

Penerapan Algoritma Floyd Warshall untuk Menentukan Rute *Pick-up* dalam Layanan *Same Day Delivery*

Andres Jerriel Sinabutar (13519218)¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13519218@std.itb.ac.id

Abstrak— Algoritma Floyd Warshall adalah salah satu varian dari pemrograman dinamis, metode untuk memecahkan masalah pencarian rute terpendek. Metode ini melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Maksudnya, solusi-solusi dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu.

Layanan *same day delivery* adalah layanan pengantaran barang dengan batas waktu satu hari setelah pemesanan dibuat oleh pengguna. Dalam layanan ini, pengemudi akan menjalankan beberapa perjalanan dalam satu hari, dimana satu perjalanan akan berisi lebih dari satu destinasi pengambilan dan pengantaran.

Algoritma Floyd Warshall ini dapat dimanfaatkan untuk menentukan rute *pick-up* dalam layanan *same day delivery* sehingga pengemudi bisa menemukan rute terpendek dan proses pengantaran barang lebih cepat, hemat biaya, dan hemat tenaga.

Kata kunci—Lintasan, *Same Day Delivery*, Rute terpendek, Algoritma Floyd Warshall.

I. PENDAHULUAN

Transportasi merupakan salah satu sarana perpindahan manusia atau barang yang sangat penting dalam segala aktivitas manusia. Seiring berjalannya waktu, transportasi berkembang menjadi semakin mudah dan cepat.

Beberapa tahun terakhir, Indonesia mengalami transformasi dalam hal transportasi. Perkembangan teknologi yang semakin modern telah merambah dunia transportasi di Indonesia. Hal ini terlihat dari bermunculannya model transportasi berbasis daring pada kota-kota besar di Indonesia.



Gambar 1. Ilustrasi Aplikasi Layanan Transportasi Daring

Sumber: <https://www.shopback.co.id/katashopback/tips-berkendara-menggunakan-transportasi-online-dengan-aman>

Jasa transportasi berbasis daring ini disebut juga dengan aplikasi *ridesharing* yang kemunculannya di Indonesia mulai marak pada tahun 2014. Awal kemunculannya dimulai oleh aplikasi Uber yang mengusung UberTaxi sebagai bisnis layanan transportasi berbasis aplikasi online. Kemudian, perkembangan layanan transportasi daring ini diikuti dengan kemunculan Gojek, GrabBike, GrabTaxi, dan aplikasi berbasis daring lainnya.

Layanan transportasi daring menawarkan kemudahan dan proses yang cepat bagi penggunanya. Pengguna dapat memesan ojek untuk bepergian dan mengantar barang ke mana saja di dalam kota. Salah satunya ada layanan *Same Day Delivery* yang memudahkan pengemudi untuk mengambil barang pengiriman dari beberapa pengguna dengan tujuan pengantaran yang berdekatan sekaligus. Layanan ini mempercepat proses pengantaran barang dan mengurangi biaya pengantaran karena lokasi tiap tujuan pengantaran barang tidak berjauhan sehingga pengemudi bisa mengantarkan barang dengan lebih efisien dan tidak memerlukan banyak waktu di jalan.

Makalah ini akan membahas bagaimana cara penerapan algoritma *Floyd Warshall* untuk menyelesaikan masalah rute pengambilan barang dari tiap lokasi pemesan supaya lebih efisien dan hemat waktu, serta hemat biaya.

II. LANDASAN TEORI

A. Graf

Graf adalah suatu struktur yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek – objek tersebut. Pada umumnya, objek-objek direpresentasikan sebagai titik, simpul, noktah, bulatan, atau bentuk lainnya. Hubungan antara objek-objek tersebut direpresentasikan dengan suatu garis.

Graf G didefinisikan sebagai pasangan himpunan (V, E) . V merupakan himpunan tidak kosong simpul-simpul (*vertices*) dan E merupakan himpunan sisi-sisi (*edges*). Hal tersebut

dinotasikan dengan:

$$G = (V, E)$$

$$V = \{v_1, v_2, \dots, v_n\}$$

$$E = \{e_1, e_2, \dots, e_n\}$$

Dari definisi di atas, V tidak boleh kosong, sedangkan E boleh kosong. Graf yang hanya mempunyai sebuah simpul tanpa satu sisi pun dinamakan graf trivial. Simpul pada graf biasanya dinamai dengan huruf (a, b, c, \dots), angka ($1, 2, 3, \dots$), ataupun keduanya (a_1, a_2, a_3, \dots). Sisi e yang menghubungkan simpul u dengan simpul v dinotasikan dengan:

$$e = (u, v)$$

Graf dapat dibedakan menjadi beberapa jenis berdasarkan ada tidaknya sisi ganda atau gelang dan orientasi arah pada sisi.

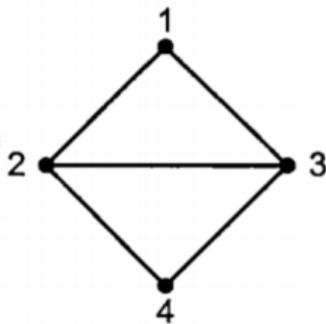
Berdasarkan ada tidaknya sisi ganda atau gelang, graf dibedakan menjadi dua jenis, yaitu sebagai berikut:

1. Graf sederhana (*simple graph*)

Graf sederhana tidak mengandung sisi ganda ataupun gelang. Pada graf sederhana, sisi merupakan pasangan tidak terurut (*unordered pairs*) sehingga (u,v) memiliki arti yang sama dengan (v, u) , dengan notasi:

$$e = (u, v)$$

$$u \neq v$$



Gambar 2. Graf sederhana

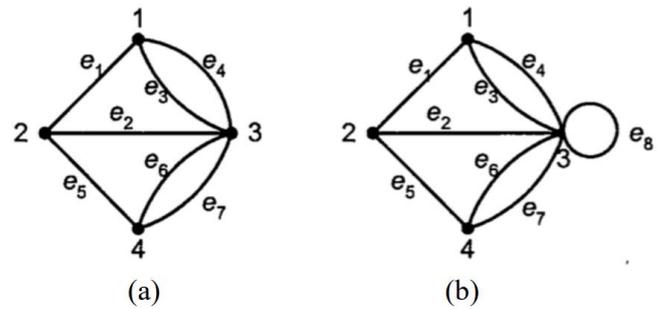
Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

2. Graf tidak sederhana (*unsimple graph*)

Graf tidak sederhana mengandung sisi ganda atau gelang. Graf tidak sederhana dibagi menjadi dua jenis, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Graf semu adalah graf yang mengandung gelang. Gelang merupakan sisi yang menghubungkan simpul dengan simpul itu sendiri, dengan notasi:

$$e = (u, v)$$

$$u = v$$



Gambar 3. (a) Graf ganda dan (b) graf semu dengan gelang pada e_8

Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

Berdasarkan orientasi arah pada sisi, graf dibedakan menjadi dua jenis yaitu sebagai berikut.

1. Graf tidak berarah (*undirected graph*)

Graf tidak berarah adalah graf yang sisinya tidak mengandung arah. Urutan pasangan simpul yang dihubungkan suatu sisi tidak diperhatikan, sehingga:

$$(u, v) = (v, u)$$

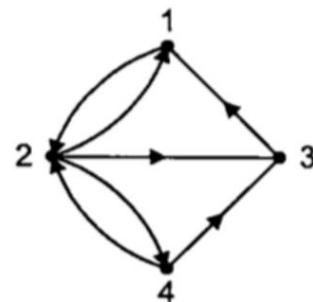
Contoh graf tidak berarah terdapat pada Gambar 2, Gambar 3 (a), dan Gambar 3 (b).

2. Graf berarah (*directed graph* atau *digraph*)

Graf berarah adalah graf yang sisinya memiliki orientasi arah. Orientasi arah umumnya direpresentasikan oleh tanda panah. Sisi berarah disebut busur (*arc*). Pada graf berarah, urutan pasangan simpul diperhatikan, sehingga:

$$(u, v) \neq (v, u)$$

Pada busur (u, v) , u adalah simpul asal (*initial vertex*), sedangkan v adalah simpul terminal (*terminal vertex*).



Gambar 4. Graf berarah

Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

Graf memiliki beberapa terminologi (istilah) dasar. Beberapa istilah dasar yang berkaitan dengan graf adalah:

1. Ketetanggaan (*Adjacency*)

Ketetanggaan dalam graf didefinisikan sebagai dua buah simpul pada graf tak berarah G yang mana keduanya terhubung langsung dengan sebuah sisi. Hal ini dilambangkan dengan (u,v) yang berarti simpul u bertetangga dengan simpul v .

2. Beririsan (*Incidency*)

Beririsan dalam graf didefinisikan sebagai sembarang sisi $e = (u,v)$, menandakan e beririsan dengan simpul u dan simpul v .

3. Simpul Terpencil (*Isolated Vertex*)

Simpul terkecil dalam graf didefinisikan sebagai simpul yang tidak mempunyai sisi yang beririsan dengannya atau simpul yang tidak satupun bertetangga dengan simpul-simpul lainnya.

4. Graf Kosong (*Null Graph* atau *Empty Graph*)

Graf Kosong didefinisikan sebagai graf yang himpunan sisinya merupakan himpunan kosong.



Gambar 5. Graf kosong

Sumber: Munir, Rinaldi. 2016. *Matematika Diskrit. Edisi Revisi Keenam*. Bandung: Informatika Bandung

5. Derajat (*Degree*)

Derajat adalah jumlah sisi yang beririsan dengan simpul pada graf tidak berarah. Pada gambar 2, simpul 1 dan 4 berderajat 2, sedangkan simpul 2 dan 3 berderajat 3, dengan notasi :

$$d(1) = d(4) = 2$$
$$d(2) = d(3) = 3$$

Derajat simpul terpencil adalah 0. Simpul dengan derajat 1 disebut simpul anting-anting (*pendant vertex*).

6. Lintasan (*Path*)

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n pada graf G adalah barisan berselangseling simpul, dengan notasi $v_0, e_0, v_1, e_1, v_2, e_2, \dots, v_{n-1}, e_{n-1}, v_n, e_n$ dengan sisi $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, $e_n = (v_{n-1}, v_n)$ himpunan sisi $E = \{e_1, e_2, \dots, e_n\}$. Lintasan juga dapat disebut jalur. Panjang lintasan adalah jumlah sisi dalam lintasan tersebut. Lintasan sederhana (*simple path*) adalah lintasan dengan semua simpulnya berbeda. Lintasan tertutup (*closed path*) adalah lintasan yang berawal dan berakhir pada simpul yang sama, sedangkan lintasan terbuka (*opened path*) adalah lintasan yang berawal dan berakhir pada simpul yang berbeda. Pada Gambar 2, lintasan 1, 2, 3, 4 adalah lintasan dengan barisan sisi $(1, 2)$, $(2, 3)$, dan $(3, 4)$ dengan panjang lintasan sebesar 3.

7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Siklus atau sirkuit termasuk lintasan tertutup. Panjang sirkuit adalah jumlah sisi pada sirkuit tersebut.

Pada Gambar 2, lintasan 1, 2, 3, 1 merupakan sebuah siklus atau sirkuit dengan panjang 3. Lintasan tersebut berawal dan berakhir pada simpul 1.

8. Terhubung (*Connected*)

Dua buah simpul pada dikatakan terhubung jika terdapat lintasan dari simpul awal ke simpul akhir. Pada graf tidak berarah, graf disebut graf terhubung (*connected graph*) jika untuk setiap pasang simpul u dan v pada himpunan V , terdapat lintasan dari u ke v , sedangkan graf disebut graf tidak terhubung (*disconnected graph*) jika untuk setiap pasang simpul u dan v pada himpunan V , tidak terdapat lintasan dari u ke v . Pada graf berarah, graf disebut graf terhubung jika graf tidak berarahnya terhubung. Pada graf berarah, graf terhubung dibedakan menjadi dua jenis, yaitu graf terhubung kuat (*strongly connected*) dan graf lemah (*weakly connected*). Graf terhubung kuat adalah graf yang memiliki lintasan dari u ke v dan juga lintasan dari v ke u . Graf terhubung lemah adalah graf yang memiliki lintasan dari u ke v , tetapi tidak memiliki lintasan dari v ke u .

9. Upagraf (*Subgraph*) dan Komplemen Upagraf

Sebuah graf $G_1 = (V_1, E_1)$ dikatakan sebagai upagraf dari graf $G = (V, E)$ jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. Sebuah graf $G_2 = (V_2, E_2)$ dikatakan komplemen dari upagraf G terhadap G_1 jika $E_2 = E - E_1$ dan V_2 merupakan himpunan simpul yang anggotanya bersisian dengannya. Komponen graf (*connected component*) adalah jumlah maksimum upagraf terhubung dalam graf tersebut. Pada graf berarah, komponen terhubung kuat (*strongly connected component*) adalah jumlah maksimum upagraf yang terhubung kuat. Graf pada Gambar 3 (a) merupakan upagraf dari graf pada Gambar 3 (b).

10. Upagraf Rentang (*Spanning Subgraph*)

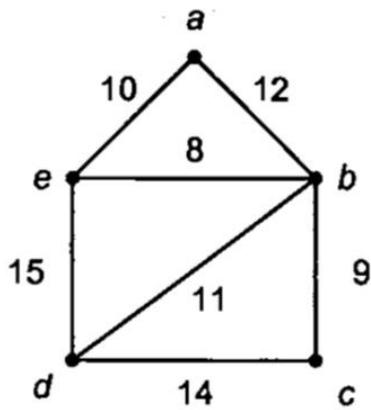
Upagraf $G_1 = (V_1, E_1)$ dari graf $G = (V, E)$ dikatakan upagraf merentang jika $V_1 = V$. Hal tersebut berarti G_1 mengandung semua simpul dari G .

11. Cut-set

Cut-set dari graf terhubung G adalah himpunan sisi yang jika dihilangkan dari G menyebabkan G tidak terhubung sehingga cut-set dua buah komponen. Istilah lain dari cut-set adalah jembatan (*bridge*).

12. Graf Berbobot (*Weighted Graph*)

Graf berbobot adalah graf yang pada setiap sisinya diberi sebuah bobot. Graf berbobot memiliki istilah lain, yaitu graf berlabel.



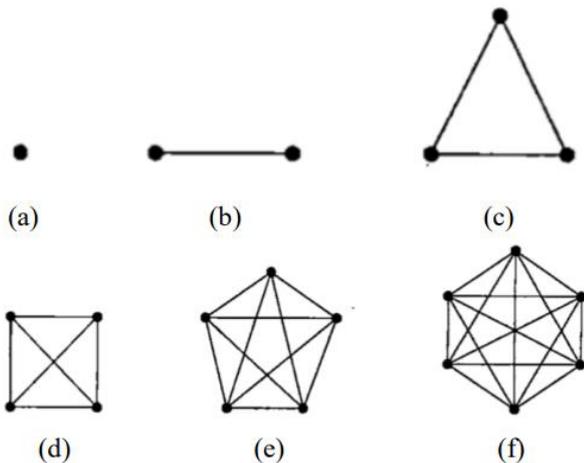
Gambar 6. Graf berbobot

Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

Graf-graf yang memiliki bentuk atau pola khusus disebut graf khusus. Graf khusus ini banyak ditemukan penerapannya. Graf-graf khusus tersebut antara lain sebagai berikut.

1. Graf Lengkap (Complete Graph)

Graf lengkap adalah graf sederhana yang setiap simpulnya memiliki sisi ke semua simpul lainnya. Artinya, setiap simpul pada graf lengkap bertetangga dengan simpul lainnya. Graf lengkap dengan n buah simpul dinotasikan dengan K_n . Derajat setiap simpul pada K_n adalah $n - 1$.

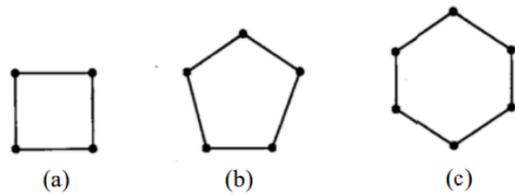


Gambar 7. Graf lengkap (a) K_1 (graf trivial), (b) K_2 , (c) K_3 , (d) K_4 , (e) K_5 , dan (f) K_6

Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

2. Graf Lingkaran

Graf lingkaran adalah graf sederhana yang derajat setiap simpulnya berjumlah dua. Graf lingkaran dengan n buah simpul dinotasikan dengan C_n . Pada Gambar 6 (c), graf K_3 termasuk graf lingkaran.



Gambar 8. Graf lingkaran (a) C_4 , (b) C_5 , dan (c) C_6

Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

3. Graf Teratur (Regular Graph)

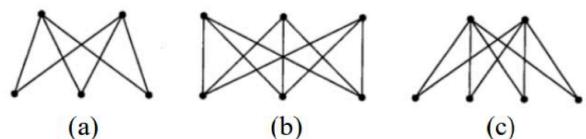
Graf teratur adalah graf yang setiap simpulnya berderajat sama. Graf teratur derajat r adalah graf yang setiap simpulnya berderajat r . Graf lengkap dan graf lingkaran termasuk graf teratur. Jumlah sisi e pada graf teratur dengan n buah simpul dirumuskan dengan:

$$e = \frac{nr}{2}$$

4. Graf Bipartit (Biparte Graph)

Graf bipartit adalah graf yang himpunan simpulnya dapat dikelompokkan menjadi dua himpunan bagian. Graf bipartit G dengan himpunan bagian simpul V_1 dan V_2 dinotasikan dengan $G(V_1, V_2)$. Setiap pasang simpul di himpunan bagian tidak bertetangga dengan simpul lain di himpunan bagian tersebut. Graf bipartit lengkap (complete bipartite graph) adalah graf bipartit yang setiap simpul di himpunan bagian satu bertetangga dengan semua simpul di himpunan bagian lainnya. Graf bipartite lengkap dinotasikan dengan $K_{m,n}$ dengan m dan n jumlah simpul di masing-masing himpunan bagian. Jumlah sisi e pada graf bipartit lengkap dirumuskan dengan:

$$e = mn$$



Gambar 9. Graf bipartit (a) $K_{2,3}$, (b) $K_{3,3}$, dan (c) $K_{2,4}$

Sumber: Munir, Rinaldi. 2016. Matematika Diskrit. Edisi Revisi Keenam. Bandung: Informatika Bandung

B. Location Based Services (LBS)

Qusay H. Mahmoud menyatakan bahwa Location Based Service (LBS) merupakan layanan yang berguna untuk mengetahui posisi dari pengguna, dimana informasi tersebut digunakan untuk menyediakan jasa dan aplikasi pribadi. Terdapat 2 pendekatan dasar yang digunakan untuk pengaplikasian LBS, yaitu:

1. memproses data lokasi di server dan mengirimkan hasilnya ke alat.
2. Memperoleh data lokasi dari alat tersebut menurut aplikasi yang menggunakannya secara langsung

Beberapa pendekatan yang digunakan aplikasi dalam mendapatkan posisi dari pengguna, antara lain :

1. Memanfaatkan jaringan telepon seluler, dimana tingkat ketepatan aplikasi dalam mendeteksi posisi berdasarkan ukuran dari sel dan memungkinkan tingkat akurasi yang lebih rendah.
2. Memanfaatkan satelit, dengan mengaplikasikan GPS dalam mendapatkan informasi posisi yang akurat.

C. Algoritma Floyd Warshall

Algoritma *Floyd Warshall* merupakan pemrograman dinamis yang melakukan penyelesaian masalah dengan melihat solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Prinsip optimalitas ialah prinsip yang diterapkan oleh pemrograman dinamis, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke- i) juga optimal.

Algoritma *Floyd Warshall* memiliki input graf berarah dan berbobot. Serta output dari algoritma Floyd Warshall adalah dengan menghitung bobot terkecil dari semua lintasan yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik.

Berikut ini digunakan formulasi rekursif untuk algoritma *Floyd Warshall* antara lain :

1. Simpul-simpul antara dalam *short path*
2. Jika $V = \{1,2,3,\dots,N\}$, untuk $k = 0, \dots, n$. Maka $d_{ij}^{(k)} =$
 - a. w_{ij} jika $k = 0$
 - b. $\min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$, untuk $k > 0$
3. Solusi dari $d_{ij}^{(n)}$ merupakan matriks *shortest path* dari simpul i ke simpul j .

Berikut ini *pseudocode* dari algoritma Floyd Warshall.

```

let dist be a  $|V| \times |V|$  array of minimum
distances initialized to  $\infty$  (infinity)

for each edge (u, v) do
  dist[u][v]  $\leftarrow$  w(u, v) // The weight of
the edge (u, v)

for each vertex v do
  dist[v][v]  $\leftarrow$  0

for k from 1 to  $|V|$ 
  for i from 1 to  $|V|$ 
    for j from 1 to  $|V|$ 
      if dist[i][j] > dist[i][k] +
dist[k][j]
        dist[i][j]  $\leftarrow$  dist[i][k] +
dist[k][j]
      end if

```

III. PENERAPAN ALGORITMA FLOYD WARSHALL PADA LAYANAN SAME DAY DELIVERY

Dari berbagai aplikasi layanan transportasi daring yang ada saat ini, penulis akan menggunakan aplikasi Grab untuk membahas bagaimana penerapan algoritma ini bekerja.

Saat pengemudi menerima pesan pengantaran barang, aplikasi Grab akan mendeteksi titik lokasi dimana pengemudi berada beserta titik-titik lokasi pemesan yang akan dijemput barangnya. Titik-titik lokasi pemesan ini didapatkan dari dalam

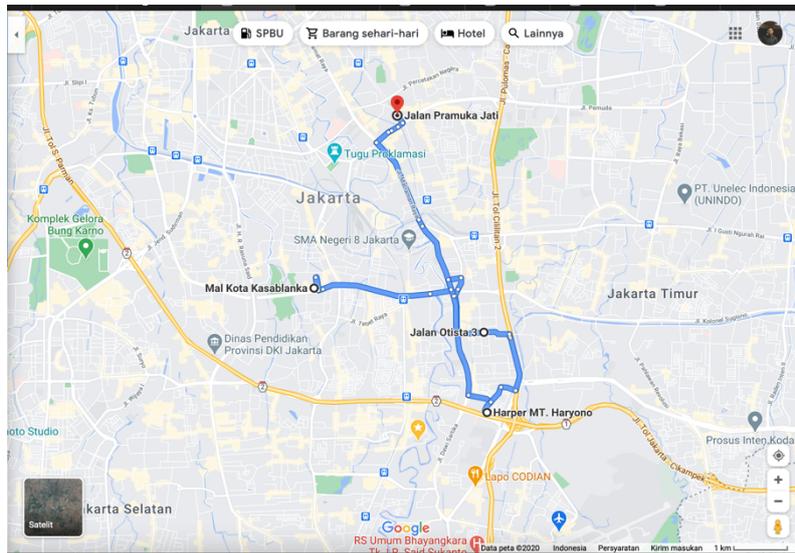
aplikasi Grab tiap pengguna yang terintegrasi dengan *Global Positioning System* atau yang biasa disebut GPS yang dilacak lokasinya secara *real time*. Rute penjemputan barang dari lokasi tiap pengguna dapat diselesaikan menggunakan teori graf dan algoritma Floyd Warshall. Hal ini akan memudahkan pengemudi untuk menentukan urutan lokasi pemesan yang barangnya harus diambil terlebih dahulu sehingga waktu yang dibutuhkan untuk mengantarkan seluruh barang pemesan bisa lebih singkat dan tidak perlu berputar-putar melewati jalan yang sama.



Gambar 10. Tampilan aplikasi pengemudi saat menerima pesan pengantaran barang dari pengguna

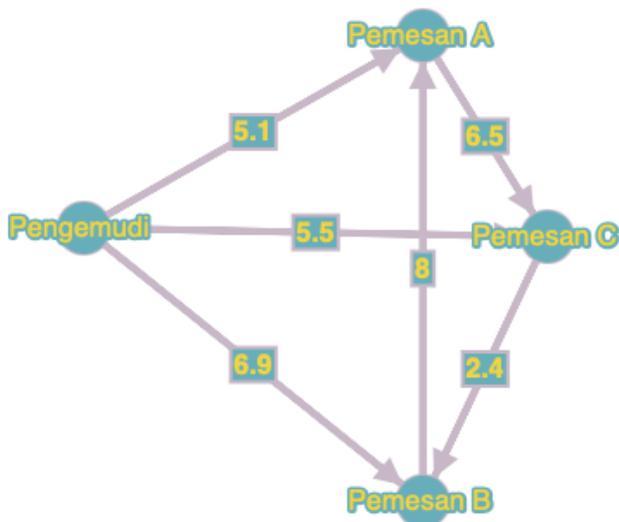
Sumber: youtube.com

Pada Gambar 10 dapat dilihat beberapa titik-titik lokasi pemesan yang sudah masuk ke dalam aplikasi pengemudi dan menunggu untuk barangnya di-pickup. Pengemudi yang saat itu berada di Mal Kota Kasablanka mendapatkan tiga pesanan dari lokasi yang berbeda seperti ditunjukkan pada gambar 11. Selanjutnya, misalkan pemesan yang berada di Jalan Pramuka Jati sebagai pemesan A, pemesan yang berada di Harper MT Haryono sebagai pemesan B, dan pemesan yang berada di Jalan Otista 3 sebagai pemesan C. Lalu, jarak tiap titik lokasi akan dicari dengan menggunakan aplikasi Google Maps dan peta pada gambar 11 dapat kita representasikan ke dalam sebuah graf berbobot sebagai berikut



Gambar 11. Peta titik-titik lokasi pemesan

Sumber: maps.google.com



Gambar 12. Graf berbobot jarak antar titik lokasi

Sumber: Dokumentasi pribadi

Berikut ini tahapan perhitungan manual menggunakan algoritma *Floyd Warshall* dalam menentukan rute terpendek yang direpresentasikan dalam graf pada gambar 12. Perhitungan ini akan menggunakan matriks dimana baris pada matriks menunjukkan titik awal dan kolom pada matriks menunjukkan titik tujuan.

Dari Gambar 12, misalkan lokasi pengemudi sebagai simpul 1, lokasi pemesan A,B, dan C sebagai simpul 2, 3, dan 4, sehingga didapat:

- k = 0, 1, 2, 3, 4
- i = 1, 2, 3, 4
- j = 1, 2, 3, 4

Dapat dibentuk sebuah matriks hubung graf, k = 0.

k = 0	1	2	3	4
1	0	5,1	6,9	5,5
2	∞	0	∞	6,5
3	∞	8	0	∞
4	∞	∞	2,4	0

Matriks hubung graf, k = 1.

k = 1	1	2	3	4
1	0	5,1	6,9	5,5
2	∞	0	∞	6,5
3	∞	8	0	∞
4	∞	∞	2,4	0

Rumus: $X[i,j] \leq X[i,k] + X[k,j]$

Penyelesaian:

- $X[2,3] = X[2,1] + X[1,3]$
 $\infty = \infty + 6,9$
- $X[2,4] = X[2,1] + X[1,4]$
 $6,5 = \infty + 5,5$
- $X[3,2] = X[3,1] + X[1,2]$
 $8 = \infty + 5,1$
- $X[3,4] = X[3,1] + X[1,4]$
 $\infty = \infty + 5,5$
- $X[4,2] = X[4,1] + X[1,2]$
 $\infty = \infty + 5,1$
- $X[4,3] = X[4,1] + X[1,3]$
 $2,4 < \infty + 6,9$

Matriks hubung graf, k = 2.

k = 2	1	2	3	4
1	0	5,1	6,9	5,5

2	∞	0	∞	6,5
3	∞	8	0	14,5
4	∞	∞	2,4	0

- $X[3,1] = X[3,4] + X[4,1]$
 $\infty = 14,5 + \infty$
- $X[3,2] = X[3,4] + X[4,2]$
 $8 < 14,5 + 10,4$

Penyelesaian:

- $X[1,3] = X[1,2] + X[2,3]$
 $6,9 < 5,1 + \infty$
- $X[1,4] = X[1,2] + X[2,4]$
 $5,5 < 5,1 + 6,5$
- $X[3,1] = X[3,2] + X[2,1]$
 $\infty = 8 + \infty$
- $X[3,4] = X[3,2] + X[2,4]$
 $14,5 = 8 + 6,5$
- $X[4,1] = X[4,2] + X[2,1]$
 $\infty = \infty + \infty$
- $X[4,3] = X[4,2] + X[2,3]$
 $2,4 < \infty + \infty$

Matriks hubung graf, $k = 3$.

$k = 3$	1	2	3	4
1	0	5,1	6,9	5,5
2	∞	0	∞	6,5
3	∞	8	0	14,5
4	∞	10,4	2,4	0

Penyelesaian:

- $X[1,2] = X[1,3] + X[3,2]$
 $5,1 < 6,9 + 8$
- $X[1,4] = X[1,3] + X[3,4]$
 $5,5 < 6,9 + \infty$
- $X[2,1] = X[2,3] + X[3,1]$
 $\infty = \infty + \infty$
- $X[2,4] = X[2,3] + X[3,4]$
 $6,5 < 8 + 14,5$
- $X[4,1] = X[4,3] + X[3,1]$
 $\infty = 2,4 + \infty$
- $X[4,2] = X[4,3] + X[3,2]$
 $10,4 = 2,4 + 8$

Matriks hubung graf, $k = 4$.

$k = 4$	1	2	3	4
1	0	5,1	6,9	5,5
2	∞	0	8,9	6,5
3	∞	8	0	14,5
4	∞	10,4	2,4	0

Penyelesaian:

- $X[1,2] = X[1,4] + X[4,2]$
 $5,1 < 5,5 + 10,4$
- $X[1,3] = X[1,4] + X[4,3]$
 $6,9 < 5,5 + 2,4$
- $X[2,1] = X[2,4] + X[4,1]$
 $\infty = 6,5 + \infty$
- $X[2,3] = X[2,4] + X[4,3]$
 $8,9 = 6,5 + 2,4$

Hasil akhir lintasan terpendek Floyd Warshall adalah sebagai berikut,

$k = 4$	1	2	3	4
1	0	5,1	6,9	5,5
2	∞	0	8,9	6,5
3	∞	8	0	14,5
4	∞	10,4	2,4	0

dimana:

- Pengemudi – Pemesan A – Pemesan B – Pemesan C
 $= 5,1 + 8,9 + 6,5 = 20,5$ km
- Pengemudi – Pemesan A – Pemesan C – Pemesan B
 $= 5,1 + 6,5 + 2,4 = 14$ km
- Pengemudi – Pemesan B – Pemesan A – Pemesan C
 $= 6,9 + 8 + 6,5 = 21,4$ km
- Pengemudi – Pemesan B – Pemesan C – Pemesan A
 $= 6,9 + 14,5 + 10,4 = 31,8$ km
- Pengemudi – Pemesan C – Pemesan A – Pemesan B
 $= 5,5 + 10,4 + 8,9 = 24,8$ km
- Pengemudi – Pemesan C – Pemesan B – Pemesan A
 $= 5,5 + 2,4 + 8 = 15,9$ km

Jadi, berdasarkan perhitungan secara manual menggunakan algoritma Floyd Warshall didapatkan rute terpendeknya adalah Pengemudi – Pemesan A – Pemesan C – Pemesan B yaitu sejauh 14 km. Hasil ini sangat bermanfaat bagi pengemudi untuk bisa menghemat waktu, bahan bakar, dan tenaga dalam proses pengantaran barang sehingga pengantaran barang bisa berlangsung lebih cepat dan efisien.

IV. KESIMPULAN

Permasalahan yang sering dialami para *driver* ojek berbasis daring adalah kurangnya waktu yang mereka miliki untuk mencapai jumlah pesanan yang mereka targetkan untuk mendapatkan bonus. Masalah ini sering muncul akibat rute *pick up* barang layanan *same day delivery* yang sering berputar-putar sehingga membuang banyak waktu. Permasalahan ini dapat diselesaikan dengan salah satu metode matematika, yaitu Graf. Titik-titik lokasi setiap pemesan dapat dianalogikan sebagai simpul-simpul dalam sebuah graf dan sisi-sisi yang saling menghubungkan tiap simpul diberi bobot sesuai dengan jaraknya. Dengan adanya algoritma *Floyd Warshall*, dapat dicari rute tercepat dari lokasi pengemudi ke masing-masing lokasi pemesan. Makalah ini masih jauh dari kata sempurna, salah satu aspek yang belum ditanani yaitu apabila salah satu jalan yang dilalui mengalami kemacetan.

Penerapan dari algoritma *Floyd Warshall* ini bisa diperluas dalam aspek kehidupan sehari-hari yang lainnya. Penggunaan algoritma ini sangat berguna bagi para *driver-driver* ojek berbasis daring untuk memudahkan mereka dalam mencari nafkah dan melayani pengguna jasa *same day delivery*.

V. UCAPAN TERIMA KASIH

Pertama-tama, penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa, atas berkat dan rahmat-Nya, penulis bisa menyelesaikan tugas makalah ini. Penulis juga mengucapkan terimakasih kepada Ibu Harlili selaku dosen mata kuliah IF2120 Matematika Diskrit, yang selama ini mengajar penulis dan juga memberikan ilmu dan berbagai nasihat yang sangat membantu pengerjaan makalah ini. Penulis juga tidak lupa mengucapkan terima kasih kepada Bapak Rinaldi Munir, yang selama satu semester ini selalu membantu penulis dengan menyediakan website yang berisi materi-materi kuliah, latihan-latihan soal untuk kuis dan ujian, dan semua dokumen pembelajaran, soal dan lainnya yang tentunya sangat berguna dalam proses pembelajaran mata kuliah IF2120 Matematika Diskrit.

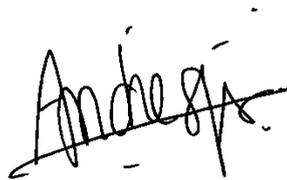
REFERENSI

- [1] <https://www.quora.com/What-are-technologies-behind-Go-Jek-app#>, diakses pada 5 Desember 2020 pukul 21.05.
- [2] R. A. D. Novandi, "Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan (Single Pair Shortest Path)", Bandung : Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2007.
- [3] Rosen, K. H. Discrete Mathematics and Its Application 7th Edition. New York: McGraw-Hill, 2012.
- [4] B. Yulianto et al., "Location based services: positioning techniques," Provider, vol. 12, no. 1, pp. 4982–4987, 2010.
- [5] M. S. Handaka, "Perbandingan Algoritma Dijkstra (Greedy), Bellman-Ford (BFS-DFS), dan Floyd-Warshall (Dynamic Programming) dalam Pengaplikasian Lintasan Terpendek pada Link-State Routing Protocol," 2011.
- [6] K.S. Avdhesh, and K. Sourabh, Finding of Shortest Path from Source to Destination by traversing every node in wired Network, International Journal of Engineering and Technology, 5, 2013, 2655-2656.

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 2020



Andres Jerriel Sinabutar / 13519218