

Penerapan Algoritma Prim dalam Perencanaan Rute Perjalanan Wisata di Kabupaten Magelang

Jundan Haris - 13520155¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13520155@std.itb.ac.id

Abstract—Kabupaten Magelang memiliki banyak destinasi wisata, baik itu wisata alam maupun wisata peninggalan sejarah. Dengan banyaknya destinasi wisata tersebut, dibutuhkan rute tercepat untuk mengunjungi tempat-tempat yang ada. Pada makalah ini, masalah tersebut dimodelkan dengan teori graf dan pohon. Untuk mencari rute tercepat, digunakan algoritma prim dalam prosesnya. Algoritma prim sendiri yaitu algoritma untuk mencari *minimum spanning tree* pada sebuah graf. Dengan penggunaan algoritma tersebut, didapatkan sebuah pohon merentang minimum yang dapat digunakan untuk penentuan rute.

Keywords—Algoritma Prim, graf, rute, Kabupaten Magelang.

I. PENDAHULUAN

Indonesia adalah negara yang kaya akan keunikan alamnya. Dari ujung timur hingga ujung barat, ujung utara hingga ujung selatan, hampir semua daerah memiliki keunikan alamnya masing-masing. Tidak pernah habis tempat di Indonesia untuk dijelajahi. Dengan wilayah Indonesia yang luas dan beraneka ragam, tentu akan memunculkan banyak tempat wisata di berbagai daerah, bahkan bisa dibilang sudah terkenal hingga mancanegara, contohnya adalah Wakatobi di Sulawesi, Raja Ampat di Papua, Danau Toba di Sumatra, Bali dengan pantainya, dan masih banyak lagi.

Begitu juga halnya dengan Provinsi Jawa Tengah, lebih tepatnya di Kabupaten Magelang juga memiliki beberapa destinasi wisata. Tentu saja mungkin beberapa tidak seterkenal wisata-wisata dari daerah di Indonesia lain yang sudah mendunia. Namun, dengan meramaikan tempat-tempat wisata di daerah, secara tidak langsung sudah membantu jalannya perekonomian di daerah tersebut. Beberapa tempat wisata yang berada di Kabupaten Magelang adalah Candi Borobudur, Candi Mendut, Punthuk Setumbu, dan masih banyak lagi.

Untuk menjelajahi semua tempat wisata secara efektif, maka penulis membuat makalah ini, yaitu penggunaan algoritma prim dalam menentukan sebuah rute perjalanan wisata. Penulis akan menggunakan teori graf pada pengerjaannya. Setiap tempat wisata akan dianggap simpul, dan jalan yang ditempuh untuk menghubungkan dua tempat dianggap sisi. Sedangkan algoritma prim sendiri adalah algoritma untuk mencari hasil minimum spanning tree pada graf berbobot.

Dengan penulisan makalah ini, penulis berharap makalah yang ditulis dapat bermanfaat bagi pembaca sekaligus menambah rasa ketertarikan pembaca terhadap Kabupaten Magelang, sehingga berniat berkunjung ke daerah ini dan

menikmati tempat-tempat wisata di suatu hari nanti.

II. DASAR TEORI

Graf adalah gabungan dari titik-titik yang disebut *vertex*(simpul) yang dihubungkan oleh beberapa garis yang disebut *edge*(sisi). Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Secara matematis, graf ditulis sebagai :

$$G = (V, E)$$

Dengan

G = Graf

V = himpunan tak kosong dari *vertices*(simpul-simpul)

E = himpunan tak kosong dari *edges*(sisi-sisi)

Berdasarkan ada tidaknya gelang atau sisi ganda, graf digolongkan menjadi dua jenis, yaitu :

1. *Simple Graph* (Graf Sederhana)

Graf yang tidak mengandung gelang maupun sisi ganda.

2. *Un-simple Graph* (Graf Tak-sederhana)

Graf yang mengandung sisi ganda atau gelang. Graf tak-sederhana sendiri dibedakan lagi menjadi :

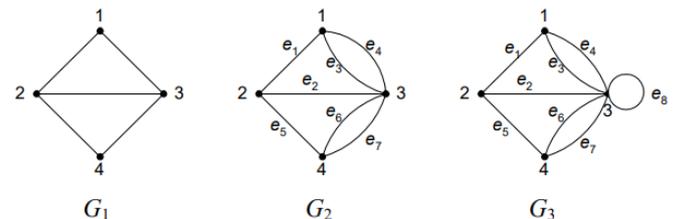
a) *Multi-graph* (Graf Ganda)

Graf yang mengandung sisi ganda

b) *Pseudo-graph* (Graf Semu)

Graf yang mengandung sisi gelang

Berikut adalah contoh graf berdasarkan ada tidaknya gelang dan sisi ganda.



Gambar 1. Contoh Graf Berdasarkan Ada Tidaknya Gelang dan Sisi Ganda. G_1 = *simple graph*, G_2 = *multi-graph*, G_3 = *pseudo-graph*

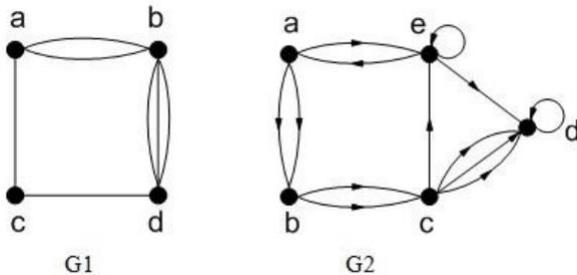
(sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>)

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

1. *Undirected Graph* (Graf Tak-berarah)
Graf yang sisinya tidak mempunyai orientasi arah.
2. *Directed Graph* (graf Berarah)
Graf yang setiap sisinya diberikan orientasi arah.

Berikut adalah contoh graf berdasarkan orientasi pada sisi graf.



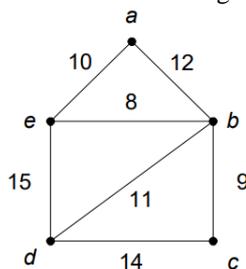
Gambar 2. Contoh Graf Berdasarkan Orientasi pada Sisi. G1 = *undirected graph*, G2 = *directed graph*
(sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>)

Berdasarkan nilai sisinya, graf dibedakan menjadi dua yaitu :

1. Graf Tidak Berbobot
Graf yang sisinya tidak mengandung nilai.
2. Graf Berbobot
Graf yang sisinya mengandung nilai/bobot. Setiap bobotnya merepresentasikan hubungan antarsimpul pada graf. Pada makalah ini, graf yang akan digunakan adalah graf berbobot ini.

Contoh dari graf berbobot adalah sebagai berikut.



Gambar 3. Contoh Graf Berbobot
(sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>)

Terdapat banyak sekali penerapan dari teori graf, diantaranya adalah :

1. Pembuatan rangkaian listrik
Komponen-komponen listrik dihubungkan satu sama lain dengan sebuah garis berdasarkan teori graf.
2. Isomer senyawa kimia
Atom-atom pada senyawa kimia dihubungkan dengan ikatan kimia. Atom direpresentasikan sebagai simpul dan ikatan kimia sebagai sisinya.

3. Jejaring makanan pada biologi
Hewan-hewan direpresentasikan sebagai simpul, sedangkan relasi yang menggambarkan hubungan makanan-yang dimakan direpresentasikan sebagai sisinya.
4. Pengujian program
Bagian-bagian dari program yang diuji merupakan simpul graf, sedangkan penghubung antara bagian-bagian program tersebut merupakan sisi graf.
5. *Vending machine*
Vending machine atau mesin penjual otomatis adalah mesin yang dapat mengeluarkan makanan, minuman, barang apapun yang dijual kepada pembeli tanpa adanya penjual. Pembeli hanya perlu memasukkan uang yang sesuai, lalu barang yang diinginkan akan didapatkan. Pada prosesnya, vending machine ini dapat dimodelkan dengan *state-state* tertentu. Pada penggambaran *state* tersebut diterapkan teori graf. Setiap *state* adalah simpul, dan input adalah sisinya.

Terdapat terminologi atau istilah-istilah dalam teori graf, beberapa diantaranya yaitu :

1. *Adjacency* (Ketetanggaan)
Dua buah simpul dikatakan bertetangga jika keduanya terhubung langsung oleh sebuah sisi.
2. *Incidency* (Bersisian)
Sebuah simpul A dan sisi E disebut bersisian jika salah satu ujung E adalah A.
3. *Isolated Vertex* (Simpul Terpencil)
Simpul A disebut terpencil jika tidak ada satu sisi pun yang bersisian dengannya.
4. *Null Graph* (Graf Kosong)
Graf yang hanya berisi kumpulan simpul, tanpa ada satupun sisi.
5. *Degree* (Derajat)
Derajat dari sebuah simpul adalah banyaknya sisi yang bersisian dengan simpul tersebut.
6. *Path* (Lintasan)
Adalah jalur dari sebuah simpul awal ke sebuah simpul tujuan.
7. *Cycle or Circuit* (Siklus atau Sirkuit)
Adalah jalur dari sebuah simpul awal ke sebuah simpul tujuan.
8. *Connected* (Keterhubungan)
Simpul A dan B dikatakan terhubung jika ada lintasan dari A ke B.
9. *Subgraph* (Upagraf)
A disebut upagraf dari graf G jika setiap vertex di A juga ada di G, dan semua edge di A juga ada di G
10. *Complement* (Komplemen Upagraf)
Jika A adalah upagraf dari G, maka komplemen dari A adalah graf yang mengandung semua simpul yang ada di G dan tidak ada di A, juga semua sisi yang tidak terhubung dengan satupun simpul di A.
11. *Spanning Subgraph* (Upagraf Merentang)
Graf A disebut upagraf merentang dari graf G, jika semua simpul di G juga di A, yang artinya simpul A dan G sama.
12. *Cut-set*

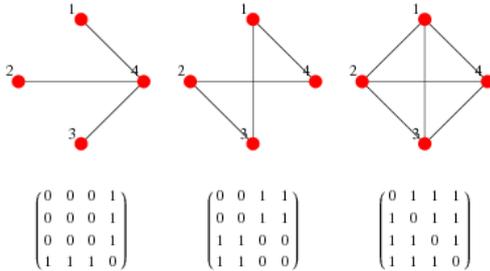
Graf A disebut upagraf merentang dari graf G, jika semua simpul di G juga di A, yang artinya simpul A dan G sama.

[021-2022/matdis21-22.htm](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm)

Graf tidak hanya berupa gambar. Graf juga bisa direpresentasikan dengan berbagai cara, di antaranya yaitu :

1. *Adjacency Matrix* (Matriks Ketetanggaan)

Adalah matriks yang merepresentasikan nilai keterhubungan antarsimpul pada sebuah graf. Untuk graf tidak berbobot, akan bernilai 1 ketika dua simpul terhubung dan 0 ketika tidak terhubung. Sedangkan pada graf berbobot, nilai yang ditunjukkan adalah nilai/bobot dari sebuah sisi yang menghubungkan dua simpul.

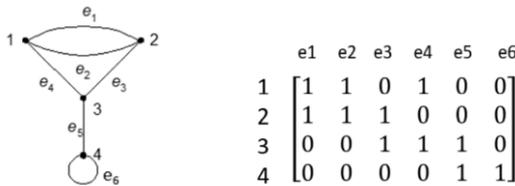


Gambar 4. Matriks Ketetanggaan dan Grafnya (sumber :

<http://dyahslib.blogspot.com/2013/11/adjacency-matrix.html>)

2. *Incidency Matrix* (Matriks Bersisian)

Adalah matriks yang merepresentasikan hubungan antara simpul dan sisi. Baris sebagai simpul dan kolom sebagai sisi. Jika simpul A dan sisi E bersisian, maka akan ditandai dengan 1, sedangkan jika tidak, akan ditandai dengan 0.

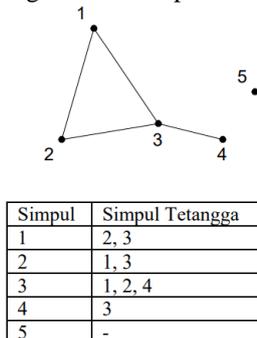


Gambar 5. Matriks Bersisian dan Grafnya (sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>)

3. *Adjacency List* (Senarai Ketetanggaan)

Adalah senarai yang mencatat setiap semua simpul yang bertetanggaan dengan suatu simpul.



Gambar 6. Senarai Ketetanggaan dan Grafnya (sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>)

III. ALGORITMA PRIM

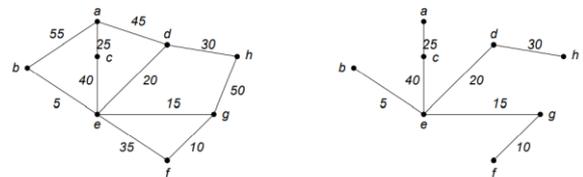
Sebelum membahas mengenai algoritma prim, terlebih dahulu harus mengetahui bentuk lain dari graf yang disebut pohon. Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Definisi lain dari pohon yaitu misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n . Maka :

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

Dapat dikatakan, setiap pohon adalah graf, namun tidak semua graf adalah pohon.

Pohon merentang atau *spanning tree* dari sebuah graf terhubung adalah sebuah upagraf merentang yang memenuhi definisi dari pohon, yaitu menghilangkan sirkuit dari upagraf merentang tersebut. Pohon merentang didapat dengan memutus sirkuit di dalam graf.

Graf terhubung-berbobot mungkin mempunyai lebih dari 1 pohon merentang. Pohon merentang yang berbobot minimum dinamakan pohon merentang minimum (*minimum spanning tree*).



Gambar 7. Contoh Pohon Merentang Minimum (sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>)

Algoritma Prim adalah salah satu algoritma yang bisa digunakan untuk mencari sebuah *minimum spanning tree* dari sebuah graf berbobot. Pada makalah ini, algoritma inilah yang akan digunakan dalam penentuan rute perjalanan wisata. Algoritmanya adalah sebagai berikut.

1. ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T.
2. pilih sisi (u, v) yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi (u, v) tidak membentuk sirkuit di T. Masukkan (u, v) ke dalam T.
3. ulangi langkah 2 sebanyak $n - 2$ kali.

Atau jika dituliskan dalam sebuah *pseudocode* adalah sebagai berikut.

Kabupaten Magelang yang paling terkenal, bahkan hingga mancanegara. Candi Borobudur ini merupakan candi Buddha terbesar yang dibangun pada zaman Mataram Kuno. Candi Borobudur bahkan dinobatkan sebagai salah satu tujuh keajaiban dunia, akan sangat disayangkan apabila sudah berkunjung ke Magelang tapi tidak ke Candi Borobudur. Kemegahan candi dengan stupa-stupa yang berjejer dihiasi dengan pemandangan yang indah, tentu sangat menarik.

4. Candi Mendut

Candi Mendut berjarak sekitar 3 km dari Candi Borobudur. Terletak di jln. Mayor Kusen Mendut, Kecamatan Mungkid, Kabupaten Magelang. Candi ini merupakan candi peninggalan agama Buddha. Candi ini penuh dengan nilai sejarah dan makna kehidupan tertuang pada relief-relief candi. Candi Mendut bisa menjadi tujuan wisata sejarah, budaya, serta religi.

5. Taman Nasional Gunung Merapi Jurang Jero

Di kawasan ini, wisatawan bisa menikmati keindahan alam berupa pohon pinus dan lainnya. Pengunjung juga bisa keliling di kawasan ini dengan naik jeep. Ada juga jalur untuk sepeda downhill. Tempat ini terletak di Ngargosoko, Srumbung, Kabupaten Magelang, Jawa Tengah. Tempat ini berjarak sekitar 8 hingga 9 km dari puncak Gunung Merapi.

6. Ketep Pass

Ketep Pass merupakan gardu pandang yang berada pada ketinggian 1.200 mdpl. Lokasinya ada di Jalan Blabak Boyolali kilometer 16, Ketep, Kecamatan Sawangan, Kabupaten Magelang. Di tempat ini pengunjung dapat menikmati pemandangan Gunung yang mengelilingi Magelang seperti Merapi, Merbabu, Sumbing, dan Sindoro. Di Ketep Pass juga terdapat Museum Volcanologi bernama Ketep Vulcano Centre yang memberikan informasi terkait Gunung Merapi.

7. Air Terjun Grenjengan Kembar

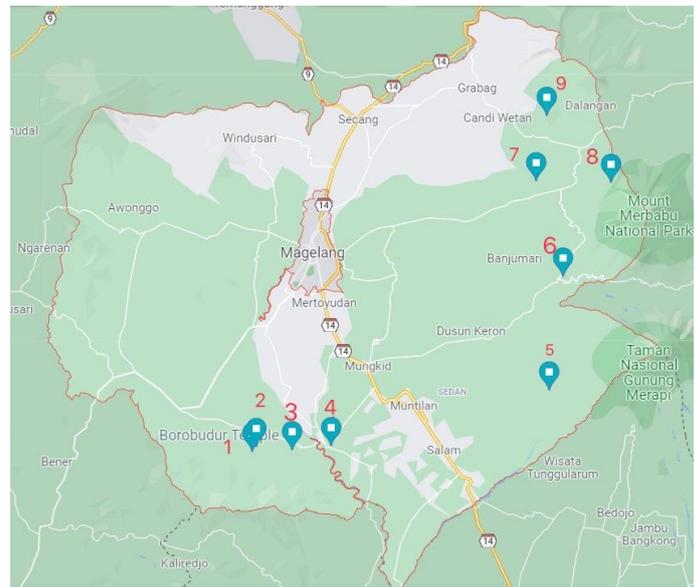
Air Terjun Grenjengan Kembar memiliki dua aliran yang bersebelahan. Lokasinya ada di Dusun Citran, Desa Muneng Warangan, Kecamatan Pakis, Kabupaten Magelang. Berada di kawasan lereng Gunung Merbabu, Air Terjun ini mengalirkan air yang bersumber dari mata air Gunung Merbabu.

8. Pendakian Gunung Merbabu

Gunung Merbabu merupakan salah satu gunung yang terletak di beberapa kabupaten, salah satunya Kabupaten Magelang. Gunung ini menjadi salah satu gunung tujuan bagi para pendaki.

9. Puncak Gunung Andong

Untuk yang belum terbiasa naik gunung, tempat ini adalah tempat yang cocok untuk mencobanya. Palsanya, dari awal pendakian hingga puncak hanya membutuhkan waktu satu hingga dua jam. Puncak gunung bisa digunakan sebagai lokasi *camping*. Pemandangan dari puncak menyajikan panorama alam yang indah. Gunung Andong terletak di Dusun Sawit, Desa Girirejo, Kecamatan Ngablak, Kabupaten Magelang.



Gambar 11. Peta Tempat Wisata Kabupaten Magelang (sumber : dokumen penulis, dengan peta dari google maps)

V. PEMBAHASAN

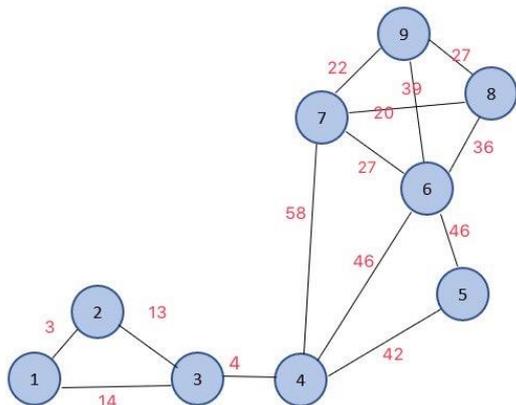
Dengan banyak tempat wisata, terkadang pengunjung bingung harus menentukan mana tempat yang harus dikunjungi terlebih dahulu. Oleh karena itu, permasalahan dimodelkan secara matematis dengan menggunakan teori graf. Untuk menentukan rute perjalanannya, digunakan pendekatan menggunakan algoritma prim. Algoritma prim mencari *minimum spanning tree* dari sebuah graf yang merepresentasikan jalan-jalan yang menghubungkan setiap tempat wisata dengan total jarak minimum.

Untuk memodelkan rute perjalanan menjadi graf berbobot, maka perlu ditentukan bobot atau nilai masing-masing sisi atau jalan yang dilalui. Pada permodelan ini, akan digunakan waktu tempuh dari masing-masing tempat ke tempat yang lain menggunakan mobil. Data diambil dari estimasi waktu (dalam menit) perjalanan menggunakan aplikasi *google maps*. Kemudian estimasi waktu tersebut akan dimasukkan dalam tabel matriks ketetanggaan (*adjacency matrix*). Angka satu sampai sembilan menjadi simpul (sesuai dengan gambar 11) dan nilai atau bobotnya yang menjadi sisi, yaitu waktu tempuh.

	1	2	3	4	5	6	7	8	9
1	∞	3	14	∞	∞	∞	∞	∞	∞
2	3	∞	13	∞	∞	∞	∞	∞	∞
3	14	13	∞	4	∞	∞	∞	∞	∞
4	∞	∞	4	∞	42	46	58	∞	∞
5	∞	∞	∞	42	∞	46	∞	∞	∞
6	∞	∞	∞	46	58	∞	27	36	39
7	∞	∞	∞	58	∞	27	∞	20	22
8	∞	∞	∞	∞	∞	36	20	∞	27
9	∞	∞	∞	∞	∞	39	22	27	∞

Tabel 1. *Matrix Adjacency* dari Peta Tempat Wisata Kabupaten Magelang (sumber : dokumen penulis)

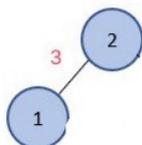
Dari *Matrix Adjacency* tersebut, dapat dibuat sebuah graf yang menggambarkan peta tempat wisata Kabupaten Magelang. Graf ini terlihat lebih sederhana dan mudah dilihat, sehingga memudahkan penulis serta pembaca dalam membayangkan kondisinya. Graf yang sudah dibuat dapat dilihat pada (gambar 12).



Gambar 12. Graf dari Peta Tempat Wisata Kabupaten Magelang (sumber : dokumen penulis)

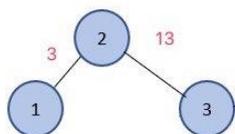
Setelah dibuat graf, saatnya menggunakan algoritma prim untuk menentukan rutennya.

- Langkah pertama. ambil sisi dari graf yang berbobot minimum dan masukkan simpul ke T. T adalah himpunan yang menyimpan simpul yang sudah dilewati. Pada kasus ini, graf dengan sisi minimum yaitu antara simpul 1 dengan simpul 2.



Gambar 13. Langkah Pertama Algoritma Prim (sumber : dokumen penulis)

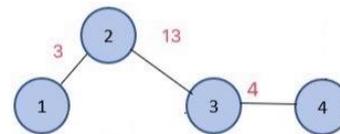
- Langkah kedua. Tentukan sisi yang bersisian dengan simpul di T dan memiliki bobot minimum. Sisi tidak boleh menghasilkan sirkuit. Maka didapat sisi antara simpul 2 dengan simpul 3, yaitu sisi dengan bobot 13. Sisi lain yang memungkinkan berbobot 14, jadi dipilih sisi dengan bobot 13. Masukkan simpul 3 ke T.



Gambar 14. Langkah Kedua Algoritma Prim (sumber : dokumen penulis)

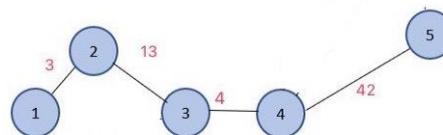
- Langkah ketiga. Masih sama dengan langkah sebelumnya, yaitu mencari sisi yang bersisian dengan simpul di T dengan bobot minimum dan

tidak boleh menghasilkan sirkuit. Karena hanya terdapat satu sisi yang memungkinkan, maka tidak perlu membandingkan dengan sisi-sisi lain. Didapatkan sisi antara simpul 3 dengan simpul 4. Masukkan simpul 4 ke T.



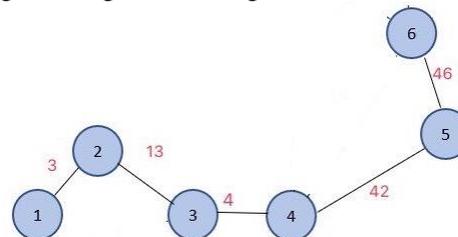
Gambar 15. Langkah Ketiga Algoritma Prim (sumber : dokumen penulis)

- Langkah keempat. Ulangi langkah seperti sebelumnya. Didapatkan sisi antara simpul 4 dengan simpul 5 yang memiliki bobot 42. Sisi yang lain memiliki bobot 46 dan 58. Oleh karena itu, dipilih sisi dengan bobot 42. Masukkan simpul 5 ke T.



Gambar 16. Langkah Keempat Algoritma Prim (sumber : dokumen penulis)

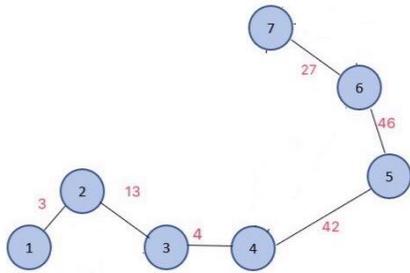
- Langkah kelima. Masih sama dengan langkah sebelumnya, didapatkan sisi di antara simpul 5 dengan simpul 6 yang memiliki bobot 46. Masukkan simpul 6 ke T. Maka bentuk *minimum spanning tree* yang sekarang adalah sebagai berikut.



Gambar 17. Langkah Kelima Algoritma Prim (sumber : dokumen penulis)

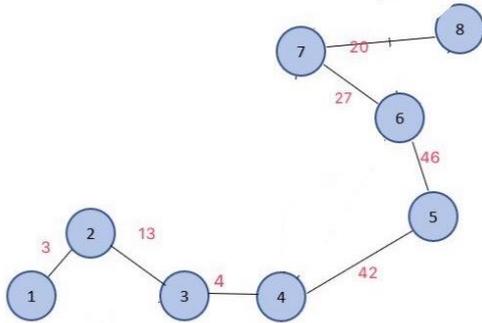
Sebenarnya bisa dipilih juga sisi antara simpul 4 dengan simpul 6 karena memiliki bobot yang sama dengan sisi antara simpul 5 dengan simpul 6. Sehingga bisa dipilih salah satu saja. Penulis memilih sisi di antara simpul 5 dan simpul 6.

- Langkah keenam. Sisi selanjutnya yang didapatkan adalah sisi di antara simpul 6 dengan simpul 7 yang memiliki bobot 27. Masukkan simpul 7 ke dalam T. bentuk pohon yang sekarang menjadi



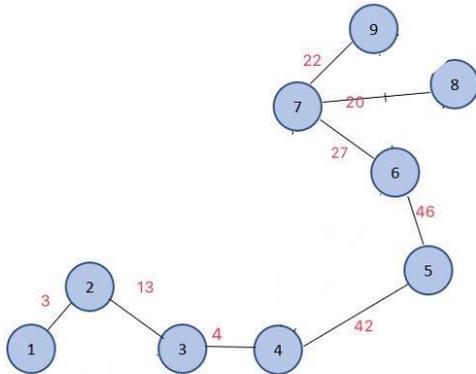
Gambar 18. Langkah Keenam Algoritma Prim (sumber : dokumen penulis)

- Langkah ketujuh. Didapatkan sisi di antara simpul 7 dan simpul 8. Masukkan simpul 8 ke dalam T. bentuk pohon sementara menjadi



Gambar 19. Langkah Ketujuh Algoritma Prim (sumber : dokumen penulis)

- Langkah kedelapan atau langkah terakhir. Dengan proses yang sama seperti sebelumnya, didapatkan sisi di antara simpul 7 dengan simpul 9. Masukkan simpul 9 ke dalam T. Karena semua simpul sudah masuk kee T, berarti ini menjadi langkah terakhir. Hasil yang didapatkan merupakan *minimum spanning tree* dari graf pada gambar 12.



Gambar 20. *Minimum Spanning Tree* dari Graf pada Gambar 12. (sumber : dokumen penulis)

Dengan begitu, didapatkan sebuah pohon merentang minimum (*minimum spanning tree*) dari graf pada gambar 12. Pohon ini dapat digunakan sebagai acuan rute perjalanan ketika mengunjungi tempat-tempat wisata yang berada di Kabupaten Magelang. Pohon ini bukanlah satu-satunya pohon merentang

minimum yang ada. Terdapat pohon merentang minimum lain yang mungkin dihasilkan dari graf yang sama. Hal ini disebabkan karena beberapa sisi memiliki bobot yang sama.

VI. KESIMPULAN

Teori matematika diskrit yang banyak dan mungkin rumit serta tidak terbayang ternyata bisa diterapkan pada permasalahan kehidupan sehari-hari. Salah satunya yaitu yang penulis bahas pada makalah ini, graf dan pohon, khususnya algoritma prim yang digunakan untuk menentukan rute paling pendek. Dengan adanya makalah ini diharapkan ketika ada pembaca yang ingin mengunjungi tempat-tempat wisata di Kabupaten Magelang dapat terbantu. Dalam pengerjaan makalah ini, tentunya masih ada faktor-faktor lain yang mungkin belum penulis perhitungkan. Jadi, makalah ini masih jauh dari kata sempurna.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kehadiran Allah SWT karena rahmat serta hidayah-Nya makalah ini bisa selesai. Selanjutnya penulis ucapkan terima kasih kepada kedua orang tua yang selalu mendoakan penulis, Ibu Ulfa selaku dosen mata kuliah Matematika Diskrit K03 yang telah banyak memberikan ilmu sehingga penulis memahami materi dan dapat menyelesaikan makalah ini, teman-teman, serta segala pihak yang tidak dapat penulis sebutkan satu persatu.

REFERENSI

- <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm> (diakses pada tanggal 14 Desember 2021)
- <http://dyahslib.blogspot.com/2013/11/adjacency-matrix.html> (diakses pada tanggal 14 desember 2021).
- <http://magelangkab.go.id/home/detail/gambaran-umum/1808> (diakses pada tanggal 14 Desember2021).
- <https://docplayer.info/49265022-Bab-ii-gambar-2-1-peta-kabupaten-magelang.html> (diakses pada tanggal 14 Desember 2021).
- <https://www.nativeindonesia.com/tempat-wisata-di-magelang/> (diakses tanggal 13 Desember 2021).
- <https://travel.kompas.com/read/2020/09/16/070700027/25-tempat-wisata-keren-di-magelang-pas-untuk-liburan?page=all> (diakses pada tanggal 13 Desember 2021).
- <https://www.google.com/maps/place/Magelang+Regency,+Central+Java/@-7.5135438,110.1046825,11z/data=!3m1!4b1!4m5!3m4!1s0x2e7a88d7b46fe04d:0x3027a76e352bac0!8m2!3d-7.4305237!4d110.2832217> (diakses pada tanggal 13 Desember 2021)

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, 14 Desember 2021

Jundan Haris - 13520155