

METODE *TRANSITIVE CLOSURE* SEBAGAI SOLUSI PENCARIAN JALUR TERPENDEK (*SHORTEST PATH*)

Dzurotul Mutimmah¹

¹ Prodi Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas PGRI Banyuwangi
Jl. Ikan Tongkol, No. 22, Kertosari, Banyuwangi

E-mail: dzurotulmutimmah69@gmail.com

ABSTRAK

Latar Belakang: Tempat wisata merupakan salah satu tempat tujuan destinasi bagi para wisatawan, salah satunya Pulau Bali. Namun ada beberapa kendala yang dihadapi oleh para wisatawan, yaitu melambungnya harga bahan bakar minyak (BBM). Selain kelangkaan BBM, efisiensi waktu juga menjadi salah satu kendala bagi para wisatawan. Seringkali para wisatawan tidak dapat memaksimalkan waktu kunjungannya disebabkan oleh ketidaktahuannya tentang rute jalan tempat wisata tersebut sehingga ada beberapa tempat wisata yang semula ingin di kunjungi menjadi batal untuk dikunjungi. Pencarian rute terpendek merupakan salah satu cara untuk menanggulangi kendala-kendala tersebut. Ada beberapa metode yang dapat digunakan dalam pencarian rute terpendek, salah satunya yaitu metode *transitive closure*.

Metode: Pada penelitian ini akan dijelaskan prosedur untuk memperoleh rute terpendek dari satu tempat wisata ke tempat wisata lain dengan studi kasus peta wisata Pulau Bali dengan mengambil delapan tempat wisata sebagai contoh objek untuk pencarian jalur terpendek menggunakan metode *transitive closure* yang direpresentasikan ke dalam sebuah graf.

Hasil: Pencarian rute terpendek menggunakan metode *transitive closure* pada beberapa tempat wisata di Pulau Bali dapat menghasilkan jarak yang paling minimum.

Kesimpulan: Berdasarkan hasil penghitungan menggunakan metode *transitive closure*, didapat jarak minimum dari tempat wisata Lake Batur (v_5) ke Uluwatu (v_8), yaitu sejauh 100 km.

Kata Kunci: Transitive Closure, Rute Terpendek, Graf

ABSTRACT

Background: Tourist attractions are one of the destinations for tourists, one of which is the Bali Island. However, there are several obstacles faced by tourists, namely the soaring price of fuel oil. In addition to the scarcity of fuel, time efficiency is also one of the obstacles for tourists. Often tourists cannot maximize the time of their visit due to their ignorance of the route of the tourists attractions so that there are several tourists attractions that they originally wanted to visit become canceled to be visited. Finding the shortest route is one way to overcome these obstacles. There are several methods that can be used in finding the shortest path, one of which is the *transitive closure* method.

Method: in this study, the procedure for obtaining the shortest route from one tourist place to another will be explain with a case study of the Bali tourist map by taking eight tourist attractions as examples of objects for finding the shortest path using the *transitive closure* method which is represented in a graph.

Result: Searching for the shortest path using the *transitive closure* method at several tourist attractions on the Bali Island can produce the minimum distance.

Conclusion: Based on the result of calculations using the *transitive closure* method, the minimum distance from Lake Batur to Uluwatu is 100 km.

Key words: Transitive Closure, Shortest Path, Graph

PENDAHULUAN

Tata kota suatu negara, semakin hari semakin berkembang. Hal ini merupakan salah satu upaya pemerintah kota untuk memperindah daerah-daerah yang dikelolanya agar lebih nyaman, aman, dan terjaga kebersihannya. Selain itu, apabila tata kota tersebut tetap terjaga keindahannya, maka para wisatawan (wisatawan lokal maupun wisatawan manca negara) akan tertarik untuk memilih kota tersebut sebagai salah satu sarana untuk berlibur. Terutama bagi pulau-pulau di Indonesia yang mempunyai tempat-tempat wisata berpotensi tinggi, seperti Pulau Bali. Tempat wisata tersebut merupakan salah satu sarana yang menghasilkan devisa cukup besar bagi suatu negara.

Namun ada beberapa kendala yang dihadapi oleh para wisatawan, yaitu melambungnya harga bahan bakar minyak (BBM). Kenaikan harga BBM yang terus meningkat disebabkan oleh kelangkaan BBM yang semakin lama semakin berkurang. Kelangkaan BBM ini merupakan salah satu masalah terbesar yang dialami oleh seluruh negara pada era globalisasi seperti sekarang ini. Para pengunjung harus mengeluarkan uang yang tidak sedikit untuk membeli BBM agar dapat berkunjung ke tempat-tempat wisata tujuannya. Terlebih lagi jika para pengunjung belum pernah mengunjungi daerah tersebut, yang mengharuskan pengunjung untuk mencari rute jalan tempat yang ingin dituju.

Selain kelangkaan BBM, efisiensi waktu juga menjadi salah satu kendala bagi para wisatawan. Seringkali para wisatawan tidak dapat memaksimalkan waktu kunjungannya disebabkan oleh ketidaktahuannya tentang rute jalan

tempat wisata tersebut sehingga ada beberapa tempat wisata yang semula ingin di kunjungi menjadi batal untuk dikunjungi.

Pencarian rute terpendek merupakan salah satu cara untuk menanggulangi kendala-kendala tersebut. Permasalahan rute terpendek merupakan sebuah permasalahan dalam menemukan lintasan antara dua buah simpul pada graf berbobot yang memiliki gabungan nilai dari jumlah bobot pada sisi graf yang dilewati dengan jumlah paling minimum (Nawagusti et al., 2018). Ada beberapa metode yang dapat digunakan dalam pencarian rute terpendek, salah satunya yaitu metode *transitive closure*. Metode ini mempunyai kemampuan untuk memberikan alternatif jalur/rute yang dapat memprediksi kemacetan yang ada pada waktu tertentu (Nurhayatie, 2007). Dengan metode ini, dapat dicari rute terpendek berdasarkan jarak antar kota sehingga para wisatawan bisa mengefisiensikan waktunya dan dapat mengurangi biaya belanja BBM. Selain itu, penggunaan metode ini selalu dapat menemukan rute optimal selama ada jalur dan rute yang didapat juga stabil (Desiree & Handojo, 2008).

Berkenaan dengan latar belakang di atas, maka permasalahan yang timbul adalah rute manakah yang akan dipilih dan ditempuh dengan memperhatikan jarak terpendek menggunakan metode *transitive closure*.

METODE

Pada penelitian ini akan dijelaskan prosedur untuk memperoleh rute terpendek dari satu tempat wisata ke tempat wisata lain dengan studi kasus peta wisata Pulau Bali dengan

mengambil delapan tempat wisata sebagai contoh objek untuk pencarian jalur terpendek menggunakan metode *transitive closure* yang direpresentasikan ke dalam sebuah graf. Seringkali graf digunakan untuk merepresentasikan suatu jaringan. Misalkan jaringan jalan raya dimodelkan graf dengan kota sebagai simpul (*vertex/node*) dan jalan yang menghubungkan setiap kotanya sebagai sisi (*edge*) yang bobotnya (*weight*) adalah panjang dari jalan tersebut (Hayati & Yohanes, 2014). Dalam penelitian ini, simpul menyatakan tempat wisata yang ada di Pulau Bali, sisi menyatakan jalan yang menghubungkan tempat wisata satu dengan tempat wisata lainnya, dan bobot menyatakan jarak antar tempat wisata.

Adapun langkah-langkah pencarian rute terpendek dengan menggunakan metode *transitive closure* yaitu :

1. Merepresentasikan beberapa tempat wisata yang ada pada peta Pulau Bali ke dalam *directed graph*.
2. Merepresentasikan *directed graph* ke dalam matriks *adjacency* yang merupakan matriks derajat satu dari *directed graph* tersebut.

$$M = \begin{bmatrix} (\alpha_{11}, m_{11}) & (\alpha_{12}, m_{12}) & \dots & (\alpha_{1n}, m_{1n}) \\ (\alpha_{21}, m_{21}) & (\alpha_{22}, m_{22}) & \dots & (\alpha_{2n}, m_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_{n1}, m_{n1}) & (\alpha_{n2}, m_{n2}) & \dots & (\alpha_{nn}, m_{nn}) \end{bmatrix}$$

- $\alpha_{ij} \in \{0,1\}$ adalah parameter untuk

menyatakan ada tidaknya jalan dari v_i ke v_j , dimana jika $\alpha_{ij} = 1$ maka ada jalan dari v_i ke v_j . Sebaliknya, jika $\alpha_{ij} = 0$ maka tidak ada jalan dari v_i ke v_j .

- m_{ij} menyatakan bobot (*weight*) dari v_i ke v_j yang merupakan jarak dari suatu daerah ke daerah lain.
3. Mencari matriks derajat dua dengan cara mengoperasikan matriks derajat satu dengan matriks derajat satu lainnya.

$$\begin{bmatrix} (\alpha_{11}, m_{11}) & (\alpha_{12}, m_{12}) & \dots & (\alpha_{1n}, m_{1n}) \\ (\alpha_{21}, m_{21}) & (\alpha_{22}, m_{22}) & \dots & (\alpha_{2n}, m_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_{n1}, m_{n1}) & (\alpha_{n2}, m_{n2}) & \dots & (\alpha_{nn}, m_{nn}) \end{bmatrix} \circ \begin{bmatrix} (\alpha_{11}, m_{11}) & (\alpha_{12}, m_{12}) & \dots & (\alpha_{1n}, m_{1n}) \\ (\alpha_{21}, m_{21}) & (\alpha_{22}, m_{22}) & \dots & (\alpha_{2n}, m_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_{n1}, m_{n1}) & (\alpha_{n2}, m_{n2}) & \dots & (\alpha_{nn}, m_{nn}) \end{bmatrix}$$

Proses pengoperasian matriks derajat satu dengan matriks derajat satu lainnya, yaitu sebagai berikut.

- a. Membandingkan baris pertama matriks derajat satu dengan kolom pertama matriks derajat satu lainnya.
- b. Menyimpan hasil proses (a) pada baris ke-1 dan kolom ke-1 matriks derajat dua dengan ketentuan :

$$\alpha_{ij}^2 = \max_{k=1}^n \{ \min(\alpha_{ik}, \alpha_{kj}) \},$$

$$\alpha_{ij} \in \{0,1\}$$

$$m_{ij}^2 = \min_{p \in M} (m_{ip} + m_{pj}),$$

$$M = \{k \mid \min(\alpha_{ik}, \alpha_{kj}) = 1\}$$

- c. Melakukan proses (a) dan (b) sampai baris ke- n

matriks derajat satu dengan kolom ke- n matriks derajat satu lainnya.

- d. Menyimpan hasil proses (c) sesuai dengan baris dan kolom yang dibandingkan, misalnya membandingkan baris ke-4 matriks derajat satu dengan kolom ke-4 matriks derajat satu lainnya, maka hasilnya disimpan pada baris ke-4 dan kolom ke-4 matriks derajat dua.
4. Mencari matriks derajat tiga dengan cara mengoperasikan matriks derajat dua dengan matriks derajat satu.

$$\begin{bmatrix} (\alpha_{11}^2, m_{11}^2) & (\alpha_{12}^2, m_{12}^2) & \dots & (\alpha_{1n}^2, m_{1n}^2) \\ (\alpha_{21}^2, m_{21}^2) & (\alpha_{22}^2, m_{22}^2) & \dots & (\alpha_{2n}^2, m_{2n}^2) \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_{n1}^2, m_{n1}^2) & (\alpha_{n2}^2, m_{n2}^2) & \dots & (\alpha_{nn}^2, m_{nn}^2) \end{bmatrix} \circ \begin{bmatrix} (\alpha_{11}, m_{11}) & (\alpha_{12}, m_{12}) & \dots & (\alpha_{1n}, m_{1n}) \\ (\alpha_{21}, m_{21}) & (\alpha_{22}, m_{22}) & \dots & (\alpha_{2n}, m_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_{n1}, m_{n1}) & (\alpha_{n2}, m_{n2}) & \dots & (\alpha_{nn}, m_{nn}) \end{bmatrix}$$

Proses pengoperasian matriks derajat dua dengan matriks derajat satu, yaitu :

- a. Membandingkan baris pertama matriks derajat dua dengan kolom pertama matriks derajat satu.
- b. Menyimpan hasil proses (a) pada baris ke-1 dan kolom ke-1 matriks derajat tiga dengan ketentuan :

$$\alpha_{ij}^3 = \max_{k=1}^n \{ \min(\alpha_{ik}, \alpha_{kj}) \},$$

$$\alpha_{ij} \in \{0,1\}$$

$$m_{ij}^3 = \max_{p \in M} \{ \min(m_{ip}, m_{pj}) \},$$

$$M = \{k \mid \min(\alpha_{ik}, \alpha_{kj}) = 1\}$$

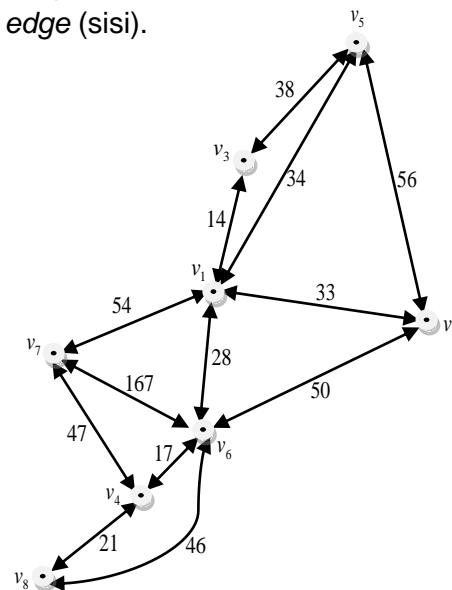
- c. Melakukan proses (a) dan (b) sampai baris ke- n matriks

derajat dua dengan kolom ke- n matriks derajat satu.

- d. Menyimpan hasil proses (c) sesuai dengan baris dan kolom yang dibandingkan, misalnya membandingkan baris ke-4 matriks derajat dua dengan kolom ke-4 matriks derajat satu, maka hasilnya disimpan pada baris ke-4 dan kolom ke-4 matriks derajat tiga.
5. Mencari matriks sampai derajat $n-1$, dimana n adalah jumlah titik pada *directed graph*.
6. Mengevaluasi matriks derajat satu sampai matriks derajat $n-1$ untuk memperoleh rute terpendek yang akan ditempuh.

HASIL DAN PEMBAHASAN

Pada penelitian ini, beberapa tempat wisata pada peta Pulau Bali ini direpresentasikan ke dalam *graph* (Gambar 1), dimana tempat-tempat wisata tersebut direpresentasikan kedalam sebuah *vertex* (titik), dan jalan yang menghubungkan tempat wisata direpresentasikan kedalam sebuah *edge* (sisi).



Gambar 1. Representasi beberapa tempat wisata di Pulau Bali pada *Directed Graph*

Beberapa tempat wisata pada peta Pulau Bali di representasikan ke dalam simbol-simbol (v_1, v_2, \dots, v_n) seperti dibawah ini.

- Goa Gajah (v_1)
- Goa Lawah (v_2)
- Gunung Kawi (v_3)
- Kuta (v_4)
- Lake Batur (v_5)
- Sanur (v_6)
- Tanah Lot (v_7)
- Uluwatu (v_8)

Sedangkan jalan yang menghubungkan daerah-daerah tersebut direpresentasikan kedalam *edge* $(v_1 \rightarrow v_2, v_1 \rightarrow v_3, \dots, v_1 \rightarrow v_n)$ seperti pada Tabel 1. Pada representasi graf tersebut, setiap jalan mempunyai dua arah. Hal ini dikarenakan jalan dari suatu tempat wisata ke tempat wisata lain merupakan jalan utama yang mempunyai dua arah.

Tabel 1. Relasi antar *vertex* dan jarak antar *vertex*

Relasi antar <i>vertex</i>	Jarak antar <i>vertex</i> (km)
$v_1 \leftrightarrow v_2$	33
$v_1 \leftrightarrow v_3$	14
$v_1 \leftrightarrow v_5$	34
$v_1 \leftrightarrow v_6$	28
$v_1 \leftrightarrow v_7$	54
$v_2 \leftrightarrow v_5$	56
$v_2 \leftrightarrow v_6$	50
$v_3 \leftrightarrow v_5$	38

$v_4 \leftrightarrow v_6$	17
$v_4 \leftrightarrow v_7$	47
$v_4 \leftrightarrow v_8$	21
$v_6 \leftrightarrow v_7$	167
$v_6 \leftrightarrow v_8$	46

Untuk mencari rute dari suatu tempat wisata ke tempat wisata lain yang terdapat pada Pulau Bali, maka diperlukan matriks *adjacency* yang merupakan matriks berderajat n , dimana derajat tersebut menyatakan ada tidaknya jalan dari suatu daerah ke daerah lainnya dengan melewati n buah jalan.. Matriks *adjacency* dari *graph* tersebut merupakan matriks 8×8 sesuai dengan jumlah *vertex* yang ada pada *graph* yang disebut sebagai matriks derajat satu, seperti dibawah ini.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(0,0)	(1,33)	(1,14)	(0,0)	(1,34)	(1,28)	(1,54)	(0,0)
v2	(1,33)	(0,0)	(0,0)	(0,0)	(1,56)	(1,50)	(0,0)	(0,0)
v3	(1,14)	(0,0)	(0,0)	(0,0)	(1,38)	(0,0)	(0,0)	(0,0)
v4	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,17)	(1,47)	(1,21)
v5	(1,34)	(1,56)	(1,38)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
v6	(1,28)	(1,50)	(0,0)	(1,17)	(0,0)	(0,0)	(1,167)	(1,46)
v7	(1,54)	(0,0)	(0,0)	(1,47)	(0,0)	(1,167)	(0,0)	(0,0)
v8	(0,0)	(0,0)	(0,0)	(1,21)	(0,0)	(1,46)	(0,0)	(0,0)

Matriks derajat dua diperoleh dari pengoperasian matriks derajat satu dengan matriks derajat satu lainnya. Setelah melakukan penghitungan, maka diperoleh matriks derajat dua seperti berikut.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(1,28)	(1,78)	(1,72)	(1,45)	(1,52)	(1,83)	(1,195)	(1,74)
v2	(1,78)	(1,66)	(1,47)	(1,67)	(1,67)	(1,61)	(1,87)	(1,96)
v3	(1,72)	(1,47)	(1,28)	(0,0)	(1,48)	(1,42)	(1,68)	(0,0)
v4	(1,45)	(1,67)	(0,0)	(1,34)	(0,0)	(1,67)	(1,184)	(1,63)
v5	(1,52)	(1,67)	(1,48)	(0,0)	(1,68)	(1,62)	(1,88)	(0,0)
v6	(1,83)	(1,61)	(1,42)	(1,67)	(1,62)	(1,34)	(1,64)	(1,38)
v7	(1,195)	(1,87)	(1,68)	(1,184)	(1,88)	(1,64)	(1,94)	(1,68)
v8	(1,74)	(1,96)	(0,0)	(1,63)	(0,0)	(1,38)	(1,68)	(1,92)

Matriks derajat tiga diperoleh dari pengoperasian matriks derajat dua dengan matriks derajat satu. Setelah melakukan penghitungan, maka diperoleh matriks derajat tiga seperti berikut.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(1,86)	(1,61)	(1,42)	(1,95)	(1,62)	(1,56)	(1,82)	(1,66)
v2	(1,61)	(1,111)	(1,92)	(1,78)	(1,85)	(1,84)	(1,114)	(1,88)
v3	(1,42)	(1,92)	(1,86)	(1,59)	(1,66)	(1,97)	(1,126)	(1,88)
v4	(1,95)	(1,78)	(1,59)	(1,84)	(1,79)	(1,51)	(1,81)	(1,55)
v5	(1,62)	(1,85)	(1,66)	(1,79)	(1,86)	(1,80)	(1,106)	(1,108)
v6	(1,56)	(1,84)	(1,97)	(1,51)	(1,80)	(1,84)	(1,114)	(1,80)
v7	(1,82)	(1,114)	(1,126)	(1,81)	(1,106)	(1,114)	(1,231)	(1,110)
v8	(1,66)	(1,88)	(1,88)	(1,55)	(1,108)	(1,80)	(1,110)	(1,84)

Matriks derajat empat diperoleh dari pengoperasian matriks derajat tiga dengan matriks derajat satu. Setelah melakukan penghitungan, maka diperoleh matriks derajat empat seperti berikut.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(1,56)	(1,106)	(1,100)	(1,73)	(1,80)	(1,111)	(1,140)	(1,102)
v2	(1,106)	(1,94)	(1,75)	(1,101)	(1,95)	(1,89)	(1,115)	(1,99)
v3	(1,100)	(1,75)	(1,56)	(1,109)	(1,76)	(1,70)	(1,96)	(1,80)
v4	(1,73)	(1,101)	(1,109)	(1,68)	(1,97)	(1,101)	(1,131)	(1,97)
v5	(1,80)	(1,95)	(1,76)	(1,97)	(1,96)	(1,90)	(1,116)	(1,100)
v6	(1,111)	(1,89)	(1,70)	(1,101)	(1,90)	(1,68)	(1,98)	(1,72)
v7	(1,140)	(1,115)	(1,96)	(1,131)	(1,116)	(1,98)	(1,128)	(1,102)
v8	(1,102)	(1,99)	(1,80)	(1,97)	(1,100)	(1,72)	(1,102)	(1,76)

Matriks derajat lima diperoleh dari pengoperasian matriks derajat empat dengan matriks derajat satu. Setelah melakukan penghitungan, maka diperoleh matriks derajat lima seperti berikut.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(1,114)	(1,89)	(1,70)	(1,123)	(1,90)	(1,84)	(1,110)	(1,94)
v2	(1,89)	(1,139)	(1,120)	(1,106)	(1,113)	(1,118)	(1,148)	(1,122)
v3	(1,70)	(1,120)	(1,114)	(1,87)	(1,94)	(1,125)	(1,154)	(1,116)
v4	(1,123)	(1,106)	(1,87)	(1,118)	(1,107)	(1,85)	(1,115)	(1,89)
v5	(1,90)	(1,113)	(1,94)	(1,107)	(1,114)	(1,108)	(1,134)	(1,118)
v6	(1,84)	(1,118)	(1,125)	(1,85)	(1,108)	(1,118)	(1,148)	(1,114)
v7	(1,110)	(1,148)	(1,154)	(1,115)	(1,134)	(1,148)	(1,178)	(1,144)
v8	(1,94)	(1,122)	(1,116)	(1,89)	(1,118)	(1,114)	(1,144)	(1,118)

Matriks derajat enam diperoleh dari pengoperasian matriks derajat lima dengan matriks derajat satu. Setelah melakukan penghitungan, maka diperoleh matriks derajat enam seperti berikut.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(1,84)	(1,134)	(1,128)	(1,101)	(1,108)	(1,139)	(1,168)	(1,130)
v2	(1,134)	(1,122)	(1,103)	(1,135)	(1,123)	(1,117)	(1,143)	(1,127)
v3	(1,128)	(1,103)	(1,84)	(1,137)	(1,104)	(1,98)	(1,124)	(1,108)
v4	(1,101)	(1,135)	(1,137)	(1,102)	(1,125)	(1,135)	(1,165)	(1,131)
v5	(1,108)	(1,123)	(1,104)	(1,125)	(1,124)	(1,118)	(1,144)	(1,128)
v6	(1,139)	(1,117)	(1,98)	(1,135)	(1,118)	(1,102)	(1,132)	(1,106)
v7	(1,168)	(1,143)	(1,124)	(1,165)	(1,144)	(1,132)	(1,162)	(1,136)
v8	(1,130)	(1,127)	(1,108)	(1,131)	(1,128)	(1,106)	(1,136)	(1,110)

Matriks derajat tujuh diperoleh dari pengoperasian matriks derajat enam dengan matriks derajat satu. Setelah melakukan penghitungan, maka

diperoleh matriks derajat tujuh seperti berikut.

	v1	v2	v3	v4	v5	v6	v7	v8
v1	(1,142)	(1,117)	(1,98)	(1,151)	(1,118)	(1,112)	(1,138)	(1,122)
v2	(1,117)	(1,167)	(1,148)	(1,134)	(1,141)	(1,152)	(1,182)	(1,156)
v3	(1,98)	(1,148)	(1,142)	(1,115)	(1,122)	(1,153)	(1,182)	(1,144)
v4	(1,151)	(1,134)	(1,115)	(1,152)	(1,135)	(1,119)	(1,149)	(1,123)
v5	(1,118)	(1,141)	(1,122)	(1,135)	(1,142)	(1,136)	(1,162)	(1,146)
v6	(1,112)	(1,152)	(1,153)	(1,119)	(1,136)	(1,152)	(1,182)	(1,148)
v7	(1,138)	(1,182)	(1,182)	(1,149)	(1,162)	(1,182)	(1,212)	(1,178)
v8	(1,122)	(1,156)	(1,144)	(1,123)	(1,146)	(1,148)	(1,178)	(1,152)

Jadi, untuk memperoleh matriks berderajat n , maka proses pengoperasiannya yaitu sampai matriks berderajat $n-1$ dengan matriks berderajat satu. Setelah matriks derajat satu sampai matriks derajat tujuh dihitung, maka jalur terpendek bisa ditentukan. Misalnya, akan di cari jalur terpendek dari Lake Batur (v_3) ke Uluwatu (v_8), maka semua matriks yang ada dari matriks derajat satu sampai matriks derajat tujuh tersebut dievaluasi untuk mendapatkan matriks *closure* dimana merupakan hasil terpendek dari semua matriks yang telah dibentuk.

Tabel 2. Jarak Terpendek Dari Setiap Derajat Matriks

Derajat Matriks	Jarak Terpendek (km)
1	Tidak terdapat jalur
2	Tidak terdapat jalur
3	108
4	100
5	118
6	128
7	146

Dari hasil yang telah diperoleh di atas, matriks berderajat n harus dicari hanya sampai matriks berderajat tujuh. Hal ini dikarenakan $n-1=8-1=7$, dimana n jumlah *vertex* pada *directeted graph* (Gambar 1). Pada matriks derajat satu dan matriks derajat dua dapat diketahui bahwa tidak ada jalur dari v_5 menuju v_8 karena nilai $\alpha_{ij} = 0$. Pada

matriks derajat tiga dapat diketahui bahwa ada jalur dari v_5 menuju v_8 dengan melewati tiga buah jalan, yaitu sejauh 108 km dengan rute perjalanan $v_5 \rightarrow v_1 \rightarrow v_6 \rightarrow v_8$. Pada matriks derajat empat, diketahui bahwa ada jalur dari v_5 menuju v_8 dengan melewati empat buah jalan, yaitu sejauh 100 km dengan rute perjalanan $v_5 \rightarrow v_1 \rightarrow v_6 \rightarrow v_4 \rightarrow v_8$. Pada matriks derajat lima, diketahui bahwa ada jalur dari v_5 menuju v_8 dengan melewati lima buah jalan, yaitu sejauh 118 km dengan rute perjalanan $v_5 \rightarrow v_3 \rightarrow v_1 \rightarrow v_6 \rightarrow v_4 \rightarrow v_8$. Pada matriks derajat enam, diketahui bahwa ada jalur dari v_5 menuju v_8 dengan melewati lima buah jalan, yaitu sejauh 128 km dan pada matriks derajat tujuh, diketahui bahwa ada jalur dari v_5 menuju v_8 dengan melewati lima buah jalan, yaitu sejauh 146 km.

Dari matriks derajat satu sampai matriks derajat tujuh diketahui bahwa jarak terpendek dari Lake Batur (v_5) ke Uluwatu (v_8) yaitu 100 km yang dihasilkan pada matriks derajat empat. Hal ini dapat diartikan bahwa dari Lake Batur (v_5) ke Uluwatu (v_8), jalan terpendek yang dapat ditempuh yaitu melewati empat buah jalan yaitu $v_5 \rightarrow v_1 \rightarrow v_6 \rightarrow v_4 \rightarrow v_8$.

KESIMPULAN

Berdasarkan hasil dan pembahasan, dapat disimpulkan bahwa metode *transitive closure* dapat digunakan dalam pencarian rute terpendek, dimana pada subbab pembahasan di berikan contoh pencarian jalur terpendek dari Lake Batur (v_5) ke Uluwatu (v_8) dan

didapat jalur terpendeknya yaitu 100 km dengan rute $v_5 \rightarrow v_1 \rightarrow v_6 \rightarrow v_4 \rightarrow v_8$.

DAFTAR PUSTAKA

- Desiree, F. J., & Handojo, A. (2008). *Aplikasi Pencarian Rute Optimal Menggunakan Metode Transitive Closure*. *Kommit*, 20–21.
- Hayati, E. N., & Yohanes, A. (2014). Pencarian Rute Terpendek Menggunakan Algoritma Greedy. *Seminar Nasional IENACO*, 2337–4349.
- Nawagusti, V. A., Nurdin, A., & Aryanti, aryanti. (2018). Penentuan Rute Terpendek Pada Optimalisasi Jalur Pendistribusian Barang Di Pt. X Dengan Menerapkan Algoritma Floyd-Warshall. *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri*, 57–64.
- Nurhayatie, R. (2007). *Pencarian Rute Tercepat Pada Biro Perjalanan Menggunakan Algoritma Transitive Closure*. <http://repository.ub.ac.id/151633/1/050702103.pdf>