Aplikasi Algoritma Dijkstra dalam Penentuan Posko Bantuan COVID-19 di Kecamatan Penjaringan Jakarta Utara

Alif Bhadrika Parikesit - 13519186[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13519186@std.stei.itb.ac.id

Abstrak*—Tidak dapat kita pungkiri bahwa pandemi COVID-19 yang masuk ke Indonesia Maret 2020 lalu sangat merugikan banyak pihak dari semua kalangan. Kebijakan PSBB membuat kegiatan perekonomian tersendat bahkan lumpuh. Perekonomian yang lumpuh, membuat banyak perusahaan terancam bangkrut dan salah satu cara menghindarinya adalah dengan melakukan PHK terhadap sebagian karyawannya atau melakukan potongan tunjangan karyawan yang tentunya sangat merugikan. Salah satu upaya untuk membantu meringankan beban masyarakat adalah dengan membangun posko-posko yang menyediakan kebutuhan penduduk di tempat-tempat yang strategis dan terjangkau seluruh daerah di sekitarnya dengan mudah. Pada makalah ini akan dibahas tentang pengaplikasian algoritma djikstra dalam mengidentifikasi lintasan terpendek dari satu kelurahan ke kelurahan lainnya di Kecamatan Penjaringan. Akhirnya, dapat ditentukan kelurahan mana yang dapat menjadi pusat posko bantuan COVID-19 yang paling dapat dijangkau dengan mudah oleh kelurahan lain, yaitu kelurahan dengan jarak rata-rata terendah kelurahan tersebut dari kelurahan lainnya di Kecamatan Penjaringan, Jakarta Utara.

***Kata Kunci*—algoritma dijkstra, graf, shortest path, posko bantuaan**

I. Pendahuluan

Awal Bulan Maret 2019, pemerintah Indonesia mengumumkan kasus COVID-19 pertamanya. Sejak saat itu, upaya pencegahan penularan sudah mulai dilakukan, namun angka penularan COVID-19 tetap saja meningkat setiap harinya. Hal itu tentu membuat pemerintah mau tidak mau menerapkan kebijakan PSBB untuk menekan angka penularan virus, namun dampak buruknya adalah lumpuhnya perekonomian negara. Puncaknya Indonesia memasuki resesi pertama setelah krisis 1998, pada awal November 2020.

Lumpuhnya kegiatan perekonomian tentu membuat banyak perusahaan terancam bangkrut. Salah satu upaya menghindari kebangkrutan adalah dengan melakukan PHK atau pemotongan tunjangan kerja karyawan. Kebijakan tersebut tentu sangat merugikan banyak pihak di seluruh daerah di Indonesia, salah satunya daerah Kecamatan Penjaringan, Jakarta Utara.

Upaya yang dapat dilakukan untuk membantu para korban terdampak yaitu dengan dilakukannya pembangunan pos komando (posko) bantuan COVID-19 yang menjadi pusat penerimaan bantuan dan penyaluran serta penyediaan bantuan untuk wilayah-wilayah lain di Kecamatan Penjaringan. Pembangunan posko tersebut harus dapat menjangkau daerah kelurahan lainnya dengan mudah dan merata.

Pada makalah ini penulis akan menerapkan Algoritma Dijkstra untuk menentukan panjang lintasan terpendek dari masing-masing kelurahan ke setiap kelurahan lainnya di Kecamatan Penjaringan. Kemudian, dari hasil tersebut dapat ditentukan kelurahan mana yang dapat dijadikan tempat pembangunan posko bantuan COVID-19.

II. Landasan Teori

A. Graf

1. Definisi Graf

Graf dalam struktur diskrit adalah sebuah struktur data yang digunakan untuk merepresentasikan objek-objek diskrit beserta hubungannya dalam antar objek tersebut. Graf memiliki dua komponen utama, yaitu simpul (*node)* yang merepresentasikan objek-objek pada graf dan dinyatakan sebagai titik, dan sisi (*edge)* yang merepresentasikan hubungan antar objek graf dan dinyatakan sebagai garis pada graf.

Secara formal, graf didefinisikan dengan notasi G = (V, E), dengan V (*vertices* atau *node*) adalah simpul graf dan E (*edges*) sebagai sisi penghubung sepasang simpul pada graf. V dapat ditulis sebagai V = $\{v_1, v_2, v_3, ..., v_n\}$ dengan V bukan himpunan kosong dan n adalah banyak simpul, sedangkan E dapat ditulis E = $\{e_1, e_2, e_3, ..., e_m\}$dengan E boleh merupakan himpunan kosong dan m adalah banyak sisi. Dengan kata lain, dimungkinkan membuat suatu graf yang tidak memiliki sisi dengan jumlah simpul minimal sebanyak satu buah simpul.

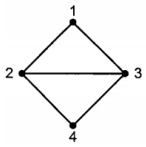
Sisi pada graf dinyatakan sebagai pasangan e = (v_1, v_2) yang berarti sisi e merupakan sisi yang menghubungkan simpul v_1 dengan simpul v_2. Simpul dapat ditulis dengan huruf, angka, maupun kombinasi keduanya. Pada graf, dimungkinkan adanya simpul yang memiliki sisi ke dirinya sendiri yang disebut dengan kalang (*loop)*, kemudian dimungkinkan pula adanya simpul yang memiliki lebih dari satu sisi yang disebut sisi ganda (*multiple edges*).

2. Jenis-Jenis Graf

Berdasarkan ada tidaknya gelang/kalang atau sisi ganda pada suatu graf, graf dapat dikelompokkan menjadi dua jenis yaitu:

a. Graf Sederhana

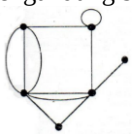
Graf sederhana adalah graf yang tidak mengandung gelang dan sisi ganda.



Gambar 1. *Graf sederhana*
(Sumber:[3])

b. Graf Tak Sederhana

Graf tak sederhana dalah graf yang dapat mengandung sisi ganda atau gelang ataupun keduanya. Graf tak sederhana dibagi lagi menjadi dua yaitu graf ganda dan graf semu. Graf ganda adalah graf yang mengandung sisi ganda sedangkan graf semu adalah graf yang mengandung gelang dan dapat pula mengandung sisi ganda.



Gambar 2. *Graf tak sederhana*
(Sumber:[3])

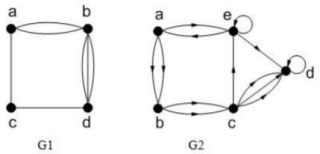
Berdasarkan orientasi arah pada sisi graf, graf dibedakan kembali menjadi dua jenis, yaitu:

a. Graf Tak Berarah

Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah, dengan kata lain sisi e = (v_1, v_2) akan sama dengan sisi (v_2, v_1).

b. Graf Berarah

Kebalikan dari graf tak berarah, graf berarah merupakan graf yang sisinya memiliki orientasi arah dan disebut sebagai busur (*arcs*). Busur (v_1, v_2) memiliki arti sebagai sisi yang menghubungan v_1 sebagai simpul asal dan v_2 sebagai simpul tujuan/termninal, dan (v_1,v_2) ≠ (v_2,v_1).



Gambar 3. *G_1 adalah graf tak berarah, G_2 adalah graf berarah*
(Sumber:[3])

Selanjutnya berdasarkan jumlah simpulnya, graf dibedakan menjadi dua jenis, yaitu:

a. Graf Berhingga

Graf berhingga adalah graf yang jumlah simpulnya berhingga n.

b. Graf Tak Berhingga

Kebalikan dari graf berhingga, graf tak berhingga adalah graf yang jumlah simpulnya tidak terhingga banyaknya.

3. Terminologi Dasar Graf

Berikut ini adalah beberapa terminologi atau istilah-istilah dasar berkaitan dengan graf.

a. Bertetangga (*Adjacent*)

Dua buah simpul pada graf G dikatakan bertetangga jika keduanya terhubung langsung dengan sebuah sisi pada graf tak berarahnya.

b. Bersisian (*Incidency*)

Sisi e = (v_1, v_2) dikatakan bersisian dengan simpul v_1 dan v_2 karena v_1 dan v_2 bertetangga melalui e.

c. Simpul Terpencil (*Isolated Vertex*)

Simpul terpencil adalah simpul yang tidak mempunyai tetangga atau dengan kata lain simpul yang tidak berhubungan sama sekali dengan simpul lain.

d. Graf Kosong (*Null Graph*)

Graf kosong adalah graf yang tidak memiliki sisi di dalamnya, dengan kata lain graf G = (V, E) dengan V adalah himpunan kosong.

Gambar 4. *Graf Kosong N_5*
(Sumber:[3])

e. Derajat (*Degree*)

Derajat suatu simpul *v* pada graf tak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut, dapat dikaatakan pula sebagai banyak sisi terhubung dengan simpul *v*. Notasi derajat pada simpul *v* dituliskan *d(v)*. Sedangkan pada graf berarah, derajat simpul *v* terdiri atas $d_{in}(v)$ sebagai jumlah busur masuk ke *v* dan $d_{out}(v)$ sebagai jumlah busur yang keluar dari simpul *v*. Kemudian penjumlahan derajat masuk dan keluar simpul *v* adalah nilai derajat simpul *v* pada graf berarah, $d(v) = d_{in}(v) + d_{out}(v)$.

f. Lintasan (*Path*)

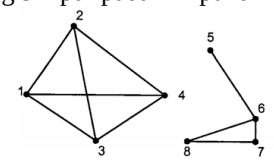
Lintasan dengan panjang n dari simpul awal ke simpul tujuan di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk v_0, e_1, v_1, e_2, v_2, ..., e_n, v_n sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ..., $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf. Pada graf sederhana, lintasan cukup dituliskan sebagai barisan dari simpul-simpul, tetapi untuk graf berarah perlu dituliskan pula sisinya untuk menghindari kerancuan. Selanjutnya, panjang lintasan merupakan jumlah sisi dalam lintasan tersebut. Lintasan yang berawal dan berakhir pada simpul yang sama disebut lintasan tertutup, sebaliknya disebut llintasan terbuka.

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

Siklus atau sirkuit adalah lintasan tertutup, yaitu lintasan yang berawal dan berakhir pada simpul yang sama. Panjang sirkuit adalah jumlah sisi di dalam sirkuit tersebut.

h. Terhubung (*Connected*)

Dua buah simpul dikatakan terhubung jika terdapat lintasan dari simpul awal ke simpul tujuan. Suatu graf G = (V, E) dikatakan terhubung jika dan hanya jika setiap simpul graf tak berarahnya terhubung atau dengan kata lain setiap pasang simpul pada himpunan V terhubung.



Gambar 5. *Graf Tak Terhubung*
(Sumber:[3])

i. Upagraf (*Subgrapgh*) dan Komplemen Upagraf

Graf $G_1 = (V_1,E_1)$ adalah *subgraph* dari graf G = (V, E) jika V_1 adalah himpunan bagian dari V dan E_1 adalah himpunan bagian dari E. Sedangkan komplemen upagraf G_1 adalah adalah graf $G_2 = (V_2, E_2)$ dengan $E_2 = E – E_1$ dan V_2 adalah simpul-simpul yang bersisian dengan anggota E_2.

j. Upagraf Merentang (*Spanning Subgraph*)

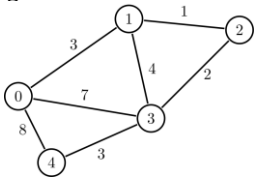
Upagraf merentang dari G = (V , E) adalah upagraf $G_1 = (V_1, E_1)$ yang mengandung semua simpul dari G, dengan kata lain $V = V_1$.

k. Cut-set

Cut-set adalah himpunan sisi yang jika dibuang dari graf terhubungnya akan menyebabkan graf tersebut menjadi tidak terhubung lagi dan menghasilkan dua komponen upagraf.

l. Graf Berbobot (*Weighted Graph*)

Graf berbobot adalah graf yang setiap sisinya diberi nilai atau bobot. Bobot dapat merepresentasikan berbagai hal tergantung permasalahannya. Sebagai contoh, bobot dapat merepresentasikan jarak antar kota, waktu tempuh antar kota, keterjangkauan antar kota, dll.



Gambar 6. *Graf Berbobot*
(Sumber: https://hyperskill.org/)

B. Algoritma Dijkstra

Permasalahan pencarian rute terpendek adalah salah satu permasalahan yang dapat diselesaikan dengan konsep graf. Salah satu metode pencarian lintasan dengan panjang terpendek adalah dengan algoritma djikstra. Algoritma Dijkstra ditemukan oleh Edsger Dijsktra dan dianggap sebagai algoritma greedy, yaitu algoritma yang digunakan untuk memecahkan

Dalam pencarian jalur terpendeknya, algoritma dijskatra mencari bobot terkecil dari suatu graf berbobot tidak negatif. Jalur terpendek dapat diperoleh dari dua atau lebih titik dari suatu graf dengan nilai total yang didapat bernilai paling kecil. Misalkan graf G adalah graf berarah dan berbobot tidak negatif dengan definisi G = {V, E}. Algoritma Dijsktra akan membentuk suatu jalur dari satu simpul ke simpul lain secara optimal (pencarian bobot terkecil) setiap langkah.

Langkah-langkah yang dilakukan pada algoritma Dijkstra adalah sebagai berikut [1]:

a. Tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap.
b. 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.
c. Set semua node yang belum dilalui dan set node awal sebagai “Node keberangkatan”
d. Dari node keberangkatan, pertimbangkan node tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru
e. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah dilalui sebagai “Node dilewati”. Node yang dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
f. Set “Node belum dilewati” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan ulangi langkah e.

Berikut ini adalah pseudocode algoritma dijkstra.

ALGORITHM 1 Dijkstra's Algorithm.

procedure *Dijkstra*(G: weighted connected simple graph, with all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$ where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G}
for $i := 1$ **to** n
  $L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{the labels are now initialized so that the label of a is 0 and all other labels are ∞, and S is the empty set}
while $z \notin S$
  $u :=$ a vertex not in S with $L(u)$ minimal
  $S := S \cup \{u\}$
  for all vertices v not in S
    if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
    {this adds a vertex to S with minimal label and updates the labels of vertices not in S}
return $L(z)$ {$L(z)$ = length of a shortest path from a to z}

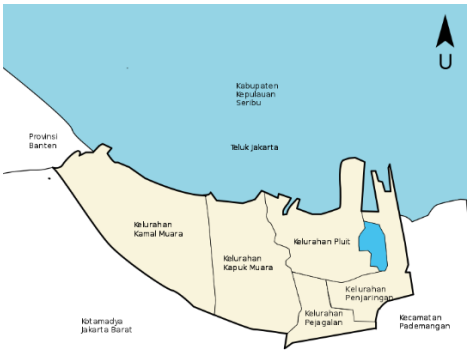
Gambar 7. *Algoritma Dijkstra*
(Sumber: [2])

C. Kecamatan Penjaringan Jakarta Utara

Penjaringan adalah kecamatan yang terletak di Kota Jakarta Utara, DKI Jakarta dengan luas wilayah sebesar 35.47 km^2. Jumlah penduduk Kecamatan Penjaringan pada tahun 2020 berjumlah 315.511 jiwa, dengan kepadatan penduduk 8.895 jiwa/km^2. Kecamatan Penjaringan memiliki 5 kelurahan yaitu:

a. Kelurahan Kamal Muara
b. Kelurahan Karang Muara
c. Kelurahan Pluit
d. Kelurahan Penjagalan

e. Kelurahan Penjaringan

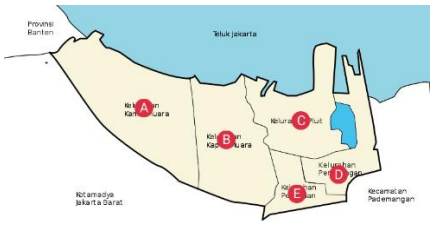


Gambar 8. *Peta Kecamatan Penjaringan Jakarta Utara*
(Sumber: https://id.wikipedia.org)

III. PEMBAHASAN

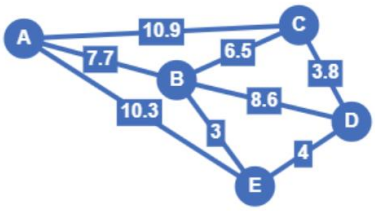
A. Pendefinisian Graf Berdasarkan Peta Kecamatan Penjaringan

Untuk dapat menerapkan algoritma djikstra dalam mencari jarak lintasan terpendek dari setiap kelurahan, diperlukan pendefinisian graf yang merepresentasikan denah atau peta Kecamatan Penjaringan. Simpul pada graf menyatakan kelurahan, sisi menyatakan jalur yang dapat ditempuh pada sepasang simpul. Adapun bobot pada sisi graf menyatakan panjang lintasan perjalanan dari kantor kelurahan yang ada di kelurahan pada simpul terkait dalam kilometer(km).



Gambar 9. *Representasi Kelurahan dengan Bentuk Simpul*
(Sumber: https://id.wikipedia.org)

Simpul A menyatakan Kelurahan Kamal Muara, simpul B menyatakan Kelurahan Kapuk Muara, simpul C menyatakan Kelurahan Pluit, simpul D menyatakan Kelurahan Penjaringan, simpul E menyatakan Kelurahan Penjagalan. Setelah dihubungkan dan diberi bobot, berikut ini adalah representasi peta Kecamatan Penjaringan dalam bentuk graf berbobot.



Gambar 10. *Representasi Peta Kecamaatan Penjaringan dalam Bentuk Graf*
(Sumber: https://graphonline.ru)

B. Pencarian Jarak Terpendek

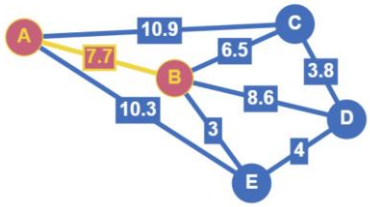
Setelah berhasil merepresentasikan peta Kecamatan Penjaringan, selanjutnya akan dicari jarak terpendek dari masing-masing pasang kelurahan pada himpunan simpul graf pada gambar dengan mengimplementasikan algoritma djikstra sesuai dengan bagian landasan teori sebelumnya

1. Lintasan terpendek dari A

V	A	B	C	D	E
A	0_A	7.7_A	10.9_A	∞	10.3_A
B		7.7_A	10.9_A	16.3_B	10.3_A
E			10.9_A	14.3_E	10.3_A
C			10.9_A	14.3_E	
D				14.3_E	

Tabel 1. *Tabel Iterasi Djikstra's Algorithm dengan A Sebagai Simpul Awal*

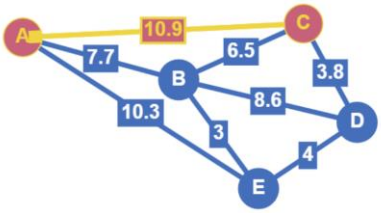
a. A menuju B



Gambar 11. *Lintasan Terpendek A-B*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 1, lintasan terpendek yang didapat adalah A-B dengan panjang 7.7 km

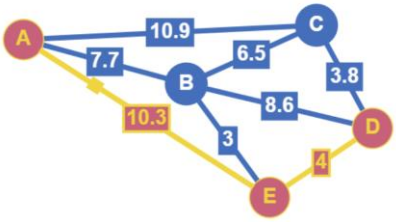
b. A menuju C



Gambar 12. *Lintasan Terpendek A-C*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 1, lintasan terpendek yang didapat adalah A-C dengan panjang 10,9 km.

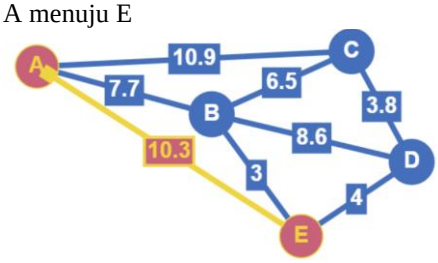
c. A menuju D



Gambar 13. *Lintasan Terpendek A-D*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 1, lintasan terpendek yang didapat adalah A-E-D dengan panjang 14,3 km

d. A menuju E



Gambar 14. *Lintasan Terpendek A-E*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 1, lintasan terpendek yang didapat adalah A-E dengan panjang 10,3 km.

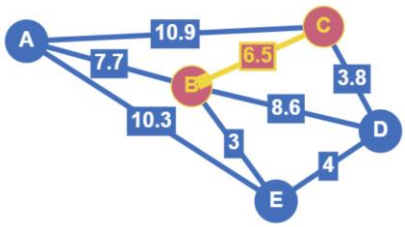
2. Lintasan terpendek dari B

V	A	B	C	D	E
B	7.7_B	0_B	6.5_B	8.6_B	3_B
E	7.7_B		6.5_B	7_E	3_B
C	7.7_B		6.5_B	7_E	
D	7.7_B			7_E	
A	7.7_B				

Tabel 2. *Tabel Iterasi Djikstra's Algorithm dengan B Sebagai Simpul Awal*

a. B menuju A
Graf berbobot tidak berarah, sehingga jarak B ke A sama dengan jarak A ke B yaitu 7,7 km.

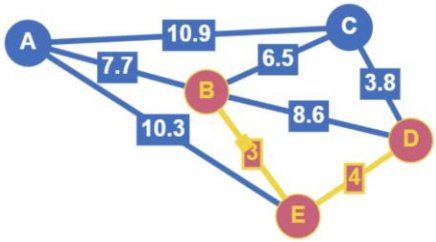
b. B menuju C



Gambar 15. *Lintasan Terpendek B-C*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 2, lintasan terpendek yang didapat adalah B-C yaitu 6,3 km.

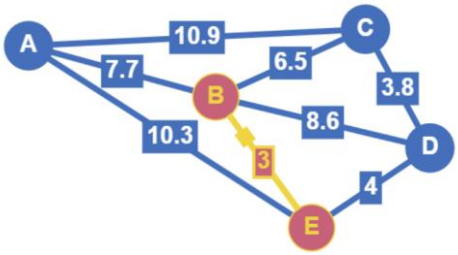
c. B menuju D



Gambar 16. *Lintasan Terpendek B-D*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 2, lintasan terpendek yang didapat adalah B-E-D dengan panjang 7 km.

d. B menuju E



Gambar 17. *Lintasan Terpendek B-E*
(Sumber: https://graphonline.ru)

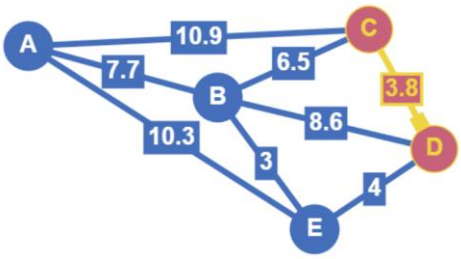
Berdasarkan tabel 2, lintasan terpendek yang didapat adalah B-E dengan panjang 3 km.

3. Lintasan terpendek dari C

V	A	B	C	D	E
C	10.9_C	6.5_C	0_C	3.8_C	∞
D	10.9_C	6.5_C		3.8_C	7.8_D
B	10.9_C	6.5_C			7.8_D
E	10.9_C				7.8_D
A	10.9_C				

Tabel 3. *Tabel Iterasi Djikstra's Algorithm dengan C Sebagai Simpul Awal*

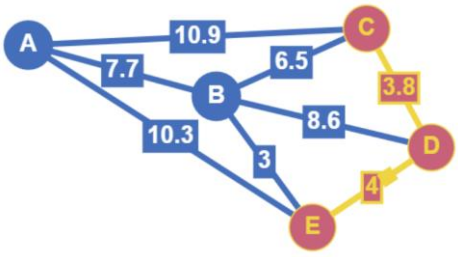
a. C menuju A
Sama dengan jarak A ke C, 10,9 km.
b. C menuju B
Sama dengan jarak B ke C, 6,5 km.
c. C menuju D



Gambar 18. *Lintasan Terpendek C-D*
(Sumber: https://graphonline.ru)

Berdasarkan tabel 3, lintasan terpendek yang didapat adalah C-D dengan panjang 3,8km.

d. C menuju E



Gambar 19. *Lintasan Terpendek C-E*
(Sumber: https://graphonline.ru)

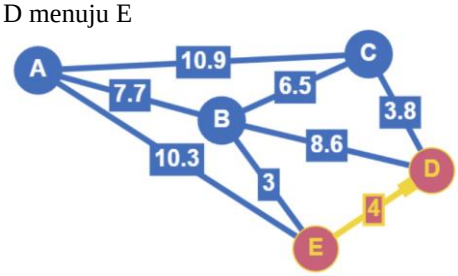
Berdasarkan tabel 3, lintasan terpendek yang didapat adalah C-D-E dengan panjang 7,8km.

4. Lintasan terpendek dari D

V	A	B	C	D	E
D	∞	8.6_D	3.8_C	0_D	4_C
C	13.8_A	8.6_D	3.8_C		4_C
E	14.3_E	7_E			4_C
B	14.3_E	7_E			
A	14.3_E				

Tabel 4. *Tabel Iterasi Djikstra's Algorithm dengan D Sebagai Simpul Awal*

a. D menuju A
   Sama dengan jarak A ke D, 14, 3 km.
b. D menuju B
   Sama dengan jarak B ke D, 7 km.
c. D menuju C
   Sama dengan jarak C ke D 3,8 km.
d. D menuju E



Gambar 20. *Lintasan Terpendek D-E (Sumber: https://graphonline.ru)*

Berdasarkan tabel 4, lintasan terpendek yang didapat adalah D-E dengan panjang 4 km.

5. Lintasan terpendek dari E

V	A	B	C	D	E
E	10.3_E	3_E	∞	4_E	0_E
B	10.3_E	3_E	9.5_B	4_E	
D	10.3_E		7.8_D	4_E	
C	10.3_E		7.8_D		
A	10.3_E				

Tabel 5. *Tabel Iterasi Djikstra's Algorithm dengan E Sebagai Simpul Awal*

a. E menuju A
   Sama dengan jarak A ke E, 10,3 km.
b. E menuju B
   Sama dengan jarak B ke E, 3 km.
c. E menuju C
   Sama dengan jarak C ke E, 7,8 km.
d. E menuju D
   Sama dengan jarak D ke E, 4 km.

C. Penentuan Lokasi Posko Bantuan COVID-19

Berdasarkan data yang telah terkumpul pada bagian B di atas, akan ditentukan kelurahan mana yang lebih tepat menjadi tempat pembangunan posko bantuan COVID-19 di Kecamatan Penjaringan, Jakarta Utara. Berikut ini adalah tabel data jarak masing-masing kelurahan terhadap kelurahan lainnya.

***Shortest Cost* (km)**	**A**	**B**	**C**	**D**	**E**
A	X	7,7	10,9	14,3	10,3
B	7,7	X	6,5	7	3
C	10,9	6,5	X	3,8	7,8
D	14,3	7	3,8	X	4
E	10,3	3	7,8	4	X

Tabel 6. *Jarak Lintasan Terpendek Setiap Kelurahan terhadap Kelurahan Lain di Kecamatan Penjaringan*

Selanjutnya, berdasarkan tabel 1 di atas, dapat dihitung nilai rata-rata jarak setiap kelurahan dengan kelurahan lainnya dengan cara menjumlahkan *shortest cost* setiap kolom pada masing-masing baris, kemudian dibagi dengan jumlah kelurahan selain kelurahan pada baris tersebut.

Kelurahan / Simpul	**Rata-Rata Jarak Kelurahan ke Kelurahan Lain (km)**
Kamal Muara / A	10,8
Kapuk Muara / B	6,05
Pluit / C	7,25
Penjaringan / D	7,275
Penjagalan / E	6,275

Tabel 7. *Rata-Rata Jarak Lintasan Setiap Kelurahan terhadap Kelurahan Lain di Kecamatan Penjaringan*

Dapat dilihat bahwa rata-rata jarak yang harus ditempuh dari A ke simpul lain atau dari simpul lain ke A adalah 10,8 km, jarak dari/ke B 6,05 km, jarak dari/ke C 7,25 km, jarak dari/ke D 7,275 km, jarak dari/ke E 6,275 km.

Unruk membangun sebuah pusat posko bantuan COVID-19 yang menyediakan bantuan untuk seluruh kelurahan diperlukan satu daerah dimana memiliki jarak ke daerah lain paling rendah agar semakin terjangkau dan cepat mendapatkan bantuan. Dengan begitu daerah yang paling sesuai dengan spesifikasi tersebut adalah Kelurahan Kapuk Muara, karena penduduk kelurahan lain memerlukan bantuan, rata-rata hanya perlu menempuh sejauh 6,05 km.

IV. KESIMPULAN

Pemahaman terkait konsep graf dalam struktur diskrit sangat berguna dalam menyelesaikan berbagai persoalan. Salah satu permasalahan yang dapat diselesaikan dengan pendekatan graf adalah persoalan pencarian jarak terpendek atau rute paling mangkus untuk menempuh suatu perjalanan. Pada makalah ini, dengan pendekatan graf dan pengaplikasian algoritma djikstra, dapat ditentukan lintasan-lintasan terpendek dari setiap kelurahan ke kelurahan lainnya di Kecamatan Penjaringan, Jakarta Utara. Setelah mendapatkan lintasan terpendek tersebut, dapat dengan mudah ditentukan kelurahan mana yang paling tepat dijadikan sebagai pusat posko bantuan COVID-19, dalam kasus ini yaitu Kelurahan Karang Muara karena memiliki rata-rata jarak terpendek menuju kelurahan lain yaitu sebesar 6,05 km.

V. UCAPAN TERIMA KASIH

Pada kesempatan ini penulis ingin manyampaikan rasa syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya penulis diberi kesehatan sehingga dapat menyelesaikan makalah ini dengan baik. Kemudia, penulis berterima kasih kepada orang tua penulis yang telah memberikan dukungan baik secara moral maupun materi dalam proses penulisan makalah ini. Tidak lupa, penulis mengucapkan terima kasih kepada dosen mata kuliah matematika diskrit K-02, yaitu Ibu Dra. Harlili S., M.Sc. yang telah membina dan membimbing penulis dalam kegiatan perkuliahan mata kuliah matematika diskrit selama satu semester ini. Terakhir, ucapan terima kasih juga tak lupa saya berikan kepada teman-teman Teknik Informatika 2019 dan Alifia Adila Asmara yang telah menjadi penyemangat sekaligus pengingat untuk selalu berproses menjadi lebih baik.

REFERENCES

[1] Girsang, Abba Suganda, "Algoritma Dijkstra", (Online), https://mti.binus.ac.id/2017/11/28/algoritma-dijkstra/, diakses 8 Desember 2020

[2] Kenneth. H. Rosen, Discrete Mathematics and Its Application (7th edition), pp.709-712

[3] Munir, Rinaldi, "Matematika Diskrit", Bandung: Informatika Bandung, 2010, pp 353-420

[4] Portal Provinsi DKI, https://jakarta.go.id/artikel/konten/3617/penjaringan-kecamatan, diakses 10 Desember 2020

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, 10 Desember 2020

Alif Bhadrika Parikesit - 13519186