

Penerapan Algoritma Prim dalam Perancangan Rute Bus Wisata di Kabupaten Karanganyar

Okugata Fahmi Nurul Yudho Fauzan - 13518031¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13518031@std.stei.itb.ac.id

Abstraksi—Kabupaten Karanganyar merupakan salah satu kabupaten yang berada di Jawa Tengah yang memiliki berbagai destinasi wisata. Tak dapat dipungkiri bahwa banyak wisatawan yang mengunjungi kabupaten tersebut untuk memanfaatkan liburan. Namun, banyaknya pengunjung yang datang dengan menggunakan kendaraan pribadi sering menyebabkan kemacetan. Dengan adanya bus wisata yang menghubungkan antara daerah wisata mungkin dapat mengurangi kemacetan tersebut. Oleh karena itu, diperlukan strategi dalam menentukan rute bus wisata agar total jarak rute yang ditempuh minimum sehingga dapat mempercepat durasi tempuh ke daerah wisata. Dengan algoritma Prim, perencanaan rute dapat dibentuk dengan sebelumnya membuat graf berbobot berdasarkan jarak antar tiap destinasi wisata.

Kata Kunci—daerah wisata, liburan, bus wisata, graf, pohon merentang minimum, algoritma Prim.

I. PENDAHULUAN

Kabupaten Karanganyar merupakan salah satu kabupaten yang berada di Jawa Tengah. Letaknya berada sekitar 20 km ke timur dari Kota Surakarta. Posisinya yang berada pada kaki Gunung Lawu membuat Kabupaten Karanganyar memiliki berbagai destinasi wisata. Daerah yang dibatasi garis merah pada Gambar 1 berikut merupakan daerah dari Kabupaten Karanganyar.



Gambar 1. Daerah Kabupaten Karanganyar

Sumber: <https://www.google.com/maps>

Berdasarkan data dari Dinas Pariwisata Pemuda dan Olahraga Kabupaten Karanganyar [2], pada tahun 2017, total ada 1.087.528 wisatawan yang berkunjung ke daerah wisata di

Kabupaten Karanganyar. Kunjungan destinasi wisata tersebut biasanya meningkat di kala hari-hari libur sekolah. Banyak masyarakat pergi ke daerah wisata dengan menggunakan kendaraan pribadi. Hal itu membuat jalan-jalan yang menuju destinasi wisata tersebut menjadi sangat padat hingga menyebabkan kemacetan.

Salah satu cara untuk mengatasi kemacetan tersebut menurut penulis ialah dengan cara membuat moda transportasi umum yang dapat menghubungkan setiap daerah destinasi wisata tersebut, contohnya ialah seperti bus wisata. Dengan adanya bus wisata, para pengunjung dapat menggunakan bus tersebut sehingga dapat mengurangi jumlah kendaraan pribadi yang digunakan untuk menuju daerah wisata.

Untuk menentukan rute bus wisata agar dapat meminimalkan jarak tempuh, penulis akan menerapkan algoritma Prim. Algoritma ini memungkinkan pembuatan sebuah pohon merentang minimum yang dapat digunakan untuk pembuatan rute bus wisata tersebut. Bus wisata tersebut akan menghubungkan beberapa destinasi wisata yang populer di Kabupaten Karanganyar dan juga beberapa terminal bus yang ada.

II. LANDASAN TEORI

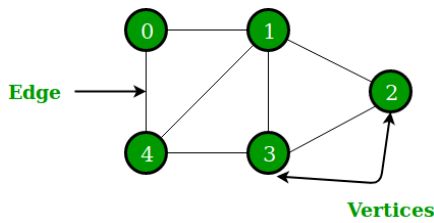
A. Graf

Graf adalah pokok bahasan yang sudah ada sejak lama, tetapi sampai sekarang masih ada banyak sekali penerapannya. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek lainnya.

1. Definisi Graf [1]

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini adalah V himpunan tidak kosong dari simpul-simpul dan E adalah himpunan sisi yang menghubungkan sepasang simpul.

Simpul-simpul dalam graf dapat diberi nomor dengan huruf, bilangan asli, atau gabungan dari keduanya. Sisi dalam graf yang menghubungkan simpul u dan v dapat dinyatakan sebagai pasangan (u, v) atau sebagai lambang seperti e_1 dan e_2 . Secara geometri, graf dapat digambarkan sebagai sekumpulan noktah (simpul) yang dihubungkan oleh sekumpulan garis (sisi).



Gambar 2. Contoh graf sederhana
 Sumber: <https://www.geeksforgeeks.org/wp-content/uploads/undirectedgraph.png>

2. Jenis-jenis Graf [1]

Graf dapat dibedakan jenisnya menurut klasifikasi tertentu, yaitu berdasarkan ada tidaknya gelang/ sisi ganda dan ada tidaknya orientasi arah pada graf.

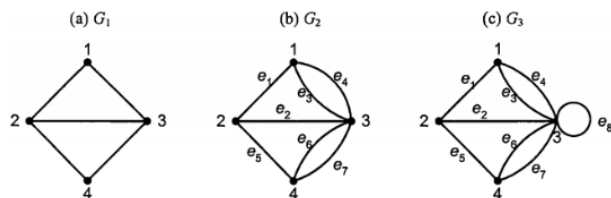
Berdasarkan ada tidaknya gelang/ sisi ganda, graf dapat dibedakan menjadi 2 jenis, yaitu:

a. Graf sederhana

Graf sederhana merupakan graf yang tidak mengandung gelang maupun sisi ganda. Pada graf ini, sisi adalah pasangan tak terurut, sehingga menuliskan sisi (u, v) sama saja dengan (v, u) . Graf G_1 pada Gambar 2 merupakan contoh dari graf sederhana.

b. Graf tak sederhana

Graf yang mengandung gelang atau sisi ganda dinamakan graf tak sederhana. Pada Gambar 2, graf G_2 merupakan contoh graf yang mengandung sisi ganda (disebut graf ganda), sedangkan graf G_3 merupakan contoh graf yang mengandung gelang (disebut graf semu).



Gambar 3. (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Sumber: [1]

Berdasarkan ada tidaknya orientasi arah pada graf, graf dapat dikelompokkan menjadi 2, yaitu:

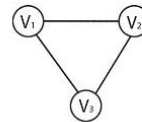
a. Graf tak berarah

Graf tak berarah merupakan graf yang tidak memiliki orientasi arah pada sisinya sehingga urutan penamaan sisi tidak diperhatikan. Sisi (u, v) memiliki arti yang sama dengan sisi (v, u) .

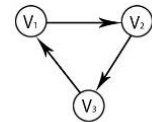
b. Graf berarah

Graf berarah merupakan graf yang memiliki orientasi arah pada setiap sisinya. Sisi berarah ini lebih sering disebut dengan busur. Pada graf berarah, sisi $(u, v) \neq (v, u)$. Sisi (u, v) menyatakan adanya sisi yang menghubungkan u dan v dengan u sebagai simpul asal dan v simpul terminal.

Undirected Graph



Directed Graph



Gambar 4. Contoh graf berarah dan tidak berarah
 Sumber: <https://www.e-education.psu.edu/>

3. Terminologi graf [1]

Di bawah ini didefinisikan beberapa terminologi (istilah) yang digunakan pada makalah ini.

3.1. Bertetangga (*Adjacent*)

Dua buah simpul saling bertetangga apabila keduanya terhubung pada sebuah sisi yang sama. Jika terdapat sebuah sisi (u, v) , simpul u dikatakan bertetangga dengan simpul v begitu juga sebaliknya.

3.2. Beririsan (*Incident*)

Sebuah sisi (u, v) dikatakan bersisian dengan simpul u dan simpul v .

3.3. Derajat (*Degree*)

Derajat suatu simpul menyatakan banyaknya sisi yang terhubung dengan simpul tersebut. Dengan kata lain, derajat juga menyatakan banyaknya sisi yang bersisian pada simpul tersebut. Derajat suatu simpul v dapat dinyatakan sebagai $d(v)$.

Berikut contoh besar derajat suatu simpul pada Gambar 2(c)

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

$$d(3) = 7 \text{ (sisi gelang dihitung 2)}$$

Pada graf berarah, nilai derajat $d(v)$ dinyatakan sebagai total dari jumlah busur yang masuk ke simpul v ($d_{in}(v)$) dan jumlah busur yang keluar dari simpul v ($d_{out}(v)$)

3.4. Lintasan (*Path*)

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

Jika graf yang ditinjau adalah graf sederhana, lintasan cukup dituliskan sebagai barisan simpul-simpul saja seperti $v_0, v_1, v_2, \dots, v_n$. Sebagai contoh pada Gambar 2(a), lintasan dari simpul 1 ke simpul 4 dapat ditulis sebagai 1, 2, 4 atau 1, 2, 3, 4 atau 1, 2, 4.

3.5. Terhubung (*Connected*)

Graf tak berarah G disebut graf terhubung jika untuk setiap pasangan simpul u dan v di dalam himpunan V terdapat lintasan dari u ke v (yang juga berarti ada lintasan dari u ke v).

3.6. Upagraf (*Subgraph*)

Misalkan terdapat graf $G = (V, E)$, graf $G_1 = (V_1, E_1)$ adalah upagraf dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$

3.7. Upagraf Merentang (*Spanning Subgraph*)

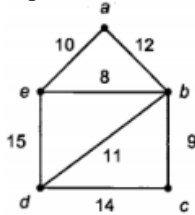
Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan upagraf merentang jika $V_1 = V$. Dengan kata lain, upagraf G_1 mengandung semua simpul dari G .

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

Siklus merupakan lintasan yang berawal dan juga berakhir pada simpul yang sama.

3.9. Graf Berbobot (Weighted Graph)

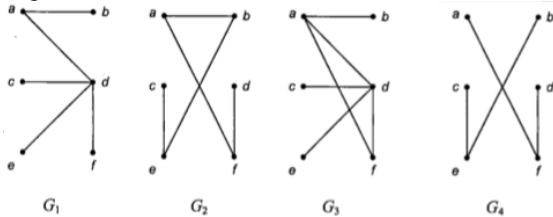
Graf berbobot adalah graf yang setiap sisinya memiliki nilai atau bobot tertentu. Bobot tersebut dapat merepresentasikan waktu, jarak, ongkos produksi, dan sebagainya tergantung persoalan yang ingin diimplementasikan dalam graf tersebut.



Gambar 5. Contoh graf berbobot
Sumber: [1]

B. Pohon

Pohon merupakan graf tak berarah terhubung yang tidak mengandung sirkuit [1]. Dari definisi tersebut, dapat diketahui bahwa ciri-ciri pohon ialah terhubung dan tidak mengandung sirkuit. Pada Gambar 6, diketahui bahwa G_1 dan G_2 termasuk pohon, sedangkan G_3 bukan pohon karena mengandung sirkuit dan G_4 juga bukan pohon karena ada simpul yang tidak saling berhubungan.



Gambar 6. Contoh pohon dan bukan pohon
Sumber: [1]

Misalkan $G = (V, E)$ adalah pohon dengan n simpul, maka dia akan memiliki beberapa sifat yaitu:

1. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
2. G terhubung dan memiliki $m = n - 1$ buah sisi.
3. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.

Dalam pokok bahasan pohon, terdapat pokok bahasan lain yang disebut sebagai pohon merentang. Misalkan $G = (V, E)$ adalah graf tak berarah terhubung yang bukan pohon (G memiliki sirkuit), G dapat diubah menjadi pohon T dengan cara memutuskan sirkuit-sirkuit yang ada. Pohon T tersebut disebut sebagai pohon merentang dari graf G .

C. Algoritma Prim

Algoritma Prim merupakan algoritma untuk membentuk pohon merentang minimum dari suatu graf. Misalkan T adalah pohon merentang yang sisi-sisinya diambil dari graf G . Berikut langkah dari algoritma Prim[1]:

1. Ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T .
2. Pilih sisi e yang mempunyai bobot minimum dan berisikan dengan simpul di T , tetapi e tidak membentuk sirkuit di T . Masukkan e ke dalam T .
3. Ulangi 2 sebanyak $n-2$ kali.

Dalam notasi *pseudo-code*, algoritma Prim dapat dituliskan sebagai berikut:

```

procedure Prim(input G: graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf
terhubung G.
Masukan: graf berbobot terhubung  $G = (V, E)$ ,  $|V| = n$ 
Keluaran: pohon merentang minimum  $T = (V, E)$ 
}

Deklarasi
e : sisi

Algoritma
T <- sisi e yang mempunyai bobot minimum di
dalam E
E <- E - {e}
for i <- 1 to n-2 do
    e <- sisi yang mempunyai bobot terkecil
    di dalam E dan bersisian dengan
    simpul di T.
    T <- T U {e} { masukkan e ke dalam T }
    E <- E - {e} { buang e dari E }
endfor
    
```

D. Daerah Wisata Kabupaten Karanganyar

Pada makalah ini, penulis akan lebih membahas daerah wisata alam yang berada di daerah kaki Gunung Lawu yang biasanya memiliki banyak pengunjung. Berikut daerah wisata yang akan dibahas [2]:

1. Candi Suku

Candi Suku merupakan candi yang terletak di Dukuh Berjo, Desa Suku, Kecamatan Ngargoyoso, Kabupaten Karanganyar. Candi ini memiliki berbagai hiasan dan relief yang unik. Selain itu, bentuk candi ini berupa trapesium yang strukturnya sangat mirip dengan bentuk piramida.

2. Candi Cetho

Candi Cetho merupakan candi yang terletak di Desa Gumeng, Kecamatan Jenawi, Kabupaten Karanganyar. Candi ini terletak ditepian hutan pinus pada ketinggian 1.470 mdpl. Arsitektur bangunannya mirip dengan bangunan pura yang ada di Bali.

3. Air Terjun Grojogan Sewu

Grojogan Sewu berarti air terjun seribu. Namun, bukan berarti air terjun tersebut berjumlah seribu. Kata sewu ini berasal dari seribu pecak atau satuan jarak yang merupakan tinggi air terjun tersebut. Satu pecak sama dengan satu telapak kaki orang dewasa.

4. Air Terjun Jumog

Air Terjun Jumog terletak di sebelah selatan Candi Suku. Air terjun ini dibuka untuk umum sejak tahun 2004. Air terjun ini tidak terlalu tinggi, tetapi memiliki keindahan yang sangat elok.

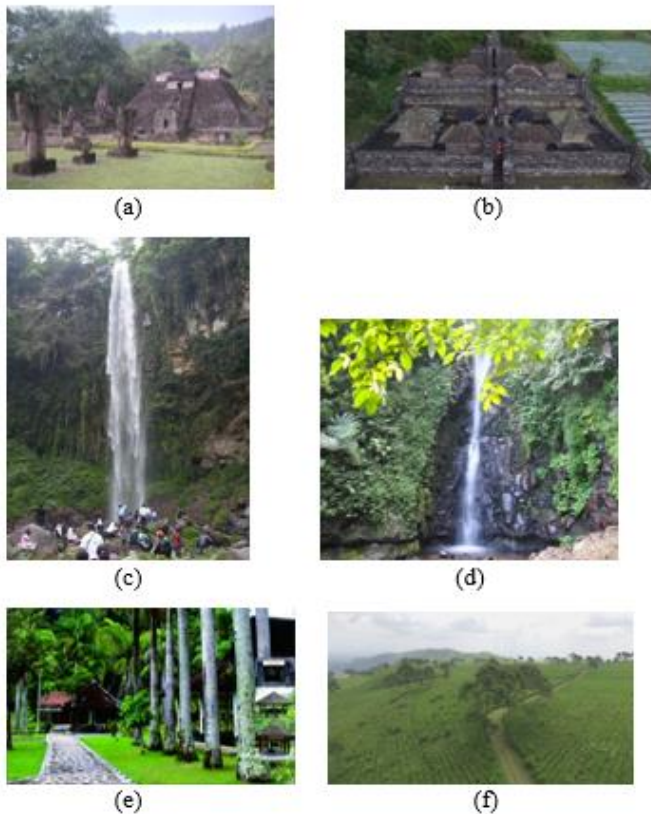
5. Sapta Tirta

Sapta tirta berarti tujuh mata air. Objek wisata ini memiliki tujuh mata air yang kandungan mineralnya satu sama lain berbeda. Objek wisata ini terletak di Desa Pablengan.

6. Kebun Teh Kemuning

Objek wisata Kebun Teh Kemuning ini salah satu objek wisata yang cocok untuk berwisata dengan keluarga. Di sini,

pengunjung dapat menikmati pemandangan yang indah dan hijau yang dipadukan dengan udara yang sejuk.



Gambar 7. (a) Candi Sukuh, (b) Candi Cetho, (c) Air Terjun Grojogan Sewu, (d) Air Terjun Jumog, (e) Sapta Tirta, (f) Kebun Teh Kemuning
Sumber: [2]

Selain itu, penulis akan menambahkan beberapa terminal bus karena sesuai dengan judul makalah ini, harapannya bus wisata juga akan mengunjungi beberapa terminal bus yang ada.

1. Terminal Bejen

Terminal Bejen merupakan terminal bus yang terletak di Kelurahan Tegalgede, Kabupaten Karanganyar. Terminal ini termasuk terminal yang menghubungkan antara daerah kota dengan daerah wisata.

2. Terminal Karangpandan

Terminal Karangpandan merupakan terminal bus yang terletak di Kecamatan Karangpandan, Kabupaten Karanganyar. Terminal ini merupakan terminal yang cukup strategis karena dapat menghubungkan berbagai daerah wisata.

III. PENERAPAN ALGORITMA PRIM UNTUK MERENCANAKAN RUTE BUS WISATA

A. Menentukan jarak simpul

Pertama, akan ditentukan terlebih dahulu jarak simpul (bobot sisi) antara simpul yang bersisian. Pada Tabel 1 berikut, baris menyatakan simpul asal, kolom menyatakan simpul tujuan. Isi dari suatu sel (i, j) menyatakan jarak dari simpul i ke simpul j dalam kilometer (km). Apabila suatu sel berisi '-', hal itu menyatakan bahwa simpul i dan simpul j tidak saling bersisian. Karena sisi yang dinyatakan tidak berarah, penulis hanya akan

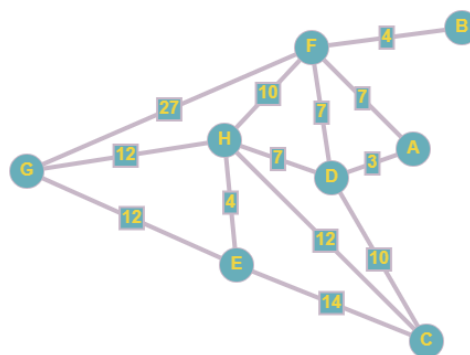
menggunakan sel segitiga bawah saja karena sisi $(i, j) = (j, i)$

Tabel 1. Jarak antar simpul

	A	B	C	D	E	F	G	H
A								
B	-							
C	-	-						
D	3	-	10					
E	-	-	14	-				
F	7	4	-	7	-			
G	-	-	-	-	12	27		
H	-	-	12	7	4	10	12	

Keterangan:

- A : Candi Sukuh
- B : Candi Cetho
- C : Air Terjun Grojogan Sewu
- D : Air Terjun Jumog
- E : Sapta Tirta
- F : Kebun Teh Kemuning
- G : Terminal Bejen
- H : Terminal Karangpandan



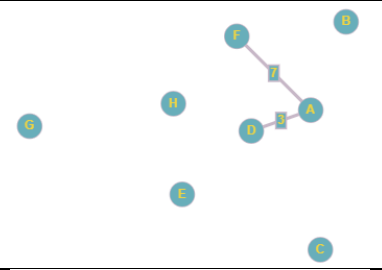
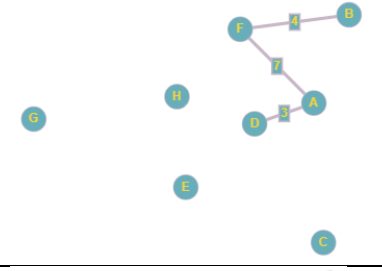
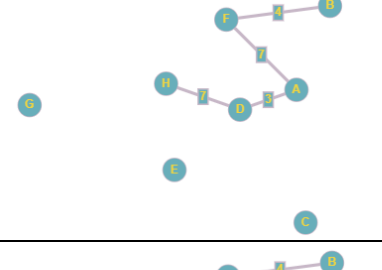
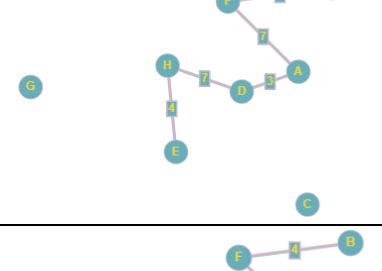
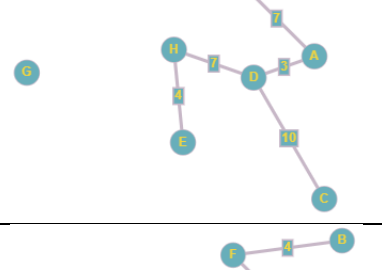
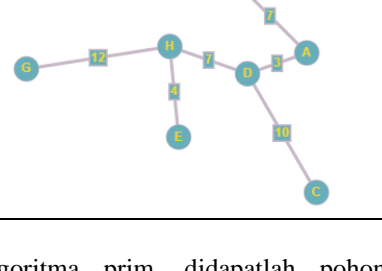
Gambar 8. Hasil visualisasi dalam graf

B. Algoritma Prim untuk mendapatkan pohon merentang minimum

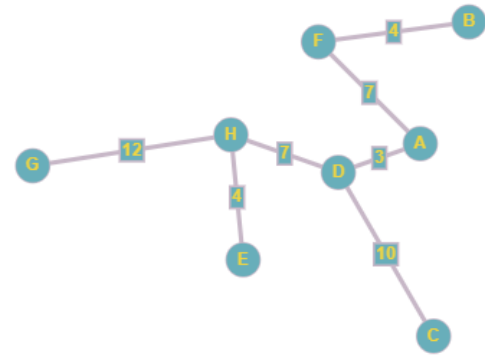
Setelah mendapatkan grafnya, lakukan algoritma Prim untuk mendapatkan pohon merentang minimum. Pertama adalah menentukan sisi dengan bobot terkecil. Dari Gambar 7, diketahui bahwa sisi yang memiliki bobot terkecil adalah sisi (A, D) dengan bobot 3 km. Berikut langkah-langkah dalam pembuatan pohon merentang minimum.

Tabel 2. Proses pembentukan pohon merentang minimum

Langkah	Sisi	Bobot	Graf
1	(A, D)	3	

2	(A, F)	7	
3	(F, B)	4	
4	(D, H)	7	
5	(H, E)	4	
6	(D, C)	10	
7	(G, H)	12	

Setelah melakukan algoritma prim, didapatkan pohon merentang minimum dari graf objek-objek wisata tersebut. Hasil pohon merentang tersebut adalah sebagai berikut.



Gambar 9. Hasil pohon merentang minimum setelah dilakukan algoritma Prim

Dari Gambar 9, diketahui bahwa total jarak tempuh untuk setiap simpul adalah $12 + 4 + 7 + 10 + 3 + 7 + 4 = 47$ km. Gambar tersebut juga merupakan representasi rute yang harus dibuat untuk rute bus wisata yang ada di Kabupaten Karanganyar.

Pembentukan rute tersebut juga bukan rute yang unik karena ada beberapa sisi yang memiliki bobot yang sama. Ketika dilakukan algoritma Prim, pada langkah 2, sebenarnya dapat diambil sisi selain (A, F) yaitu (D, F) karena kedua sisi tersebut memiliki bobot yang sama yaitu 7. Meskipun pengambilan sisi yang dilakukan berbeda, hasil jumlah total bobot pohon akan memiliki hasil yang sama, tetapi dengan struktur pohon yang berbeda.

IV. SIMPULAN

Algoritma Prim dapat digunakan untuk membentuk suatu pohon merentang minimum dari suatu graf berbobot. Pada makalah ini, algoritma Prim digunakan untuk membentuk rute bus wisata pada Kabupaten Karanganyar dengan total rute terpendek antara tiap simpul yang ada. Total rute yang dihasilkan adalah 47 km.

Dalam pembentukan pohon merentang minimum dengan algoritma Prim, tidak pasti dihasilkan satu jenis pohon yang unik apabila ada lebih dari satu sisi yang memiliki bobot yang sama. Namun, total bobot pada setiap pohonnya pasti akan sama meskipun bentuk pohon yang didapatkan berbeda.

V. UCAPAN TERIMA KASIH

Puji syukur saya ucapkan kepada Tuhan yang Maha Esa atas selesainya makalah matematika diskrit berikut. Saya juga mengucapkan terima kasih kepada para dosen matematika diskrit terutama kepada Pak Dr. Ir. Rinaldi Munir, MT. Selaku dosen matematika diskrit kelas 01 yang telah mengajarkan berbagai ilmunya agar para mahasiswanya dapat menyelesaikan berbagai permasalahan yang berhubungan dengan matematika diskrit. Selain itu, saya juga berterima kasih kepada semua orang yang telah membantu saya baik secara langsung maupun tidak langsung hingga sekarang.

REFERENSI

- [1] Munir, Rinaldi. 2016. Matematika Diskrit Edisi Revisi keenam. Bandung: Informatika Bandung.
- [2] <http://disparpora.karanganyarkab.go.id/>, diakses 5 Desember 2019

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 6 Desember 2019



Okugata Fahmi N. Y. F.
13518031