Pengaplikasian Graf dalam Menentukan Rute Angkutan Kota Tercepat

Rachel Sidney Devianti/13515124[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13515124@std.stei.itb.ac.id

***Abstrak*—Angkot (angkutan kota) merupakan salah satu sarana transportasi yang cukup banyak dijumpai di daerah Bandung. Para penumpang angkutan kota ini terdiri dari berbagai kalangan, seperti: ibu rumah tangga, anak sekolah, dan terutama mahasiswa. Idealnya, para penumpang angkutan kota ini mengharapkan dapat sampai ke tempat tujuan dengan cepat. Namun, terkadang, para penumpang kesulitan menentukan angkutan kota manakah yang lebih efektif untuk dapat sampai ke tempat tujuan mereka. Akibatnya, untuk jarak yang dekat, mereka harus menempuh perjalanan yang cukup lama karena dibawa berkeliling oleh supir angkot. Oleh sebab itu, dengan penerapan graf, algoritma Djikstra, dan algoritma Prim, penulis membuat suatu rancangan perhitungan rute tercepat yang dapat dilalui untuk menghemat waktu dan tenaga.**

***Kata Kunci*—Algoritma Djikstra, Aplikasi Graf, Algoritma Prim, Rute Angkot Tercepat.**

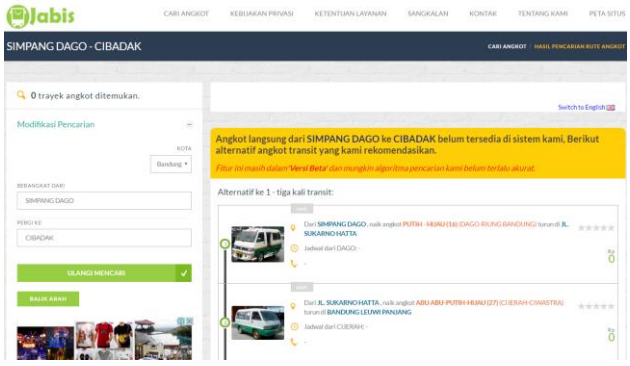
I. PENDAHULUAN

Angkutan kota merupakan salah satu sarana transportasi yang banyak digunakan oleh masyarakat di daerah Bandung, khususnya mahasiswa. Akibat banyaknya universitas yang dibangun di daerah Bandung, tidak sedikit pendatang dari berbagai daerah yang menuntut ilmu di kota ini. Mereka yang berasal dari daerah yang jauh umumnya menggunakan angkutan kota sebagai sarana transportasi utama karena harganya yang terjangkau. Selain itu, angkutan kota juga praktis digunakan untuk bepergian karena tidak perlu memikirkan mengenai masalah parkir.

Masing-masing angkutan kota umumnya memiliki rute yang berbeda-beda. Sebagai contoh, angkutan kota Cisitu-Tegallega yang berwarna ungu memiliki rute sebagai berikut: Jalan Cisitu, Jalan Sangkuriang, Jalan Siliwangi, Jalan Sumur Bandung, Jalan Taman Sari, Jalan Cihampelas, Jalan Wastu Kencana, Jalan Padjadjaran, Jalan Cicendo, Jalan Kebon Kawung, Stasiun Bandung, Jalan Pasir Kaliki, Jalan Kebon Jati, Jalan Suniaraja, dan seterusnya. Rute yang dilalui ini unik dan tidak ada angkutan kota lain yang melewati rute yang sama persis.

Para penumpang umumnya memilih angkutan kota dengan pertimbangan rute yang melewati tempat tujuannya. Namun, tidak banyak yang mempertimbangkan efektifitas waktu dalam pengambilan keputusan ini. Akibatnya, untuk suatu tempat tujuan yang dekat, para penumpang harus menghabiskan waktu yang cukup lama di perjalanan. Selain itu, untuk lokasi tujuan yang ingin ditempuh dari suatu lokasi tertentu, seringkali penumpang harus dibawa berkeliling oleh angkutan kota yang dinaiki karena masalah rute paten yang dimiliki oleh angkutan kota tersebut. Padahal, dengan melakukan transit, penumpang dapat sampai ke tempat tujuan dengan lebih cepat sehingga dapat menghemat waktu dan tenaga.

Dengan semakin berkembangnya teknologi, sudah banyak aplikasi yang dapat digunakan untuk menemukan rute angkot yang efektif. Aplikasi ini bermacam-macam, mulai dari berbasis *Web*, hingga aplikasi yang dapat diakses dengan mudah melalui *smartphone*. Pengguna cukup memasukkan lokasi asal dan lokasi tujuan kemudian aplikasi ini akan memberikan alternatif-alternatif rute yang dapat ditempuh oleh penumpang. Selain itu, adapula yang menggunakan GPS (*Global Positioning System*) untuk melacak lokasi asal sehingga penumpang hanya perlu memasukkan lokasi tujuan.



Gambar 1. Salah satu situs pencarian rute angkot yang bernama jadwalangkot.com. (sumber: http://jadwalangkot.com/jadwal/hasilPencarianRuteAngkot/rute-angkot-BANDUNG-dari-SIMPANG+DAGO-CIBADAK)

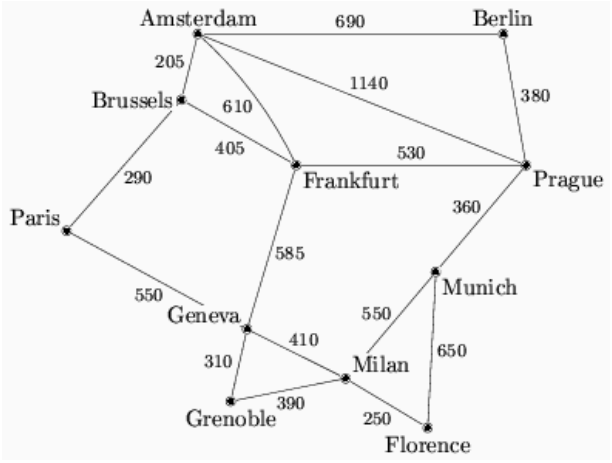
Pada kesempatan kali ini, penulis akan membahas mengenai pengaplikasian teori graf, algoritma Djikstra, dan algoritma Prim dalam penentuan rute angkutan kota

tercepat dari suatu lokasi ke lokasi tertentu untuk menjawab permasalahan yang ada. Lebih lanjut landasan teori yang digunakan akan dibahas pada upabab setelah ini.

II. LANDASAN TEORI

2.1. Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai noktah, bulatan, atau titik (*node*), sedangkan hubungan antara objek tersebut dinyatakan dengan garis atau sisi (*edge*). Menurut catatan sejarah, masalah jembatan Konigsberg adalah masalah yang pertama kali menggunakan graf, masalah ini kemudian diselesaikan oleh seorang ahli bernama Euler.



Gambar 2. Graf berbobot yang merepresentasikan jarak antarkota. (sumber: https://cordis.europa.eu/tmr/src/res970302.htm)

2.1.1. Definisi Graf

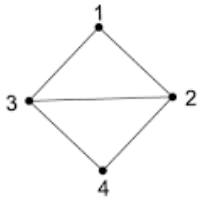
Graf G merupakan pasangan himpunan (V, E) dimana:
V = himpunan tidak kosong dari simpul-simpul (*node*) = $\{v_1, v_2, v_3, ..., v_n\}$
E = himpunan yang mungkin kosong dari sisi (*edge*) yang menghubungkan sepasang simpul = $\{e_1, e_2, e_3, ..., e_n\}$
Graf G dapat ditulis dengan menggunakan notasi G = {V, E}.

Simpul-simpul pada graf tidak boleh merupakan himpunan kosong, sedangkan sisi-sisi pada graf boleh merupakan himpunan kosong. Graf yang hanya mempunyai satu buah simpul tanpa sisi dinamakan graf trivial. Simpul pada graf dapat dinomori dengan huruf, angka, atau gabungan keduanya. Sedangkan, sisi yang menghubungkan dua buah simpul pada graf dapat dinyatakan dengan pasangan (v_i, v_j) atau dengan lambang e_1, e_2, dan seterusnya. Apabila e adalah sisi yang menghubungkan simpul v_i dan v_j maka dapat dituliskan dengan notasi sebagai berikut:
$e = (v_i, v_j)$

2.1.2. Jenis-Jenis Graf

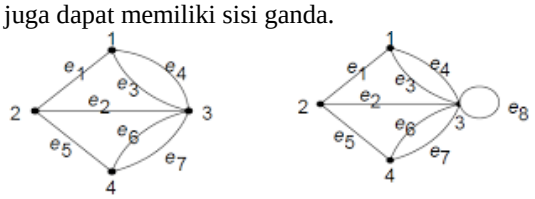
Graf dapat dikelompokkan ke dalam beberapa jenis tergantung pada sudut pandang pengelompokkannya, baik dari segi ada tidaknya sisi ganda atau kalang, jumlah simpul, maupun dari segi orientasi arah pada sisi. Berdasarkan ada tidaknya sisi ganda atau kalang, graf dikelompokkan menjadi:

1. Graf Sederhana
   Graf sederhana merupakan graf yang tidak memiliki sisi ganda maupun kalang.



Gambar 3. Graf Sederhana

2. Graf Tak Sederhana
   Graf tak sederhana merupakan graf yang mengandung sisi ganda atau kalang. Graf tak sederhana ini dibagi lagi menjadi dua, yaitu: graf ganda dan graf semu. Sesuai dengan namanya, graf ganda adalah graf yang mengandung sisi ganda (berjumlah dua atau lebih dari dua) yang menghubungkan dua buah simpul. Graf semu adalah graf yang mengandung gelang. Graf semu juga dapat memiliki sisi ganda.



Gambar 4. Graf Tak Sederhana

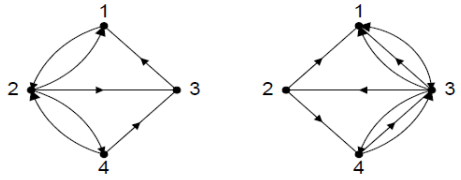
Berdasarkan jumlah simpul pada suatu graf, graf dapat digolongkan menjadi dua jenis:

1. Graf Berhingga
   Graf berhingga merupakan graf yang jumlah simpulnya berhingga (dapat dihitung).
2. Graf Tak Berhingga
   Graf tak berhingga merupakan graf yang jumlah simpulnya tidak berhingga (tidak dapat dihitung).

Sisi pada graf dapat mempunyai orientasi arah. Berdasarkan orientasi arah pada sisi, graf dapat digolongkan menjadi dua jenis, yaitu:

1. Graf Tak Berarah
   Graf tak berarah merupakan graf yang sisinya tidak mempunyai orientasi arah. Pada graf ini, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(v_i, v_j) = (v_j, v_i)$ adalah sisi yang sama. Gambar 3 dan gambar 4 merupakan graf tak berarah.
2. Graf Berarah
   Graf berarah merupakan graf yang setiap sisinya diberikan orientasi arah. Sisi yang berarah disebut juga busur (*arc*). Pada graf berarah, (v_i, v_j) dan (v_j, v_i)

merupakan dua busur yang berbeda. Busur (v_i, v_j) titik pangkalnya berada pada v_i (simpul asal) dan titik ujungnya berada pada v_j (simpul terminal). Graf berarah yang mempunyai sisi ganda atau kalang disebut graf ganda berarah.



(a) (b)

Gambar 5. (a) Graf Berarah (b) Graf Ganda Berarah

2.1.3. Terminologi Graf

Terdapat beberapa terminologi (istilah) yang berkaitan dengan graf. Berikut ini adalah beberapa terminologi tersebut.

1. Bertetangga
   Dua simpul pada graf tak berarah dikatakan bertetangga jika keduanya dihubungkan oleh sebuah sisi. Pada graf berarah, dua buah simpul dikatakan bertetangga jika dihubungkan oleh sebuah busur.
2. Bersisian
   Untuk sembarang sisi e = (v_i, v_j), sisi e dikatakan bersisian dengan simpul v_i dan v_j.
3. Simpul Terpencil
   Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya.
4. Graf Kosong
   Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong, dinyatakan dengan N_n dimana n merupakan jumlah simpul.
5. Derajat
   Derajat suatu simpul pada graf tak berarah adalah jumlah sisi yang bersisian pada simpul. Pada graf berarah, derajat simpul v dinyatakan dengan $d_{in}(v)$ dan $d_{out}(v)$.
   $d_{in}(v)$ = derajat masuk = jumlah busur yang masuk ke simpul v.
   $d_{out}(v)$ = derajat keluar = jumlah busur yang keluar dari simpul v.
   $d(v) = d_{in}(v) + d_{out}(v)$
6. Lintasan
   Lintasan yang panjangnya 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, cukup dituliskan lintasan sebagai barisan dari simpul-simpul.
7. Siklus atau Sirkuit
   Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.
8. Terhubung
   Dua buah simpul v_i dan v_j dikatakan terhubung jika terdapat lintasan dari v_i ke v_j. Apabila setiap pasang simpul di dalam graf terhubung maka graf dikatakan graf terhubung. Graf berarah G dikatakan terhubung jika graf tak berarahnya terhubung.
9. Graf Berbobot
   Graf berbobot merupakan graf yang sering digunakan dalam pemecahan permasalahan. Graf berbobot memiliki suatu nilai pada setiap sisinya. Graf berbobot juga dapat memiliki arah yang disebut sebagai graf berarah berbobot.

2.2. Algoritma Djikstra

Algoritma Dijkstra ditemukan oleh Edsger Dijkstra. Algoritma ini merupakan sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah. Namun, algoritma ini juga dapat digunakan untuk graf tak berarah. Apabila simpul-simpul dari sebuah graf melambangkan kota-kota dan bobot sisi melambangkan jarak antara kota-kota tersebut, maka algoritma Dijkstra dapat digunakan untuk menentukan jarak terpendek antara dua kota.

Prinsip rakus (*greedy*) pada algoritma ini menyatakan bahwa pada setiap langkah memilih sisi yang berbobot minimum dan memasukkan ke dalam himpunan solusi. Input algoritma ini adalah sebuah graf berarah berbobot G yang diimplementasikan dalam matriks ketetanggaan dan sebuah sumber simpul (simpul asal) dalam G.

Salah satu contoh implementasi algoritma ini dalam *pseudocode* adalah sebagai berikut.

```
procedure ADjikstra (input m : matriks, a : simpul awal)
Deklarasi Variabel
s1, s2, ..., sn : integer
d1, d2, ..., dn : integer
i : integer
Algoritma
{langkah 0}
for i ← 1 to n do
    si ← 0
    di ← mai
endfor
{langkah 1}
sa ← 1 {simpul awal terpilih menjadi lintasan terpendek}
da ← ∞ {tidak ada lintasan terpendek simpul a ke a}
{langkah 2, 3, 4, ..., n-1}
for i ← 2 to n-1 do
    cari j sehingga sj = 0 dan dj = min{d1, d2, d3, ..., dn}
    sj ← 1 {simpul j terpilih menjadi lintasan terpendek}
    perbaharui di, untuk i = 1, 2, 3, ..., n dengan: di (baru) =
min {di (lama), dj + mji}
endfor
```

2.3. Algoritma Prim

Algoritma Prim adalah algoritma yang digunakan untuk mencari pohon merentang minimum dari suatu graf. Untuk setiap langkah, diambil sisi dari graf G yang mempunyai bobot minimum namun terhubung dengan pohon minimum yang telah terbentuk. Adapun karena graf

yang digunakan adalah graf berarah, maka pada saat pengambilan sisi (busur) dari graf G juga harus dipastikan yang terhubung dengan pohon minimum adalah pangkal dari busur. Pada makalah ini, algoritma Prim yang digunakan dimodifikasi untuk mencari upagraf dari pohon merentang yang menghubungkan simpul awal dan simpul tujuan saja kemudian dicari lintasan busur yang memiliki total bobot yang sama dengan hasil perhitungan bobot minimum pada algoritma Djikstra.

III. METODOLOGI

3.1. Batasan-Batasan Penelitian

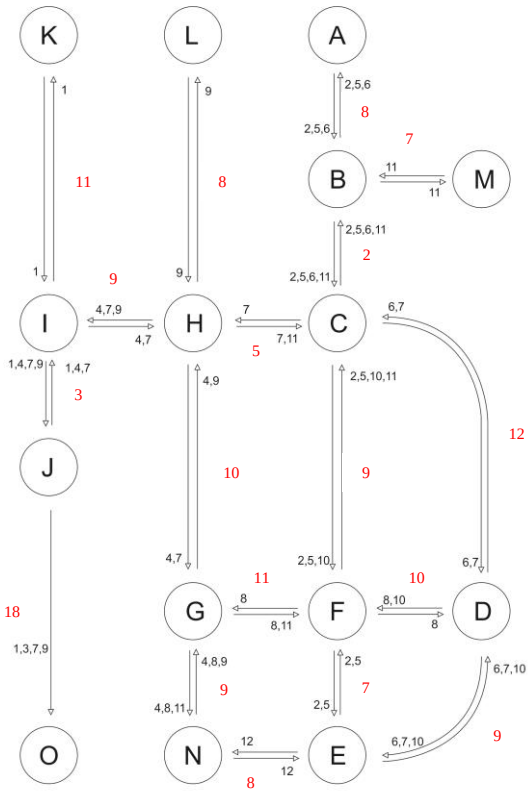
Beberapa batasan yang ada di dalam penentuan rute angkot tercepat dengan meggunakan teori graf, algoritma Djikstra, dan algoritma Prim adalah:

1. Variabel kemacetan diabaikan sehingga jarak berbanding lurus dengan waktu tempuh.
2. Rute tercepat diutamakan dibandingkan dengan jumlah transit yang dilakukan.
3. Frekuensi kemunculan angkot diasumsikan seragam.
4. Untuk lintasan angkot-angkot yang memiliki bobot yang sama, harus dipilih jumlah transit yang paling minimum dengan memprioritaskan memilih angkot yang sama.

3.1. Langkah-Langkah Penentuan Rute Tercepat

Berikut ini adalah langkah yang dilakukan untuk menentukan rute angkot tercepat di daerah Bandung.

1. Mencari denah rute angkot yang ada di kawasan Bandung.
2. Menerjemahkan denah tersebut ke dalam bentuk graf berarah berbobot. Simpul pada graf melambangkan lokasi-lokasi yang dilewati oleh angkot. Pangkal busur graf melambangkan lokasi awal angkot dan ujung pada busur graf melambangkan lokasi tujuan atau lokasi yang dilewati angkot. Bobot pada graf melambangkan jarak angkot antara kedua lokasi (simpul). Bobot dituliskan dengan angka berwarna merah.



Gambar 6. Graf berarah berbobot yang melambangkan rute angkot di suatu kawasan Bandung.

Keterangan:

Lokasi		Jenis Angkot	
A	Dago Atas	1	St.Hall – Ciumbuleuit
B	Persimpangan Tubagus Ismail – H Djuanda	2	Kelapa – Dago
C	Simpang Dago	3	Kelapa – Ledeng
D	Monumen Perjuangan Dipatiukur	4	Cicaheum – Ledeng
E	Persimpangan H Djuanda – Pasupati	5	Dago – St.Hall
F	Persimpangan Ganeca – H Djuanda	6	Dago – Riung Bandung
G	Persimpangan Ganeca – Taman Sari	7	Cicaheum – Ciroyom
H	Persimpangan Cisitu – Taman Sari - Siliwangi	8	Panghegar Permai – Dipatiukur
I	Persimpangan Ciumbuleuit – Cihampelas - Siliwangi	9	Cisitu – Tegallega
J	Persimpangan Setiabudhi – Cihampelas	10	Simpang Dago – Gede Bage
K	Ciumbuleuit	11	Sadang Serang – Caringin
L	Cisitu	12	Ciroyom – Ciburial
M	Tubagus Ismail		
N	Persimpangan Kebun Binatang – Pasupati		
O	Persimpangan Cihampelas – Pasupati		

3. Membuat matriks ketetanggaan dari graf berbobot pada gambar 6.

	A	B	C	D	E	F	G
A	0	8	∞	∞	∞	∞	∞
B	8	0	2	∞	∞	∞	∞
C	∞	2	0	12	∞	9	∞
D	∞	∞	12	0	9	10	∞
E	∞	∞	∞	9	0	7	∞
F	∞	∞	9	10	7	0	11
G	∞	∞	∞	∞	∞	11	0
H	∞	∞	5	∞	∞	∞	10
I	∞	∞	∞	∞	∞	∞	∞
J	∞	∞	∞	∞	∞	∞	∞
K	∞	∞	∞	∞	∞	∞	∞
L	∞	∞	∞	∞	∞	∞	∞
M	∞	7	∞	∞	∞	∞	∞
N	∞	∞	∞	∞	8	∞	9
O	∞	∞	∞	∞	∞	∞	∞

	H	I	J	K	L	M	N	O
A	∞	∞	∞	∞	∞	∞	∞	∞
B	∞	∞	∞	∞	∞	7	∞	∞
C	5	∞	∞	∞	∞	∞	∞	∞
D	∞	∞	∞	∞	∞	∞	∞	∞
E	∞	∞	∞	∞	∞	∞	8	∞
F	∞	∞	∞	∞	∞	∞	∞	∞
G	10	∞	∞	∞	∞	∞	9	∞
H	0	9	∞	∞	8	∞	∞	∞
I	9	0	3	11	∞	∞	∞	∞
J	∞	3	0	∞	∞	∞	∞	18
K	∞	11	∞	0	∞	∞	∞	∞
L	8	∞	∞	∞	0	∞	∞	∞
M	∞	∞	∞	∞	∞	0	∞	∞
N	∞	∞	∞	∞	∞	∞	0	∞
O	∞	∞	∞	∞	∞	∞	∞	0

Gambar 7. Matriks ketetanggaan dari graf berarah berbobot pada gambar 6.

Keterangan:

a. Kolom pertama pada matriks melambangkan simpul asal, sedangkan baris pertama pada matriks melambangkan simpul tujuan.
b. Simpul yang tidak dapat dicapai dari suatu simpul tertentu dilambangkan dengan simbol ∞ (*infinity*).
c. Jarak suatu simpul dengan dirinya sendiri adalah 0 (nol).

4. Menentukan lokasi awal keberangkatan.
5. Menerapkan algoritma Djikstra dan algoritma Prim dengan beberapa modifikasi pada matriks ketetanggaan untuk menentukan rute tercepat dari lokasi awal (simpul awal) ke lokasi tujuan (simpul terminal). Algoritma ini lebih lanjut akan langsung diaplikasikan pada upabab berikutnya.

IV. Studi Kasus Rute Angkot

Pada bab ini akan dibahas mengenai langkah-langkah pengaplikasian algoritma Djikstra dan algoritma Prim dalam menentukan rute angkot tercepat. Asumsi ada seorang penumpang yang ingin menaiki angkot dari Persimpangan Kebun Binatang – Pasupati. Penumpang ini ingin mengetahui angkot apa saja yang harus ia naiki agar dapat sampai ke Tubagus Ismail dengan rute tercepat. Selain itu, pada saat yang sama, ia juga ingin mengetahui angkot apa saja yang harus dinaiki agar dapat sampai ke Cisitu dengan rute tercepat. Berikut ini adalah penjabaran pengaplikasian algoritma Djikstra dan algoritma Prim dalam menentukan rute angkot tercepat.

1. Pada graf, Persimpangan Kebun Binatang – Pasupati dilambangkan dengan simpul N. Baris simpul N yang terdapat pada matriks ketetanggaan disalin. Simpul N dipilih sebagai simpul awal.

	A	B	C	D	E	F	G	
N	∞	∞	∞	∞	8	∞	9	
	H	I	J	K	L	M	N	O
N	∞	∞	∞	∞	∞	∞	0	∞

Simpul yang telah dipilih diberi tanda agar tidak dipilih kembali.

2. Memilih simpul terpendek yang dituju oleh simpul N dan belum pernah dipilih sebelumnya, yaitu simpul E.

	A	B	C	D	E	F	G	
N	∞	∞	∞	∞	8	∞	9	
	H	I	J	K	L	M	N	O
N	∞	∞	∞	∞	∞	∞	0	∞

3. Melakukan perbandingan sebagai berikut:
   A = jarak simpul N ke suatu simpul (diperiksa simpul A – O)
   B = jarak simpul N ke E + jarak simpul E ke suatu simpul (diperiksa simpul A – O)
   Apabila nilai B lebih kecil daripada nilai A, maka nilai yang ada pada matriks ketetanggaan diperbaharui dengan B.

	A	B	C	D	E	F	G	
N	∞	∞	∞	17	8	15	9	
	H	I	J	K	L	M	N	O
N	∞	∞	∞	∞	∞	∞	0	∞

4. Mengulangi langkah 2 dan 3 terus-menerus hingga semua simpul telah dilewati. Namun, harus diingat bahwa untuk simpul yang telah dipilih, tidak boleh dipilih kembali.

	A	B	C	D	E	F	G	
N	∞	∞	∞	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	∞	∞	∞	∞	∞	0	∞

	A	B	C	D	E	F	G	
N	∞	∞	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	∞	∞	∞	∞	∞	0	∞

	A	B	C	D	E	F	G	
N	∞	∞	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	∞	∞	∞	∞	∞	0	∞

	A	B	C	D	E	F	G	
N	∞	∞	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	∞	∞	27	∞	0	∞

	A	B	C	D	E	F	G	
N	∞	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	∞	∞	27	∞	0	∞

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	∞	∞	27	33	0	∞

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	∞	∞	27	33	0	∞

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	31	39	27	33	0	∞

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	31	39	27	33	0	49

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	31	39	27	33	0	49

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	

	H	I	J	K	L	M	N	O
N	19	28	31	39	27	33	0	49

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	31	39	27	33	0	49

	A	B	C	D	E	F	G	
N	34	26	24	17	8	15	9	
	H	I	J	K	L	M	N	O
N	19	28	31	39	27	33	0	49

Matriks terakhir yang dihasilkan merupakan matriks yang merepresentasikan bobot rute angkot tercepat dari N menuju ke setiap lokasi (simpul). Pada matriks diketahui bahwa untuk menuju ke Tubagus Ismail yang dilambangkan dengan simpul M, maka bobot terkecil yang dilalui adalah 33. Sedangkan untuk menuju ke Cisitu yang dilambangkan dengan simpul L, bobot terkecil yang dilalui adalah 27.

5. Dengan menggunakan algoritma Prim yang telah dibahas pada landasan teori, kita dapat memilih rute terpendek dari simpul N ke simpul M dan rute terpendek dari simpul N ke simpul L. Adapun dalam memilih rute ini harus diperhatikan jumlah bobot lintasannya harus sesuai dengan jumlah bobot yang didapatkan dari hasil pengaplikasian algoritma Djikstra. Dengan demikian, rute terpendek yang dilalui dari simpul N ke M adalah N – E – F – C – B – M. Sedangkan, rute terpendek yang dilalui dari simpul N ke L adalah N – G – H – L.

6. Melalui informasi pada graf gambar 6, maka dapat diketahui angkot apa saja yang harus dinaiki untuk melewati rute N – E – F – C – B – M dengan jumlah transit yang paling minimum. Angkot-angkot tersebut adalah (12) Ciroyom – Ciburial, (2) Kelapa – Dago, dan (11) Sadang Serang – Caringin. Selain itu, dapat juga dipilih angkot alternatif lain, yaitu: (12) Ciroyom – Ciburial, (5) Dago – St. Hall, dan (11) Sadang Serang – Caringin. Sedangkan, angkot yang melewati rute N – G – H – L dengan jumlah transit yang minimum adalah (9) Cisitu – Tegallega dengan tidak perlu melakukan transit sama sekali karena hanya 1 angkot saja.

V. KESIMPULAN

Dalam menentukan rute angkot tercepat dari suatu lokasi awal ke lokasi tujuan, dapat diaplikasikan konsep graf, algoritma Djikstra, dan algoritma Prim yang telah dimodifikasi. Algoritma ini dimodifikasi menyesuaikan dengan bentuk graf yang digunakan untuk

merepresentasikan rute angkot yaitu graf berarah berbobot. Hal ini dapat memecahkan permasalahan mengenai ketidakefektifan dalam pemilihan angkot untuk menuju ke lokasi tujuan.

DAFTAR PUSTAKA

[1] Rosen, Kenneth H., 2012. *Discrete Mathematics and Its Application. 7th Edition*. New York: McGraw-Hill.

[2] Rinaldi Munir. 2006. *Diktat Kuliah IF2120: Matematika Diskrit*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

[3] Cormen, Thomas H., dkk. 2001. *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill.

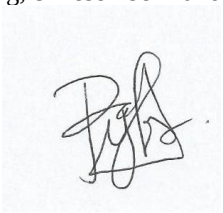
[4] Wira Setiawan. 2015. *Tentang Algoritma Djikstra*. https://wirasetiawan29.wordpress.com/2015/04/02/tentang-algoritma-djikstra/, 8 Desember 2016.

[5] TransportasiUmum. 2016. *Angkutan kota: Angkot Kota Bandung*. transportasiumum.com/content/rute-angkot-bandung/, 4 Desember 2016.

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, 8 Desember 2016



Rachel Sidney Devianti/13515124