigsberg atau yang sekarang bernama Kota Kaliningrad, terdapat sungai Pregal yang mengalir mengitari Pulau Kneighof lalu bercabang menjadi dua buah anak sungai. Daratan yang terbelah oleh

sungai tersebut dihubungkan oleh tujuh buah jembatan. Masalah Konigsberg ini mempertanyakan apakah mungkin setiap jembatan dilalui tepat satu kali dan kembali lagi ke tempat semula, namun pada saat belum ditemukannya teori graf tidak ada penjelasan lain selain menggunakan cara coba-coba. Kemudian akhirnya, pada tahun 1736, Leonhard Euler, berhasil menemukan jawaban masalah Konigsberg dengan pembuktian sederhana (bukan coba-coba). Masalah Konigsberg ini kemudian dimodelkan sebagai graf, yaitu daratan berupa titik-titik (*simpul*) yang dihubungkan oleh jembatan dan jembatan dinyatakan sebagai garis yang menghubungkan titik-titik (*sisi*). Setiap daratan diberi label huruf A,B,C dan D.



Visualisasi Jembatan Konigsberg

Sumber : <http://majalah1000guru.net/2013/08/jembatan-konigsberg/>



Jembatan Konigsberg dalam Representasi Graf

Sumber : <http://majalah1000guru.net/2013/08/jembatan-konigsberg/>.

Jawaban yang dikemukakan oleh Euler adalah tidak mungkin dapat melalui ketujuh jembatan masing-masing satu kali dan kembali ke tempat semula jika derajat tiap simpul tidak seluruhnya genap. Yang dimaksud dengan derajat adalah jumlah sisi yang terhubung dengan simpul (jumlah jembatan yang terhubung dengan sebuah daratan). Dalam kasus ini, simpul A,B,dan D berderajat tiga, sedangkan simpul C berderajat lima. Karena tidak semua simpul memiliki derajat genap, maka tidak mungkin dilakukan perjalanan berupa sirkuit (sirkuit euler) pada graf tersebut.

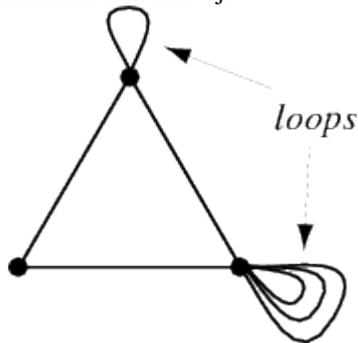
Secara umum, graf memiliki persamaan

$$G = (V,E)$$

dengan G adalah graf, V adalah himpunan tak kosong dari simpul dan E adalah sisi- sisi yang menghubungkan antar simpul.

Berdasarkan sumber referensi dari bahan ajaran *Matematika Diskrit*, tulisan Rinaldi Munir, graf terdiri dari beberapa jenis:

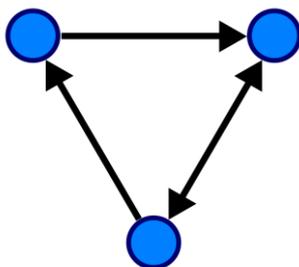
1. Graf sederhana. Graf ini tidak mengandung gelang (*loop*) maupun sisi ganda. Antara dua buah simpul akan memiliki satu sisi saja.



Sumber :

<http://mathworld.wolfram.com/GraphLoop.html>

2. Graf tak sederhana. Graf ini merupakan lawan dari graf sederhana, yaitu graf yang mengandung gelang (*loop*) maupun sisi ganda. Graf ini disebut juga sebagai graf semu.
3. Graf tak berarah. Graf yang sisinya tidak mempunyai orientasi arah.
4. Graf berarah. Graf yang sisinya diberikan orientasi arah. Pada graf ini, tidak ada satupun sisi yang



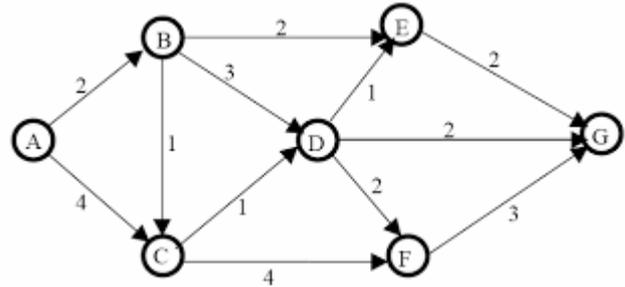
berorientasi dua arah sekaligus.

Graf Berarah

Sumber :

https://en.wikipedia.org/wiki/Directed_graph

5. Graf berbobot. Graf yang memiliki informasi tambahan pada tiap sisinya, biasanya berupa angka, yang merepresentasikan sesuatu biasanya berupa jarak, waktu,dll.

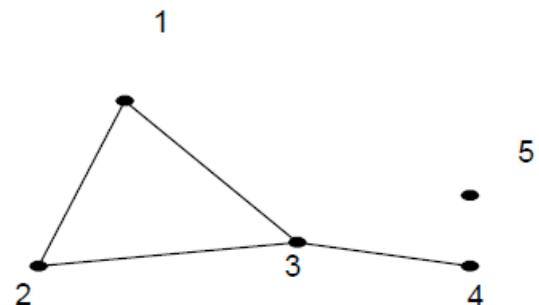


Graf Berbobot. Sumber :

<http://repository.usu.ac.id/bitstream/handle/123456789/26193/Chapter%20II.pdf?sequence=3>.

Disamping variasi jenis graf, terdapat pula beberapa terminologi graf, yaitu:

1. Ketetanggaan (*Adjacent*)
Dua simpul bertetangga jika keduanya terhubung langsung.
2. Bersisian (*Incidency*)
Untuk sembarang sisi $e = (v_1, v_2)$ maka e bersisian dengan simpul v_1 atau e bersisian dengan simpul v_2 .
3. Simpul Terpencil (*Isolated Vertex*)
Simpul yang tidak memiliki sisi yang bersisian dengannya.

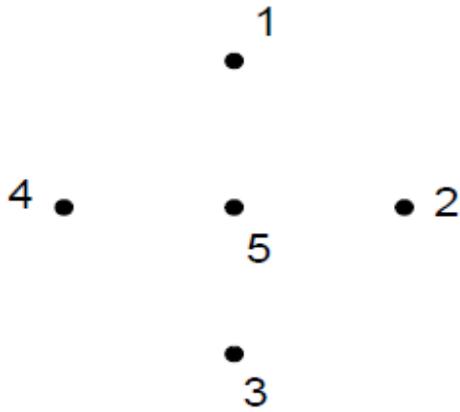


Simpul Terpencil

Sumber :

<http://rabbitjeyek.blogspot.com/2011/12/teori-graf-6.html>

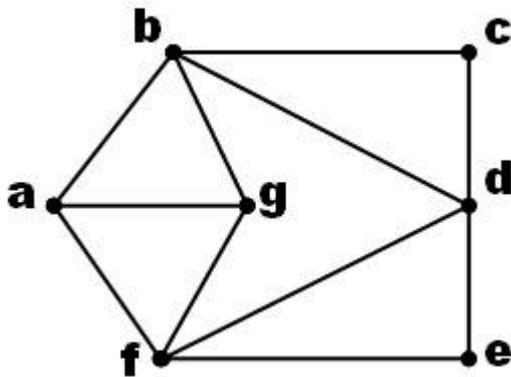
4. Graf Kosong (*Null Graph*)
Graf yang tidak memiliki sisi (himpunan sisinya kosong).



Graf Kosong
Sumber :

<http://rabbitjeyek.blogspot.com/2011/12/teori-graf-6.html>

5. Derajat (*Degree*)
Jumlah sisi yang bersisian dengan sebuah simpul.
Lemma jabat tangan : jumlah derajat semua simpul pada suatu graf adalah genap, yaitu dua kali jumlah sisi pada graf tersebut.



Graf untuk Contoh Kasus Lintasan dan Sirkuit
Sumber :

<http://rabbitjeyek.blogspot.com/2011/12/teori-graf-6.html>

6. Lintasan (*Path*)
Lintasan ialah barisan selang-seling simpul-simpul dan sisi-sisi yang berangkat dari simpul awal v_0 ke simpul tujuan v_n .
Panjang lintasan : jumlah sisi dalam lintasan.
Contoh lintasan pada graf diatas : a-g-b-c-d-e-f
7. Sirkuit (*Circuit*)
Lintasan yang berawal dan berakhir pada simpul yang sama.
Panjang sirkuit : jumlah sisi dalam sirkuit.
Contoh sirkuit pada graf diatas : a-g-b-a
8. Terhubung (*Connected*)
Dua simpul dikatakan terhubung jika terdapat lintasan

yang menghubungkan kedua simpul tersebut.

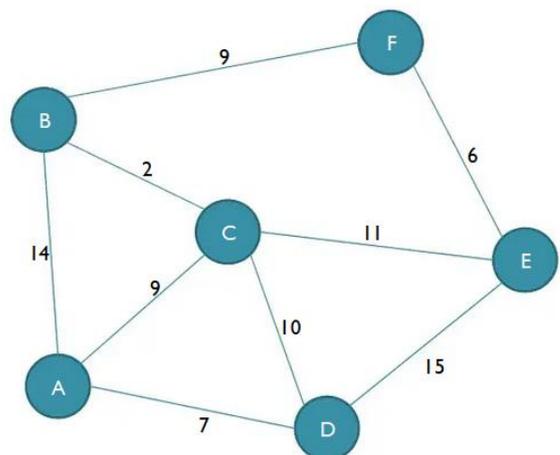
9. Upagraf (*Subgraph*)
Bagian dari sebuah graf dengan simpulnya merupakan himpunan bagian dari simpul graf dan sisinya merupakan himpunan bagian dari sisi graf.
10. *Cut-set*
Himpunan sisi yang bila dibuang dari graf, menyebabkan graf tidak terhubung.

B. Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma yang digunakan untuk memecahkan permasalahan jarak terpendek untuk sebuah graf berarah dan berbobot. Algoritma ini pertama kali dipublikasi pada tahun 1959 dalam jurnal *Numerische Mathematik* yang berjudul "A Note on Two Problems in Connexion with Graphs". Permasalahan jarak terpendek pada graf merupakan sebuah masalah yang dekat kaitannya dengan dunia nyata. Misalnya, jika kita ingin ke kota A dari kota B dan terdapat dua jalan menuju kota A, misal C dan D. Maka, kita akan melalui jalan yang merupakan jarak terpendek untuk ke kota A.

Langkah-langkah algoritma Dijkstra adalah sebagai berikut:

1. Tentukan titik yang akan menjadi node awal.
2. Beri bobot jarak dari node awal ke node terdekat satu per satu.
3. Beri nilai bobot jarak untuk setiap simpul ke simpul lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga pada node yang belum terisi bobotnya.
4. Dari node asal, cari node tetangga yang belum dilalui dan hitung jaraknya dari node asal. Jika ditemui jarak yang lebih kecil, hapus data yang sebelumnya lalu simpan ulang data dengan jarak yang baru ditemukan.
5. Setelah selesai mengoptimasi setiap jarak ke node tetangga, tandai node yang telah dilalui sebagai "node terlewati". Node yang sudah dilewati tidak akan di cek kembali.
6. Set node belum terlewati dengan bobot terkecil (dari node asal) sebagai node asal yang baru dan ulangi langkah 5.



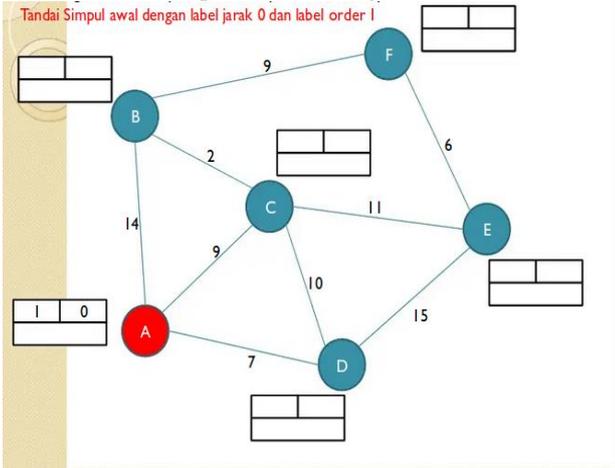
Graf Algoritma Dijkstra

Sumber :

https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Penentuan_Lintasan_Terpendek

Langkah Dijkstra untuk menentukan jarak terpendek dari A ke F:

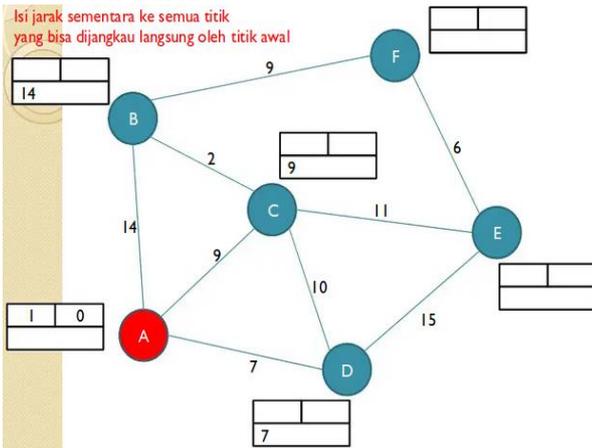
1. Tandai Simpul A memiliki jarak 0 dan label urutan 1



Sumber :

[https://www.academia.edu/14778326/Algoritma Dijkstra Untuk Penentuan Lintasan Terpendek](https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Pentuan_Lintasan_Terpendek)

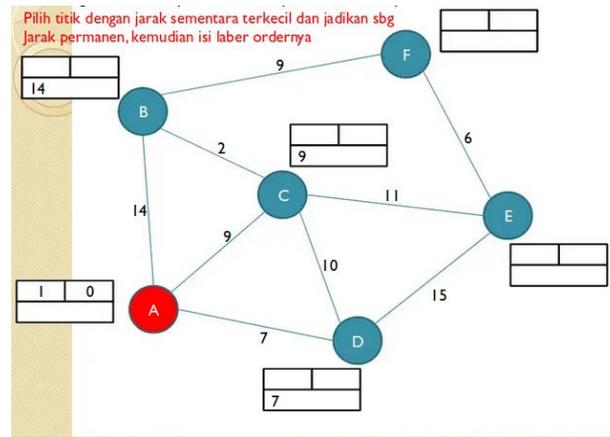
2. Refresh dan isi semua label jarak titik yang terjangkau oleh titik A



Sumber :

[https://www.academia.edu/14778326/Algoritma Dijkstra Untuk Penentuan Lintasan Terpendek](https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Pentuan_Lintasan_Terpendek)

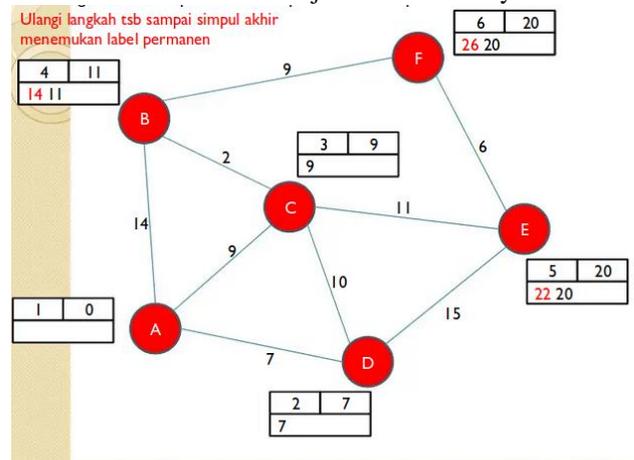
3. Pilih titik dengan jarak sementara terkecil dan jadikan sebagai jarak permanen, kemudian isi label urutannya.



Sumber:

[https://www.academia.edu/14778326/Algoritma Dijkstra Untuk Penentuan Lintasan Terpendek](https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Pentuan_Lintasan_Terpendek)

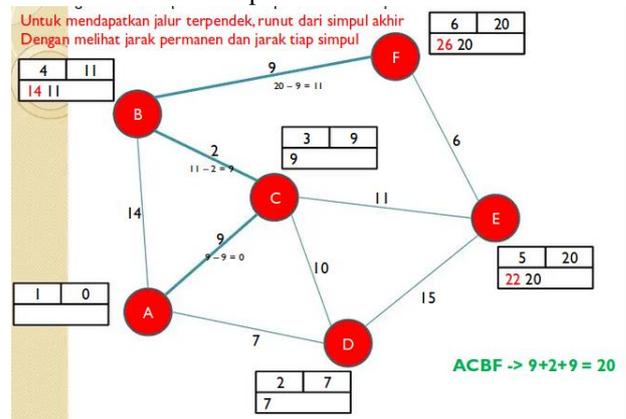
4. Melakukan langkah ketiga berulang kali sampai semua titik telah terisi label jarak dan urutannya



Sumber :

[https://www.academia.edu/14778326/Algoritma Dijkstra Untuk Penentuan Lintasan Terpendek](https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Pentuan_Lintasan_Terpendek)

5. Setelah semua titik memiliki label jarak dan urutan, dioptimisasi jarak masing-masing simpul dengan menelusuri simpul akhir.



Sumber :

[https://www.academia.edu/14778326/Algoritma Dijkstra Untuk Penentuan Lintasan Terpendek](https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Pentuan_Lintasan_Terpendek)

III. ANALISIS DAN PEMBAHASAN

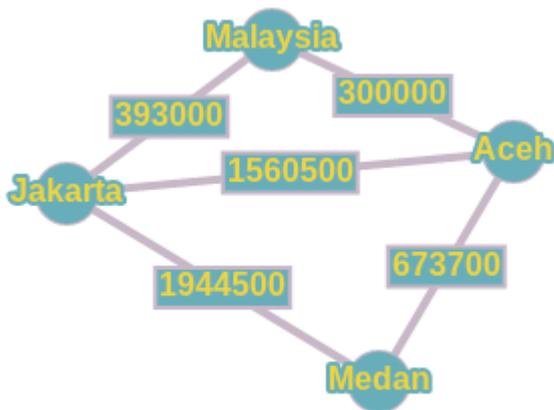
1. Pendefinisian Graf

Untuk dapat mencari harga tiket termurah dari kota A ke kota B maka perlu didefinisikan terlebih dahulu graf keterhubungan antara kota A dengan kota yang memungkinkan untuk menuju B. Simpul = kota, Sisi = adanya jalur penerbangan dari kota asal ke kota tujuan. Setiap sisi diberi bobot berupa harga tiket pesawat dari kota asal ke kota tujuan.

2. Contoh Kasus

Misalnya, kita ingin pergi ke Banda Aceh dari Jakarta. Definisikan terlebih dahulu graf keterhubungan antarkota yang sisi-sisinya memiliki bobot berupa harga tiket. Dalam kasus ini, misalnya graf yang terdefinisi, yaitu: (Semua harga adalah pemisalan)

- (Jakarta,Aceh) → Rp 1.560.500
- (Jakarta,Medan) → Rp 1.944.500
- (Medan,Aceh) → Rp 673.700
- (Jakarta, Malaysia) → Rp 393.000
- (Malaysia,Aceh) → Rp 300.000

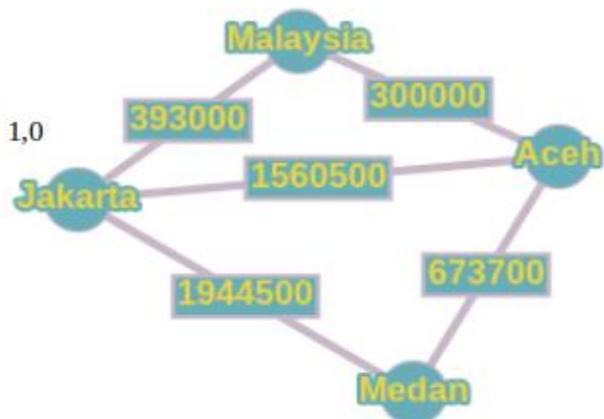


Representasi Graf

Sumber : <https://graphonline.ru/en/>

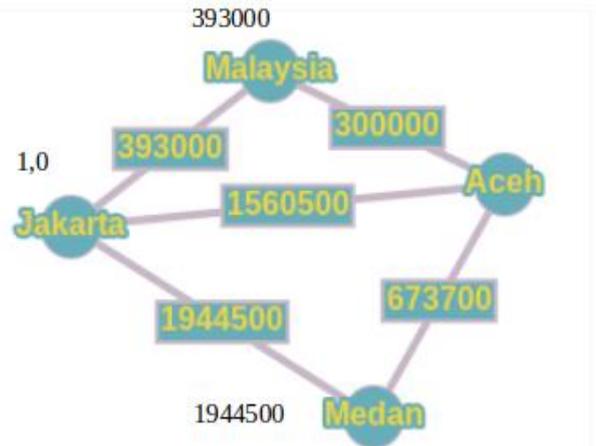
Penerapan Algoritma Dijkstra:

1. Jakarta → Aceh, maka Jakarta di set sebagai simpul awal (memiliki harga 0 dan memiliki urutan 1)

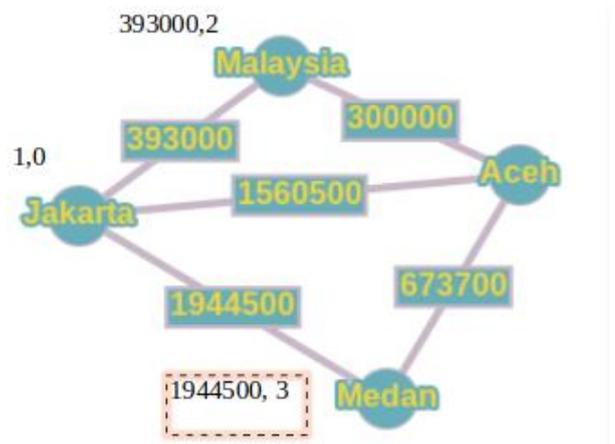


2. Isi jarak sementara dari titik yang bisa dijangkau

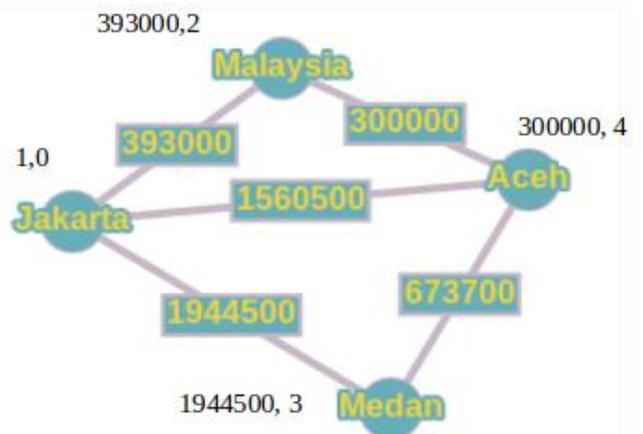
langsung oleh Jakarta



3. Pilih titik dengan harga sementara terkecil dan jadikan sebagai titik permanen, kemudian mengisi label urutannya.



4. Isi harga sementara ke semua titik yang bisa dijangkau langsung oleh titik permanen (Ulangi sampai simpul akhir terisi label urutannya)



5. Optimasi terakhir dari harga simpul yang akan dilewati dari oleh lintasan simpul akhir – simpul awal
Dalam kasus ini, hasilnya sama saja dengan nomor 4.

REFERENCES

- [1] Munir, R., *Graf*.
Dari :
[PDF]. [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf)
- [2] Algoritma Dijkstra. (2017,Nov 28). Diakses pada 6 Desember 2019.
Dari : <https://mti.binus.ac.id/2017/11/28/algoritma-dijkstra/>
- [3] Kurniawan, Imam, Algoritma Dijkstra untuk Penentuan Lintasan Terpendek. Diakses pada 6 Desember 2018.
Dari:
https://www.academia.edu/14778326/Algoritma_Dijkstra_Untuk_Penentuan_Lintasan_Terpendek/.

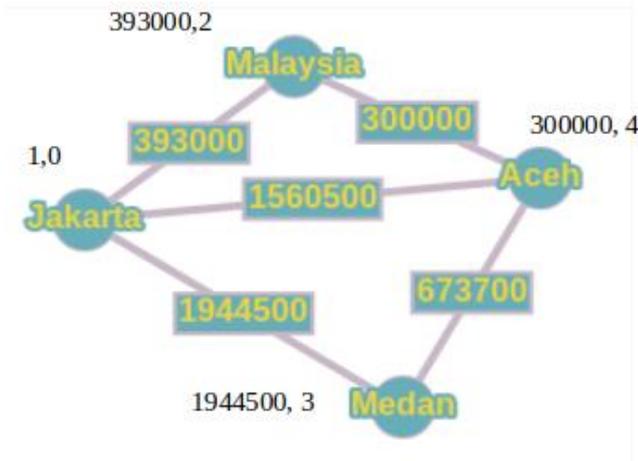
PERNYATAAN

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, 6 Desember 2019



Arief Darmawan Tantriady 13518015



Didapatkan harga termurah yaitu dari Jakarta → Malaysia → Aceh (693000)

IV. KESIMPULAN

Banyak sekali hal-hal di dunia nyata yang bisa diterapkan sebagai representasi graf. Sehingga algoritma-algoritma yang ada pada graf mudah diterapkan ke kegiatan-kegiatan atau wujud-wujud aksi di dunia nyata, sehingga banyak optimasi yang dapat dilakukan akibat adanya graf ini. Algoritma Dijkstra dan algoritma pencarian jarak terpendek lainnya, saya rasa sangat membantu perkembangan berbagai bidang di dunia. Misalnya, terciptanya *Google Maps*. Hasil sekomples *Google Maps* saya rasa tidak akan tercipta kalau tidak adanya pemahaman akan teori graf terlebih dahulu.

Dengan demikian, dapat saya simpulkan mempelajari graf sebagai salah satu topik dalam mata kuliah IF2120 Matematika Diskrit adalah hal yang fundamental dan dapat diterapkan pada pencarian harga tiket pesawat termurah dari kota A ke kota B.

V. UCAPAN TERIMA KASIH

Pada kesempatan ini, saya ingin mengucapkan terima kasih kepada Tuhan yang telah membimbing saya hingga hari ini, dan telah memberi saya kesempatan menduduki bangku kuliah Institut Teknologi Bandung. Saya juga berterima kasih kepada orangtua yang telah mendukung saya dari saya lahir sampai sekarang, serta seluruh dosen pembimbing Matematika Diskrit yang saya hormati, yaitu Bapak Dr. Ir. Rinaldi Munir, MT., Ibu Dra. Harlili M.Sc., dan Ibu Fariska Zakhralativa Ruskanda, S.T., M.T., karena telah dengan sabar menyumbangkan waktunya membimbing kami pada mata kuliah Matematika Diskrit. Terakhir, ucapan terima kasih saya berikan kepada teman-teman saya yang telah melewati susah dan senang bersama, menjadi penyemangat kuliah saya, dan mendukung saya di segala kesempatan.