

Aplikasi Graf untuk menentukan rute yang direkomendasikan Google Maps

Hizbulloh Ash-Shidiqy
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518047@std.stei.itb.ac.id

Abstrak — Sejak zaman dahulu, peta merupakan suatu hal yang penting. Entah untuk menjelajah dunia, untuk mengetahui kawasan musuh dalam kondisi perang, dan lainnya. Di zaman sekarang, peta sangat dibutuhkan untuk mobilisasi modern. Penggunaan peta yang paling sering ditemui saat ini adalah pada ojek-ojek online seperti Grab, Go-Jek dan lainnya. Umumnya peta online yang digunakan oleh ojek online ini adalah Google Maps. Peta ini memberikan bukan hanya berguna untuk mengetahui lokasi tujuan penumpang, tetapi juga untuk memperkirakan prediksi harga yang harus dibayarkan penumpang, berdasarkan jarak perjalanan. Perlu diketahui bahwa cara penentuan rute ini tidak sesederhana mengambil citra dari satelit kemudian diteruskan ke smartphone. Tetapi penentuannya menggunakan pemanfaatan dari suatu cabang ilmu matematik diskrit yaitu graf. Makalah ini membahas mengenai cara kerja penentuan rute yang direkomendasikan oleh Google Maps beserta algoritma yang digunakannya.

Kata kunci — Google Maps, Graf, rute rekomendasi.

I. PENDAHULUAN

Kita sering mendengar zaman sekarang adalah zaman dengan tingkat mobilisasi tertinggi dimana, kita dengan mudah pergi ke suatu Negara, dan keesokan paginya sudah dinegara lainnya. Hal ini bukan hanya semata-mata karena kendaraan-kendaraan yang canggih dengan kecepatan tinggi, namun kita juga harus berterimakasih kepada peta, sehingga kita dapat menempuh rute yang lebih singkat, dapat menghindari kemacetan, dan lain sebagainya.

Tak dapat dipungkiri Google Maps saat ini merupakan salah satu aplikasi peta online yang hampir semua orang miliki dan setidaknya pernah menggunakannya sekali. Apalagi bagi orang-orang yang sering kali memesan ojek online untuk pulang ataupun pergi ke suatu tempat. Tercatat, Google Map sudah 10 juta kali di unduh di platform android. Belum terhitung pada platform Windows, maupun Apple. Pernahkah kita berpikir bagaimana sih cara Google Map memberikan rute rekomendasi yang paling efektif kepada penggunanya? Mungkin sebagian besar dari kita berpikiran bahwa hal itu sesederhana satelit mengirimkan citra gambar ke ponsel pintar pengguna, yang kemudian ponsel kita mengetahui bahwa rute ini mengalami macet parah, rute ini berjarak sekian meter, rute ini melewati banyak lampu merah, dan lain sebagainya.

Kenyataannya tidak semudah itu Google Maps beroperasi. Bayangkan Google Map yang harus menampung

begitu banyak data. Dilansir dari inet.detik.com, Google Maps memiliki lebih dari 20 petabyte data atau setara dengan 21 juta Gigabyte. Selain itu, Google Maps juga tidak hanya menggunakan satu jenis satelit tetapi berbagai satelit masing-masing untuk keperluan berbeda sebelum kemudian digabungkan oleh algoritma dalam Google Maps sehingga menjadi informasi yang berguna. Diantara semua itu tentunya, hal yang paling krusial dan paling bermanfaat bagi kita adalah bagaimana cara Google Maps memebri tahu kita rute terbaik yang mungkin ditempuh untuk mencapai tujuan yang sama. Kecanggihan ini bukan didapat melalui satelit melainkan menggunakan algoritma yang diimplementasikan dari cabang ilmu graf.

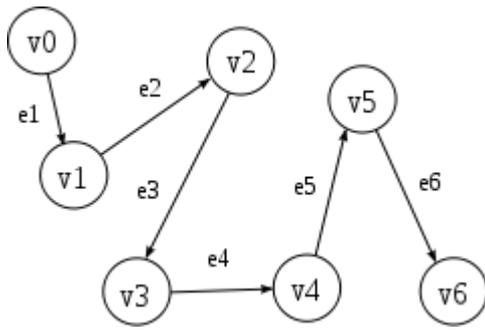
II. DASAR TEORI

A. Definisi Graf

Sejarah teori graf bermula saat ahli matematika Swiss Leonhard Euler memecahkan masalah jembatan Königsberg. Masalah jembatan Königsberg adalah teka-teki lama mengenai kemungkinan menemukan jalan setapak di tujuh jembatan yang membentang di sepanjang sebuah sungai bercabang yang melewati sebuah pulau tanpa melewati jembatan yang sama dua kali. Euler berpendapat bahwa jawabannya adalah tidak ada. Buktinya hanya mengacu pada susunan fisik jembatan, namun intinya dia membuktikan teorema pertama dalam teori graf.

Secara matematis graf didefinisikan sebagai pasangan himpunan simpul dan sisi. Jika graf disimbolkan dengan G , himpunan simpul-simpul yang terdapat dalam graf disimbolkan dengan $V = \{v_1, v_2, v_3, \dots, v_n\}$, dan himpunan sisi yang menghubungkan sepasang simpul dalam graf disimbolkan dengan $E = \{e_1, e_2, e_3, \dots, e_n\}$, maka graf dapat ditulis dengan notasi matematis $G = (V, E)$.

Dengan definisi di atas, kita ketahui bahwa sebuah graf harus memiliki paling tidak sebuah simpul. Namun sebuah graf boleh tidak memiliki sisi, graf yang hanya memiliki simpul dan tidak memiliki sisi disebut graf kosong / null graph. Graf kosong menunjukkan ketiadaan hubungan antar objek-objek di dalamnya.



Gambar 2.1. Contoh graf

B. Jenis Graf

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

- Graf sederhana : graf yang tidak mengandung gelang maupun sisi-ganda.
- Graf tak-sederhana : graf yang mengandung sisi ganda atau gelang.

Berdasarkan banyaknya simpul, graf dapat digolongkan menjadi dua yaitu:

- Graf berhingga : graf dengan banyak simpul terhingga.
- Graf tak-berhingga : graf dengan banyak simpul tak terhingga.

Berdasarkan ada tidaknya arah pada sisi graf, maka graf dapat digolongkan menjadi dua yaitu:

- Graf tak-berarah : graf yang tidak memiliki arah pada sisinya.
- Graf berarah : graf yang pada semua sisinya terdapat arah.

C. Graf Berbobot

Suatu graf dapat dilengkapi dengan bobot pada tiap sisinya untuk memberikan informasi tambahan. Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Misalnya pada kasus graf pembuatan jalan antarkota, sisi-sisi yang menghubungkan tiap kota dapat memiliki bobot harga pembuatan jalan tersebut.

D. Terminologi Graf

Istilah-istilah yang umum digunakan dalam graf adalah sebagai berikut:

- Ketetanggaan : Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.
- Bersisian (Incidency): Untuk sembarang sisi $e = (v_j, v_k)$ dikatakan e bersisian dengan simpul v_j , atau e bersisian dengan simpul v_k
- Simpul Terpencil (Isolated Vertex): Simpul terpencil ialah

simpul yang tidak mempunyai sisi yang bersisian dengannya.

4. Graf Kosong (null graph atau empty graph): Graf yang himpunan sisinya merupakan himpunan kosong (N_n).

5. Derajat (Degree): jumlah sisi yang bersisian dengan simpul tersebut. Notasi: $d(v)$

6. Lintasan (Path): Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ..., $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G .

7. Siklus (Cycle) atau Sirkuit (Circuit): Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.

8. Terhubung (Connected): Dua buah simpul v_1 dan simpul v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2 . G disebut graf terhubung (connected graph) jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j . Jika tidak, maka G disebut graf tak-terhubung (disconnected graph).

- Graf berarah G dikatakan terhubung jika graf tidak berarahnya terhubung (graf tidak berarah dari G diperoleh dengan menghilangkan arahnya).
- Dua simpul, u dan v , pada graf berarah G disebut terhubung kuat (strongly connected) jika terdapat lintasan berarah dari u ke v dan juga lintasan berarah dari v ke u .
- Jika u dan v tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya, maka u dan v dikatakan terhubung lemah (weakly connected). Rinaldi Munir/IF2120 Matematika Diskrit 36
- Graf berarah G disebut graf terhubung kuat (strongly connected graph) apabila untuk setiap pasang simpul sembarang u dan v di G , terhubung kuat. Kalau tidak, G disebut graf terhubung lemah.

9. Upagraf (Subgraph) dan Komplemen Upagraf: Misalkan $G = (V, E)$ adalah sebuah graf. $G_1 = (V_1, E_1)$ adalah upagraf (subgraph) dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. Komplemen dari upagraf G_1 terhadap graf G adalah graf $G_2 = (V_2, E_2)$ sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang anggota-anggota E_2 bersisian dengannya.

10. Upagraf Rentang (Spanning Subgraph) Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan upagraf rentang jika $V_1 = V$ (yaitu G_1 mengandung semua simpul dari G)

11. Cut-Set Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung. Jadi, cut-set selalu menghasilkan dua buah komponen.

D. Lintasan Terpendek (Shortest Path)

Salah satu persoalan dalam graf adalah persoalan lintasan terpendek. Permasalahan mencari lintasan terpendek dalam graf merupakan salah satu persoalan optimasi. Graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot, yaitu graf yang setiap sisinya diberikan suatu nilai

atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya. Terdapat berbagai macam variasi dalam persoalan lintasan terpendek, yaitu:

- a. Lintasan terpendek antara dua simpul tertentu.
- b. Lintasan terpendek antara semua pasangan simpul.
- c. Lintasan terpendek dari satu simpul ke suatu simpul lainnya.
- d. Lintasan terpendek antara dua buah simpul dengan melalui simpul-simpul tertentu.

2. Google Maps

Google didirikan oleh Larry Page dan Sergey Brin saat masih mahasiswa Ph.D. di Universitas Stanford. Mereka berdua memegang 16 persen saham perusahaan. Mereka menjadikan Google sebagai perusahaan swasta pada tanggal 4 September 1998. Pernyataan misinya adalah "mengumpulkan informasi dunia dan membuatnya dapat diakses dan bermanfaat oleh semua orang" dan slogan tidak resminya adalah "Don't be evil". Pada tahun 2006, kantor pusat Google pindah ke Mountain View, California.

Sejak didirikan, pertumbuhan perusahaan yang cepat telah menghasilkan berbagai produk, akuisisi, dan kerja sama di bidang mesin pencari inti Google. Perusahaan ini menawarkan perangkat lunak produktivitas daring (dalam jaringan), termasuk surat elektronik (surel), paket aplikasi perkantoran, dan jejaring sosial. Produk-produk komputer mejanya meliputi aplikasi untuk menjelajah web, mengatur dan menyunting foto, dan pesan instan. Perusahaan ini memprakarsai pengembangan sistem operasi Android untuk telepon genggam dan Google Chrome OS¹ untuk jajaran netbook Chromebook. Google sudah beralih ke perangkat keras komunikasi. Mereka bekerja sama dengan berbagai produsen elektronik besar untuk memproduksi perangkat Nexus-nya dan mengakuisisi Motorola Mobility pada Mei 2012. Tahun 2012, infrastruktur serat optik dipasang di Kansas untuk memfasilitasi layanan Internet pita lebar Google Fiber.

Perusahaan ini diperkirakan mengoperasikan lebih dari satu juta server di beberapa pusat data di seluruh dunia dan memproses lebih dari satu miliar kueri pencarian dan sekitar 24 petabita data buatan pengguna setiap harinya. Pada bulan Desember 2012, Alexa menyebut google.com sebagai situs web paling banyak dikunjungi di dunia. Situs-situs Google dalam bahasa lain masuk peringkat 100 teratas, sebagaimana halnya situs milik Google seperti YouTube dan Blogger. Google menempati peringkat kedua di basis data ekuitas merek BrandZ. Dominasi pasarnya menuai kritik mengenai hak cipta, penyensoran, dan privasi.. Pada tahun 2014, Google juga mendapat penghargaan dari Business Indeed sebagai perusahaan yang memiliki merk paling bernilai.

Sebagaimana yang kita ketahui, Google telah sukses dalam menjadi web pencari terbaik di dunia. Tidak puas dengan hal tersebut, Google saat ini melebarkan sayapnya ke berbagai bidang. Contohnya bidang periklanan seperti Google AdSense, bidang komunikasi seperti Google Drive dan Google Docs, bidang hiburan seperti Google video dan Youtube dan bidang

lainnya. Salah satu hasil pengembangan tersebut adalah Google Maps.

Google Maps adalah layanan pemetaan web yang dikembangkan yang memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk bepergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum. Pada mulanya, Google Maps dimulai sebagai program desktop C++, dirancang oleh Lars dan Jens Eilstrup Rasmussen pada Where 2 Technologies. Pada Oktober 2004, perusahaan ini diakuisisi oleh Google, yang diubah menjadi sebuah aplikasi web. Setelah akuisisi tambahan dari perusahaan visualisasi data geospasial dan analisis lalu lintas, Google Maps diluncurkan pada Februari 2005. Layanan ini menggunakan Javascript, XML, dan AJAX. Google Maps menawarkan API yang memungkinkan peta untuk dimasukkan pada situs web pihak ketiga, dan menawarkan penunjuk lokasi untuk bisnis perkotaan dan organisasi lainnya di berbagai negara di seluruh dunia. Pada Google Maps API terdapat 4 jenis pilihan model yang disediakan oleh Google, diantaranya adalah :

1. Roadmap, untuk menampilkan peta biasa 2 dimensi
2. Satellite, untuk menampilkan foto satelit
3. Terrain, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
4. Hybrid, akan menunjukkan foto satelit yang diatanya tergambar pula apa yang tampil pada roadmap (jalan dan nama kota)

Google Map Maker memungkinkan pengguna untuk bersama-sama mengembangkan dan memperbarui pemetaan layanan di seluruh dunia.

Tampilan satelit Google Maps adalah "top-down". Sebagian besar citra resolusi tinggi dari kota adalah foto udara yang diambil dari pesawat pada ketinggian 800 sampai 1.500 kaki (240–460 meter), sementara sebagian besar citra lainnya adalah dari satelit. Sebagian besar citra satelit yang tersedia adalah tidak lebih dari berusia tiga tahun dan diperbarui secara teratur. Google Maps menggunakan varian dekat dari proyeksi Mercator, dan karena itu Google Maps tidak dapat secara akurat menunjukkan daerah di sekitar kutub.

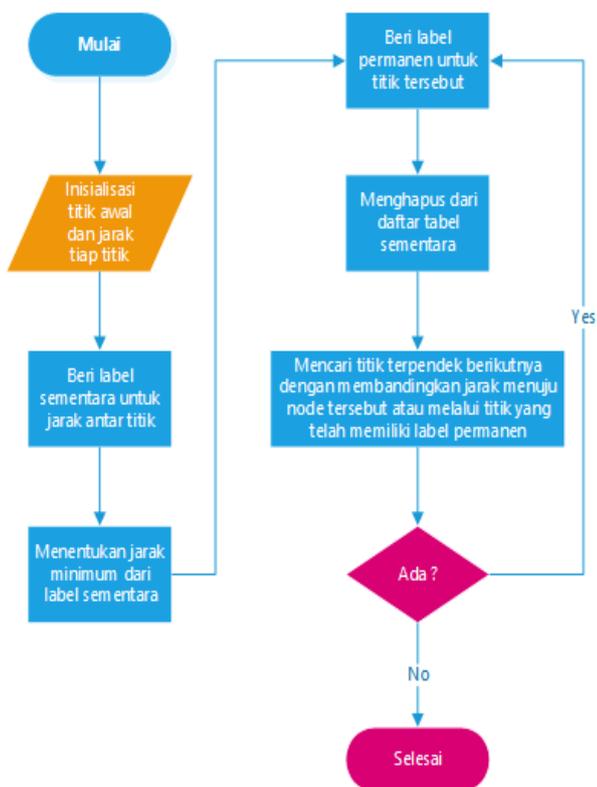
Google Maps untuk seluler dirilis pada bulan September 2008. Pada Agustus 2013, Google Maps bertekad untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% dari pemilik ponsel cerdas di seluruh dunia menggunakannya setidaknya sekali.

III. IMPLEMENTASI GRAF DALAM PENENTUAN RUTE REKOMENDASI GOOGLE MAPS

Dalam implementasinya, Google Maps menggunakan Algoritma Dijkstra. Algoritma ini ditemukan oleh seorang ilmuwan komputer bernama Edsger W. Dijkstra pada tahun 1956 dan diterbitkan tiga tahun kemudian pada sebuah jurnal

Numerische Mathematik yang berjudul “A Note on Two Problems in Connexion with Graphs”. Algoritma ini sering digambarkan sebagai algoritma greedy (tamak). Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infinity (Tak Hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute lintasan terpendek.

Berikut Pseudo Code dan Flowchart Algoritma Dijkstra:



Gambar 3.1. Flowchart Pseudocode untuk algoritma Dijkstra (sumber: <https://wirasetiawan.blog/>)

Cara kerja algoritma Dijkstra adalah sebagai berikut:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum Terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari no keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai

“Node terjamah”. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3

Algoritma Dijkstra secara umum dituliskan sebagai berikut:

```
function Dijkstra(Graph, source):
    for each vertex v in Graph:
        dist[v] := infinity ;

        previous[v] := undefined ;
    end for

    dist[source] := 0 ;
    Q := the set of all nodes in Graph ;

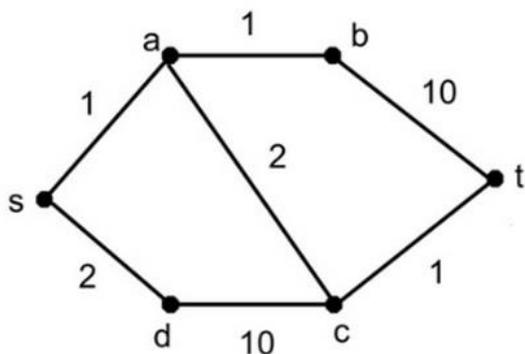
    while Q is not empty:
        u := vertex in Q with smallest distance in dist[] ;
        remove u from Q ;
        if dist[u] = infinity:
            break ;
        end if

        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;
```

Gambar 3.2. Algoritma Dijkstra (sumber: <https://wirasetiawan .blog/>)

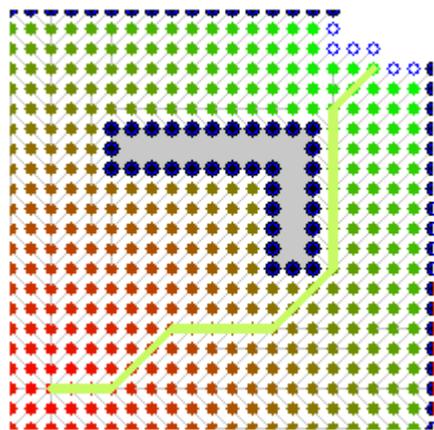


Gambar 3.3. Graf untuk Penjelasan cara kerja algoritma Dijkstra
(sumber: <https://www.vice.com/>)

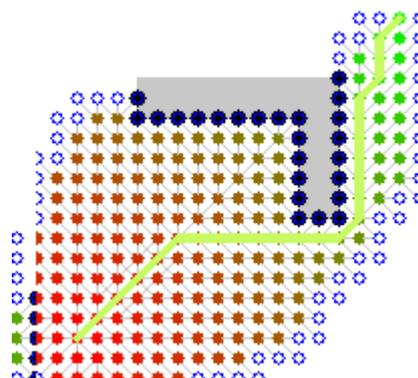
Sebagai contoh adalah graf pada gambar 3.3, dimana nomor yang diberikan di sisi nya adalah bobot dari masing-masing keterhubungan. Bobot bisa jadi dianggap sesederhana jarak, tapi sebagaimana yang digunakan dalam Google Maps, bobot ini bisa jadi bukan sekadar jarak tetapi meliputi beberapa factor lainnya seperti tingkat kemacetan, banyak lampu lalu lintas yang dilalui, dan factor apapun yang menurut kita berhubungan dan ingin dibuat seminimal-minimalnya. Untuk memulai, kita menulis s sebagai posisi awal kita, bobotnya 0. Karena untuk mencapainya kita hanya memerlukan 0 meter, atau melewati 0 lampu lalu lintas, atau apapun factor lainnya. Selanjutnya, kita melihat pada simpul-simpul yang bertetangga dengan simpul S. Untuk langkah pertama, kita mencari simpul tetangga terdekat, atau bobot terendah, yang berdasarkan gambar adalah simpul a, 1 bobot. Selanjutnya kita kembali mencari simpul tetangga terdekat dari posisi awal kita, node s, dari simpul a maupun simpul s yang belum dilewati. simpul b berbobot 1 (berbobot 2 dari posisi awal) dan simpul c berbobot 2 (berbobot 3 dari posisi awal), kita juga memiliki simpul d yang berjarak 2 dari posisi awal. Karena kita mencari simpul dengan jarak terdekat dengan posisi awal, maka kita memilih d. Langkah selanjutnya hanyalah mengulangi langkah sebelumnya, dari d kita melihat c, yang berjarak 10 bobot, (12 bobot dari s), tapi jangan melupakan simpul a, yang kita dapat menuju simpul c dengan bobot 2 (3 dari simpul s) and b dengan 1 bobot (2 dari posisi awal). Maka tujuan kita selanjutnya adalah pada simpul b. Penjelajah yang ditempatkan di b berada dalam kekecewaan. Satu-satunya kemungkinan pindah ke t adalah 10 unit jauhnya (12 dari awal). Dan ini lebih dari 2 unit dari a ke c (3 dari awal) dan sama dengan perjalanan dari s ke c hingga d, kemungkinan kita sekarang dapat dengan aman membuang (setelah sampai di c hanya dalam 3 unit, daripada 12 diperlukan melalui d). Sekarang, kita berada di c dan jika ini tampak rumit itu benar-benar tidak. Kami hanya membuat langkah-langkah tentatif berhati-hati dari node ke node, sementara dipaksa oleh algoritma untuk selalu mempertimbangkan jalur terpendek dari awal. Akhirnya, kita kembali melihat dari b ke t, lagi-lagi mencatat total path sebagai 12. Sementara itu, lompatan terakhir dari c biaya 1, untuk total total jarak jalur terpendek dari 4. Walaupun nampaknya rumit, persoalan ini dapat diselesaikan hitungan detik jika dikerjakan diatas kertas, bahkan lebih sedikit lagi

waktu yang diperlukan jika menggunakan computer sebagaimana yang dilakukan Google Maps. Inilah yang membuat Google Maps dapat mencari seluruh variasi, atau setidaknya beberapa variasi. Inilah yang memungkinkan pencarian rute seperti itu.

Namun, tentu saja Google Maps tidak hanya menggunakan algoritma Dijkstra murni dalam skala Global, karena bukan hanya akan menggunakan banyak waktu karena mengecek segala kemungkinan tapi juga, algoritma Dijkstra cenderung menghabiskan sangat banyak memori ketika dijalankan. Biasanya algoritma Dijkstra digunakan untuk keperluan umum dan skala yang tidak begitu besar serta tidak digunakan untuk waktu yang efektif. Sebagai contoh, karena Google Maps digunakan untuk mengkomputasi rute dalam bidang Euclidian dua dimensi, maka dalam hal kecepatan, jelas algoritma Dijkstra kalah jauh dengan algoritma pencarian A*. Sederhananya, cara kerja algoritma pencarian A* adalah menarik garis lurus dari posisi awal menuju posisi yang hendak dituju, jika ada halangan barulah ia akan mencari jalan lewat samping dan lain sebagainya, tidak seperti algoritma Dijkstra yang mencoba segala cara.



Gambar 3.4. Ilustrasi penggunaan algoritma Dijkstra
(sumber: <https://www.quora.com/>)



Gambar 3.5. Ilustrasi penggunaan algoritma pencarian A*
(sumber: <https://www.quora.com/>)

Dapat dilihat dengan jelas dari gambar diatas, bahwa dalam hal waktu algoritma pencarian A* jauh lebih cepat dibandingkan algoritma Dijkstra. Sehingga penulis yakin bahwa Google Maps tidak menggunakan algoritma Dijkstra secara murni, melainkan melalui beberapa tahap sehingga pencarian rute yang direkomendasikan lebih efisien baik secara waktu, memori maupun yang lainnya.

IV. KESIMPULAN

Google Maps memanfaatkan cabang ilmu matematika diskrit yaitu graf untuk menentukan rute terbaik yang mungkin ditempuh oleh pengguna. Pengambilan keputusan ini tidak hanya berdasarkan jarak, tetapi Google Maps juga memasukkan factor lainnya dalam pengambilan keputusan seperti banyak lampu lalu lintas yang dilalui, tingkat kemacetan, dan lain sebagainya. Walaupun Google memang merupakan perusahaan raksasa saat ini, hal itu tidak dapat menjamin bahwa Google Maps sudah menggunakan algoritma terefektif dalam penentuan rute yang direkomendasikan. Bahkan menurut Andrew Goldberg, peneliti utama di Microsoft Research Silicon Valley, mengatakan ada banyak alasan mengapa peneliti terus mempelajari masalah pencarian jalan terpendek. "Jalan terpendek adalah masalah optimasi yang relevan untuk berbagai macam aplikasi, seperti jaringan routing, game, desain sirkuit, dan pemetaan," – Goldberg.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Allah SWT karena hanya berkat rahmat-Nya lah penulis berhasil merampungkan makalah yang berjudul "Aplikasi Graf untuk Menentukan Rute yang direkomendasikan Google Maps" ini. Selain itu penulis juga ingin mengucapkan terima kasih kepada dosen Matematika Diskrit; Dr. Ir. Rinaldi Munir, MT., Dra. Harlili M.Sc., dan Dr. Judhi Santoso, M.Sc. atas ilmu dan waktu yang telah diberikan kepada penulis sehingga penulis mampu merampungkan makalah ini tepat waktu.

REFERENCES

- [1] Rinaldi Munir, Matematika Diskrit, Bandung : Penerbit Informatika, Palasari.
- [2] <https://wirasetiawan.blog/2015/04/02/tentang-algoritma-dijkstra/> Diakses 05 Desember 2019
- [3] <http://www.ms.unimelb.edu.au/~moshe/620-261/dijkstra/dijkstra.html> Diakses 05 Desember 2019
- [4] <http://algs4.cs.princeton.edu/44sp/> Diakses 05 Desember 2019
- [5] http://interactivepython.org/RkmcZ/courselib/static/python_d/Graphs/graphshortpath.html Diakses 05 Desember 2019
- [6] https://www.vice.com/en_us/article/4x3pp9/the-simple-elegant-algorithm-that-makes-google-maps-possible Diakses 06 Desember 2019
- [7] <https://www.quora.com/Does-Google-Maps-use-Dijkstras-algorithm-If-so-can-you-explain-how-they-discretize-the-world> Diakses 06 Desember 2019

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



Hizbulloh Ash-Shidiqy
13518047