

# Pembuatan Rute Jalan Tol Terpendek di Kalimantan Tengah dengan Menggunakan Algoritma Prim dan Algoritma Kruskal

Fikri Khoiron Fadhila - 13520056<sup>1</sup>  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
<sup>1</sup>13520056@std.stei.itb.ac.id

**Abstract**—Pemerintah sedang mempersiapkan pembangunan sejumlah jalan tol di Provinsi Kalimantan Tengah untuk periode tahun 2020-2024. Pembangunan jalan tol ini bertujuan untuk memudahkan mobilitas antar kabupaten atau antar kota di Kalimantan Tengah. Pemilihan rute terpendek untuk jalan tol yang hendak di bangun dapat di aplikasikan menggunakan pohon merentang minimum dengan menggunakan algoritma prim dan algoritma kruskal. Rute jalan tol yang terpendek ini diharapkan dapat memberikan efisiensi, baik untuk biaya pembangunan itu sendiri maupun efisiensi dalam bentuk jalur tercepat bagi pengguna jalan.

**Keywords**—Jalan Tol, Kruskal, Pohon Merentang Minimum, Prim.

## I. PENDAHULUAN

Beberapa tahun terakhir, pemerintah berfokus pada pembangunan infrastruktur khususnya di pulau luar jawa. Infrastruktur yang pembangunannya lebih di fokuskan oleh pemerintah adalah jalan tol. Pemerintah beranggapan apabila sudah tersedia akses mobilitas antara 1 daerah ke daerah lain maka persebaran sumber daya alam maupun sumber daya manusia akan lebih mudah. Tidak semua daerah di Indonesia memiliki jalan tol, seperti Kalimantan Tengah, Kalimantan Tengah merupakan provinsi dengan luas wilayah terbesar di pulau kalimantan yang memiliki luas sebesar 153.563 kilometer persegi. Akan tetapi, belum semua kabupaten/kota di Provinsi Kalimantan Tengah terhubung dengan akses yang memadai.

Pembangunan jalan tol di Kalimantan Tengah ini diharapkan dapat menghubungkan semua kabupaten dan kota di Kalimantan Tengah yang terdiri dari 13 Kabupaten dan 1 Kota. Dengan mengaplikasikan graf dan pohon, kita dapat membuat sebuah rute jalan tol yang efisien atau memiliki rute terpendek, yang dapat menghubungkan semua kabupaten dan kota di Provinsi Kalimantan Tengah. Penyelesaian dilakukan dengan mengaplikasikan graf berbobot pada koordinat setiap kabupaten dan kota di Provinsi Kalimantan Tengah dan menghubungkan dengan sebuah sisi yang memiliki bobot yang merupakan representasi dari jalur berjarak yang memiliki satuan kilometer. Kemudian akan dicari sebuah pohon merentang minimum dari

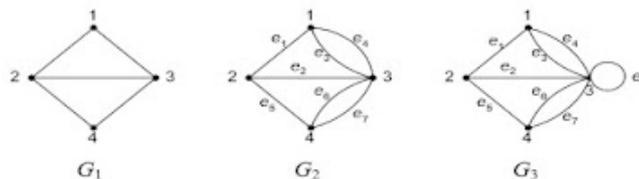
graf tersebut dengan menggunakan algoritma prim dan algoritma kruskal.

## II. LANDASAN TEORI

### A. Graf

Graf adalah representasi visual dari objek-objek diskrit dan hubungan antar objek – objeknya. Dalam bentuk matematis, graf dituliskan dengan  $G = (V, E)$  dengan  $V$  merupakan himpunan simpul dari graf dan  $E$  merupakan himpunan sisi yang menghubungkan sepasang simpul dari graf.

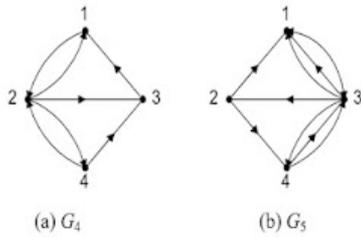
Berdasarkan keberadaan dari sisi ganda dan gelang pada graf, graf digolongkan menjadi 2 jenis yaitu graf tidak sederhana dan graf sederhana. Graf tidak sederhana yaitu graf yang mengandung sisi ganda atau gelang. Graf tidak sederhana juga dapat digolongkan menjadi 2 yaitu graf semu dan graf ganda. Graf semu merupakan graf yang mengandung gelang, dan graf ganda merupakan graf yang memiliki sisi ganda. Sedangkan graf sederhana merupakan graf yang tidak memiliki sisi ganda maupun sisi gelang.



Gambar 1. (G1) graf sederhana, (G2) graf ganda, dan (G3) graf semu

Sumber: <http://teorigraf-teknik-informatika.blogspot.com/2017/11/teori-graf.html>

Berdasarkan orientasi sisinya, graf dibedakan menjadi 2 jenis yaitu graf berarah dan graf tidak berarah. Graf berarah merupakan graf yang sisinya memiliki orientasi arah. Graf berarah dibagi menjadi 2 yaitu graf berarah yang tidak memiliki sisi ganda dan graf-ganda berarah yaitu graf berarah yang memiliki sisi ganda. Sedangkan graf tidak berarah merupakan graf yang sisinya tidak memiliki orientasi arah.



Gambar 2. (a) graf berarah, (b) graf-ganda berarah

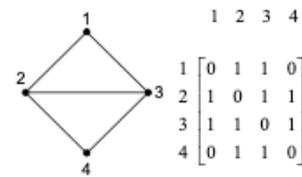
Sumber: <http://teorigraf-teknik-informatika.blogspot.com/2017/11/teori-graf.html>

Beberapa terminologi graf yang sering digunakan:

- Bersisian (*incidency*)**  
Sebuah sisi  $e = (v1, v2)$  dikatakan bersisian dengan simpul  $v1$  dan  $v2$ .
- Ketetangaan (*adjacent*)**  
Apabila kedua simpul terhubung secara langsung maka dapat dikatakan dua simpul tersebut bertetangga.
- Graf Kosong**  
Graf kosong merupakan graf yang tidak memiliki himpunan sisi.
- Simpul Terpencil**  
Simpul terpencil merupakan simpul yang tidak memiliki sisi yang bersisian dengan simpul tersebut.
- Derajat (*degree*)**  
Derajat merupakan jumlah sisi yang bersisian dengan simpul tersebut.
- Lintasan**  
Lintasan merupakan barisan berselang-seling dan sisi-sisi yang terbentuk.
- Siklus**  
Siklus merupakan lintasan yang berawal dan berakhir pada simpul yang sama
- Graf Berbobot**  
Graf berbobot adalah graf yang setiap sisinya memiliki suatu bobot/harga.
- Cut-set**  
*Cut-set* adalah himpunan sisi sisi yang apabila dihapus dari graf  $G$  akan membagi graf  $G$  menjadi tidak terhubung, lebih spesifiknya membagi graf  $G$  menjadi dua buah komponen.
- Upagraf**  
Apabila didefinisikan suatu graf  $G = (V, E)$  dan terdapat graf  $G_1 = (V_1, E_1)$  dengan  $V_1 \subseteq V$  dan  $E_1 \subseteq E$ ,  $G_1$  disebut sebagai *upagraf* dari  $G$ .

Sebuah graf tidak hanya direpresantisikan sebagai suatu gambar yang mempunyai sisi-sisi dan simpul-simpul. Graf bisa direpresentasikan dalam beberapa bentuk seperti:

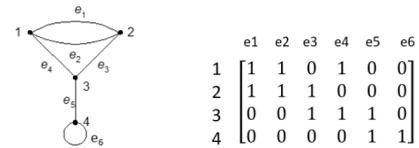
- Matriks ketetangaan.** Setiap baris dan kolom dari matriks ini disusun berdasarkan simpul dan elemen pada setiap elemen  $a_{ij}$  menyatakan hubungan dari simpul  $i$  dan simpul  $j$ . Pada matriks ini, nilai 0 menyatakan kedua simpul tidak terhubung, sedangkan nilai 1 menyatakan matriks tersebut terhubung.



Gambar 3 Matriks Ketetangaan

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

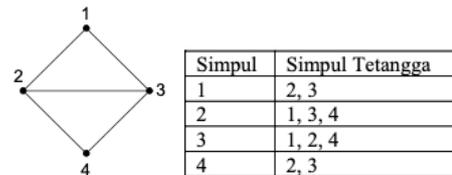
- Matriks berisian.** Baris ke- $i$  pada matriks ini menyatakan simpul dan kolom ke- $j$  pada matriks ini menyatakan sisi  $j$ . Elemen pada matriks ini menyatakan hubungan berisian antara simpul dengan sisinya. Nilai 1 pada setiap elemen menyatakan simpul ke  $i$  dan sisi ke  $j$  tersebut berisian, sedangkan nilai 0 menyatakan simpul dan sisi tersebut tidak berisian.



Gambar 4 Matriks Bersisian

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

- Senarai ketetangaan.** Representasi ini dituliskan dalam bentuk *list*. Representasi ini mencatat simpul-simpul yang merupakan tetangga dari sebuah simpul.

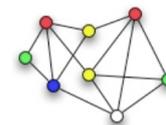


Gambar 5 Senarai Ketetangaan

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

### B. Pewarnaan Graf

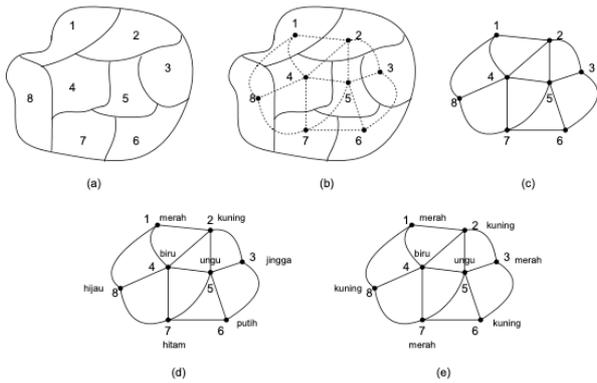
Ada dua macam pewarnaan pada graf, yaitu pewarnaan simpul dan pewarnaan sisi. Pada pewarnaan simpul, kita memberikan warna pada simpul-simpul graf sedemikian sehingga kedua simpul yang bertetangga memiliki warna yang berbeda. Pada pewarnaan graf, kita bisa mengaplikasikannya pada beberapa hal seperti peta wilayah. Peta diwarnai sedemikian sehingga dua wilayah yang bertetangga memiliki warna yang berbeda.



Gambar 6 Graf Berwarna

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

Untuk merepresentasikan sebuah wilayah pada peta menjadi graf ada beberapa tahapan. Kita bisa menyatakan sebuah wilayah sebagai simpul dan batas antar dua wilayah bertetangga sebagai sisi. Mewarnai wilayah pada peta berarti kita mewarnai simpul pada graf yang berkoresponden. Setiap wilayah yang bertetangga harus memiliki warna yang berbeda, yang mengakibatkan warna setiap simpul juga harus berbeda

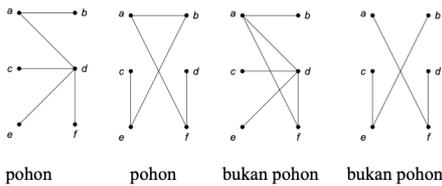


Gambar 7 (a) Peta, (b) Peta dan graf yang merepresentasikannya, (c) graf yang merepresentasikan peta, (d) pewarnaan simpul, setiap simpul memiliki warna berbeda

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

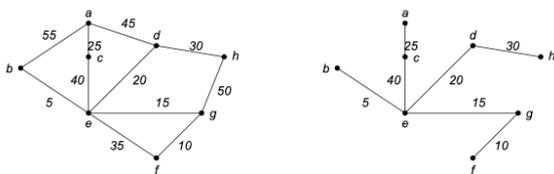
### C. Pohon

Pohon merupakan graf yang tidak memiliki arah dan tidak memiliki sirkuit. Pohon merentang merupakan pohon yang diperoleh dengan memutus sirkuit dari graf. Sebuah graf terhubung berbobot tidak hanya memiliki 1 pohon. Pohon merentang dengan bobot yang paling minimum yang dibuat dari sebuah graf disebut sebagai pohon merentang minimum. Untuk mendapatkan pohon merentang minimum, kita bisa menggunakan algoritma prim dan algoritma kruskal.



Gambar 8 Pohon

Sumber: Bahan Kuliah IF2120 Matematika Diskrit



Gambar 9 Pohon Merentang Minimum

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

### D. Algoritma Prim

Algoritma prim merupakan salah satu algoritma untuk mendapatkan pohon merentang minimum dari graf terhubung berbobot. Langkah pertama dalam algoritma prim yaitu mengambil sebuah sisi yang berbobot minimum dari graf kemudian dimasukkan ke pohon minimum yang akan kita buat misalkan pohon T. Kedua, pilih sebuah sisi (u,v) yang memiliki bobot minimum dan bersisian dengan simpul di T, tetapi (u,v) tidak membentuk sirkuit di T, masukkan sisi tersebut ke T. Kemudian ulangi langkah kedua sebanyak simpul - 1 kali. Untuk pseudocodenya sendiri dapat kita lihat pada gambar berikut:

```

procedure Prim(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung
berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
i, p, q, u, v : integer

Algoritma
Cari sisi (p,q) dari E yang berbobot terkecil
T ← {(p,q)}
for i←1 to n-2 do
  Pilih sisi (u,v) dari E yang bobotnya terkecil namun
  bersisian dengan simpul di T
  T ← T ∪ {(u,v)}
endfor

```

Gambar 10 Pseudocode Algoritma Prim

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

### E. Algoritma Kruskal

Algoritma Kruskal adalah algoritma untuk mencari pohon merentang minimum. Algoritma ini memiliki langkah mirip seperti algoritma prim. Langkah yang dilakukan dalam algoritma kruskal adalah menambahkan sebuah sisi (u,v) dari graf asal dengan bobot minimum dan tidak membentuk sirkuit di pohon yang akan kita cari secara bertahap (satu-persatu). Langkah tersebut akan diulangi sebanyak simpul - 2 kali. Dalam pseudocode, algoritma kruskal dapat dilihat pada gambar berikut:

```

procedure Kruskal(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung -
berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
i, p, q, u, v : integer

Algoritma
( Asumsi: sisi-sisi dari graf sudah diurut menaik
berdasarkan bobotnya - dari bobot kecil ke bobot
besar)
T ← {}
while jumlah sisi T < n-1 do
  Pilih sisi (u,v) dari E yang bobotnya terkecil
  if (u,v) tidak membentuk siklus di T then
    T ← T ∪ {(u,v)}
  endif
endfor

```

Gambar 11 Pseudocode Algoritma Kruskal

Sumber: Bahan Kuliah IF2120 Matematika Diskrit

## III. POHON MERENTANG MINIMUM DENGAN ALGORITMA PRIM DAN KRUSKAL

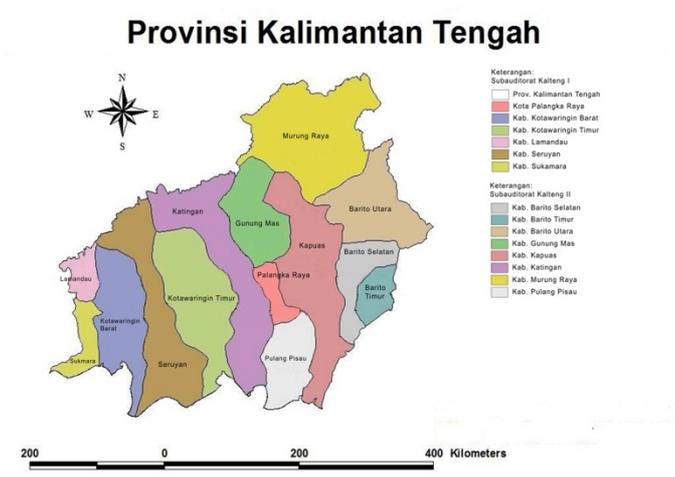
### A. Pengumpulan Data

Provinsi Kalimantan Tengah memiliki 13 kabupaten dan 1 kota. Berikut merupakan daftar kabupaten dan kota di Provinsi Kalimantan Tengah:

1. Kabupaten Barito Selatan
2. Kabupaten Barito Timur
3. Kabupaten Barito Utara
4. Kabupaten Gunung Mas
5. Kabupaten Kapuas
6. Kabupaten Katingan
7. Kabupaten Kotawaringin Barat
8. Kabupaten Kotawaringin Timur
9. Kabupaten Lamandau
10. Kabupaten Murung Raya

11. Kabupaten Pulang Pisau
12. Kabupaten Sukamara
13. Kabupaten Seruyan
14. Kota Palangka Raya

Berikut ini adalah peta batas administrasi kabupaten dan kota tersebut menurut Pejabat Pengelola Informasi dan Dokumentasi:



Gambar 12 Peta Procinsi Kalimantan Tengah

Sumber: <https://kalteng.bpk.go.id/peta-administrasi/>

Adapun jarak antar ibukota kabupaten/kota yang direpresentasikan dalam bentuk *adjacency matrix* graf berbobot berikut:

Tabel 1 Jarak Antar Ibukota Kabupaten/Kota di Provinsi Kalimantan Tengah yang Direpresentasikan dalam bentuk *adjacency matrix* graf berbobot

Sumber: Dokumentasi Penulis

Kode Kabupaten/Kota														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	0	0	237	110	452	0	0	0	0	0	0
2	0	0	0	0	0	0	0	230	139	0	0	0	0	0
3	0	0	0	0	468	0	0	0	0	44	0	0	553	142
4	0	0	325	0	143	0	0	0	0	0	0	93	0	0
5	0	0	468	143	0	0	0	0	0	0	0	0	85	0
6	237	0	0	0	0	0	136	0	0	0	0	0	0	0
7	110	0	0	0	0	136	0	0	0	0	0	0	0	0
8	452	230	0	0	0	0	0	0	369	0	0	0	0	0
9	0	139	0	0	0	0	0	369	0	186	268	0	0	88
10	0	0	44	0	0	0	0	0	186	0	0	0	0	98
11	0	0	0	0	0	0	0	0	268	0	0	0	525	180
12	0	0	0	93	0	0	0	0	0	0	0	0	0	0
13	0	0	553	0	85	0	0	0	0	0	525	0	0	0
14	0	0	142	0	0	0	0	88	98	180	0	0	0	0

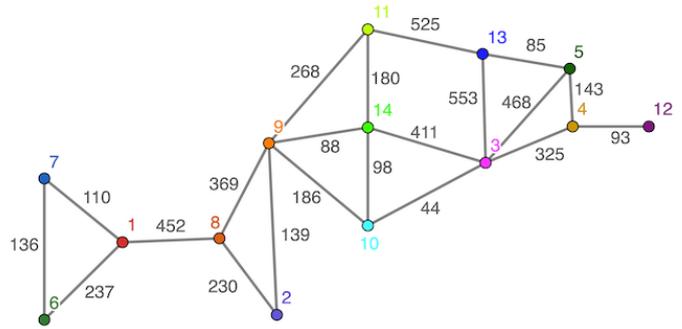
Kode	Nama Ibukota	Kode	Nama Ibukota
01	Pangkalan Bun	08	Kuala Pembuang
02	Sampit	09	Kasongan
03	Kuala Kapuas	10	Pulang Pisau
04	Buntok	11	Kuala Kurun
05	Muara Teweh	12	Tamiang Layang
06	Sukamara	13	Puruk Cahu
07	Nanga Bulik	14	Palangka Raya

Ibukota kabupaten/kota tersebut akan direpresentasikan menjadi simpul dalam graf berbobot tidak berarah dan jarak antar ibukota kabupaten/kota tersebut akan direpresentasikan menjadi sisi yang memiliki bobot.

### B. Representasi Ibukota Kabupaten/Kota dalam Bentuk Graf Berbobot Berwarna dan Tidak Berarah

Dari tabel 1 dan gambar 1, akan dibuat sebuah graf berbobot berwarna dan tidak berarah dengan simpulnya merupakan ibukota kabupaten/kota dan sisinya merupakan jarak

antardaerah dalam satuan kilometer. Berikut ini merupakan gambar graf berbobot tak berarah yang dibuat dengan menghilangkan label bobot dari graf untuk memudahkan pembacaan.



Gambar 13 Representasi Ibukota Kabupaten/Kota dalam Graf Berbobot Tidak Berarah

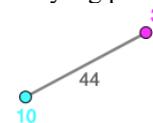
Sumber: Dokumentasi Penulis

Keterangan:

1. Pangkalan Bun
2. Sampit
3. Kuala Kapuas
4. Buntok
5. Muara Teweh
6. Sukamara
7. Nanga Bulik
8. Kuala Pembuang
9. Kasongan
10. Pulang Pisau
11. Kuala Kurun
12. Tamiang Layang
13. Puruk Cahu
14. Palangka Raya

### C. Pohon Merentang Minimum Menggunakan Algoritma Prim

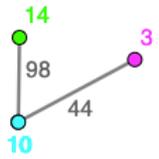
Algoritma prim adalah salah satu cara yang dapat digunakan untuk membuat pohon merentang minimum. Langkah pertama yang harus dilakukan yaitu mengambil sisi dari graf yang memiliki bobot paling minimum, dalam hal ini kita bisa mengambil sisi yang menghubungkan node (10) dengan (3) dikarenakan memiliki bobot yang paling kecil yaitu 44.



Gambar 14 Langkah Pertama Algoritma Prim

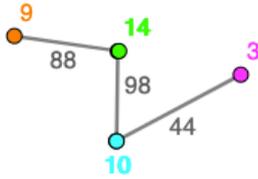
Sumber: Dokumentasi Penulis

Langkah berikutnya adalah mengambil satu sisi dengan bobot terkecil yang bersisian dengan salah satu simpul dari pohon yang kita buat dan tidak membentuk sirkuit di pohon.



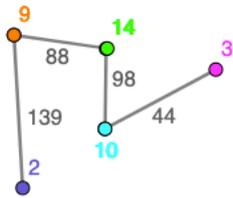
Gambar 15 Langkah Kedua Algoritma Prim

Sumber: Dokumentasi Penulis



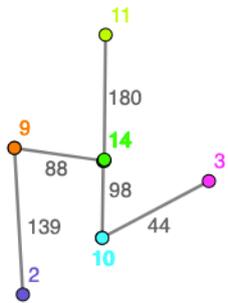
Gambar 16 Langkah Ketiga Algoritma Prim

Sumber: Dokumentasi Penulis



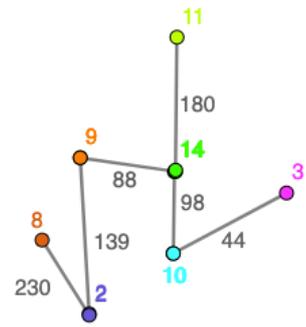
Gambar 17 Langkah Keempat Algoritma Prim

Sumber: Dokumentasi Penulis



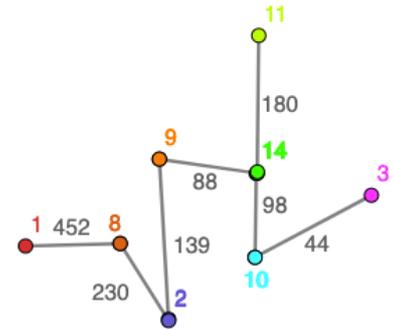
Gambar 18 Langkah Kelima Algoritma Prim

Sumber: Dokumentasi Penulis



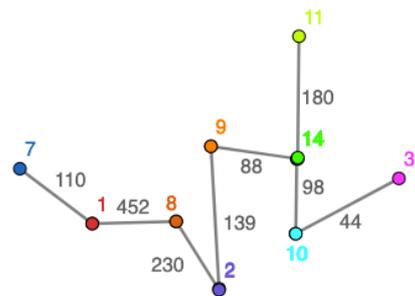
Gambar 19 Langkah Keenam Algoritma Prim

Sumber: Dokumentasi Penulis



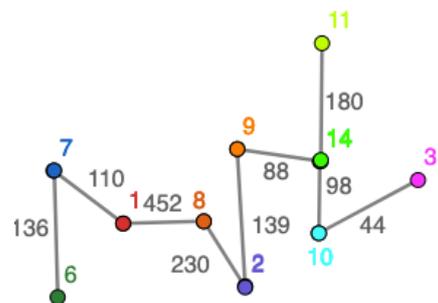
Gambar 20 Langkah Ketujuh Algoritma Prim

Sumber: Dokumentasi Penulis



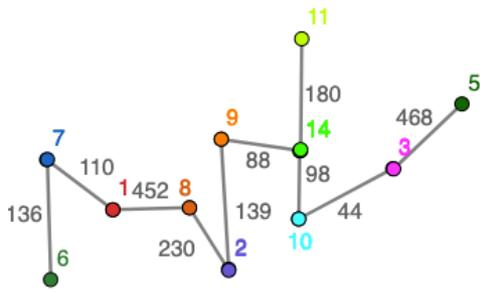
Gambar 21 Langkah Kedelapan Algoritma Prim

Sumber: Dokumentasi Penulis



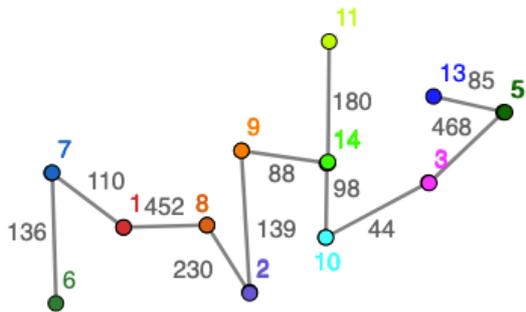
Gambar 22 Langkah Kesembilan Algoritma Prim

Sumber: Dokumentasi Penulis



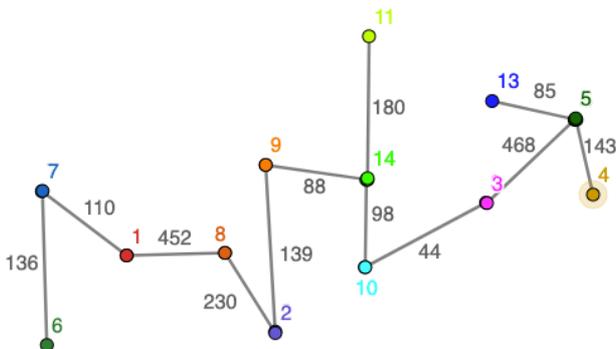
Gambar 23 Langkah Kesepuluh Algoritma Prim

Sumber: Dokumentasi Penulis



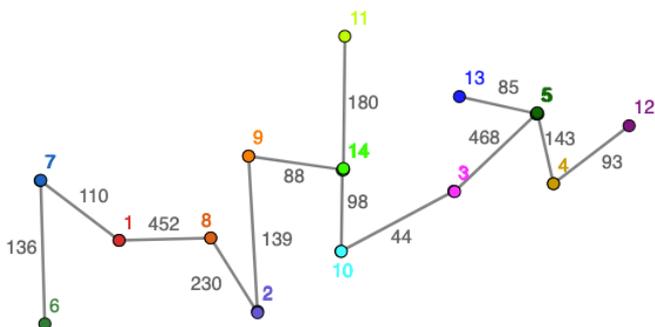
Gambar 24 Langkah Kesebelas Algoritma Prim

Sumber: Dokumentasi Penulis



Gambar 25 Langkah Keduabelas Algoritma Prim

Sumber: Dokumentasi Penulis

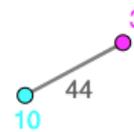


Gambar 26 Langkah ketigabelas Algoritma Prim

Sumber: Dokumentasi Penulis

#### D. Pohon Merentang Minimum Menggunakan Algoritma Kruskal

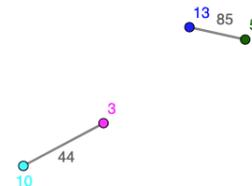
Algoritma kruskal merupakan salah satu algoritma alternatif yang dapat dipakai untuk membuat pohon merentang minimum. Algoritma yang dipakai mirip seperti algoritma prim, akan tetapi ada sedikit perbedaan. Untuk membuat sebuah pohon merentang minimum dengan algoritma kruskal, kita harus memilih sebuah sisi yang paling minimum. Pohon awalnya kosong, karena tidak ada tol yang dibangun di Provinsi Kalimantan Tengah. Kemudian, kita ambil sebuah sisi dengan bobot paling minimum dan tidak membentuk sebuah sirkuit di pohon. Tambahkan sisi tersebut ke pohon dan ulangi langkah tersebut sebanyak 13 kali.



Gambar 27 Langkah Pertama Algoritma Kruskal

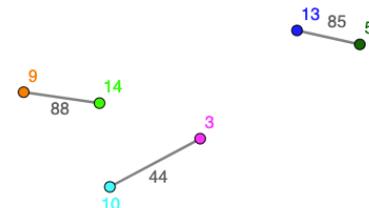
Sumber: Dokumentasi Penulis

l



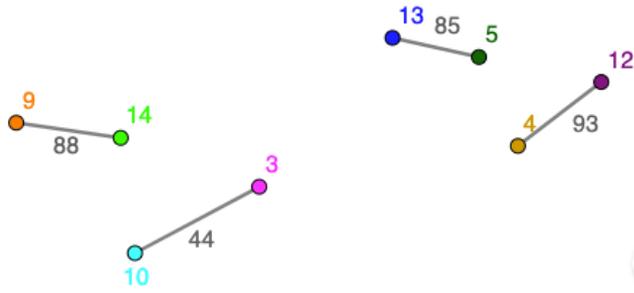
Gambar 28 Langkah Kedua Algoritma Kruskal

Sumber: Dokumentasi Penulis

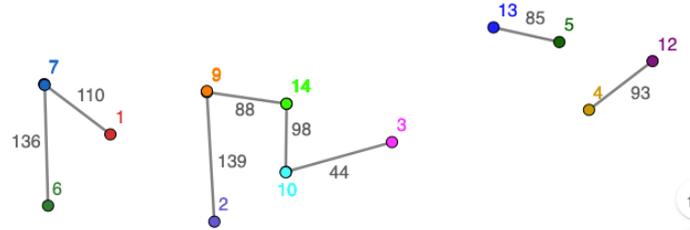


Gambar 29 Langkah Ketiga Algoritma Kruskal

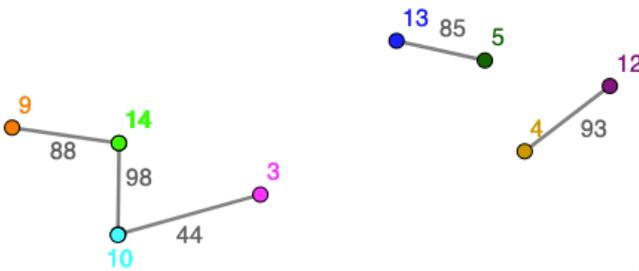
Sumber: Dokumentasi Penulis



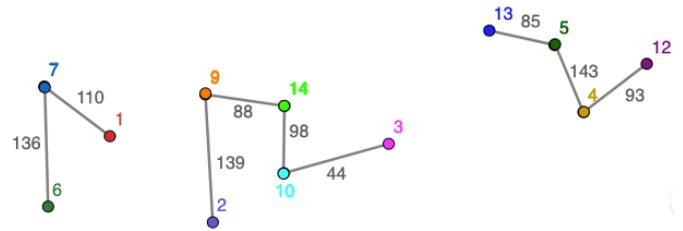
Gambar 30 Langkah Keempat Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



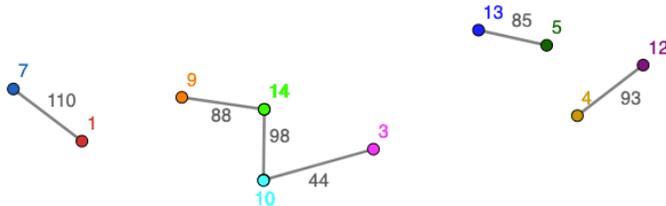
Gambar 34 Langkah Kedelapan Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



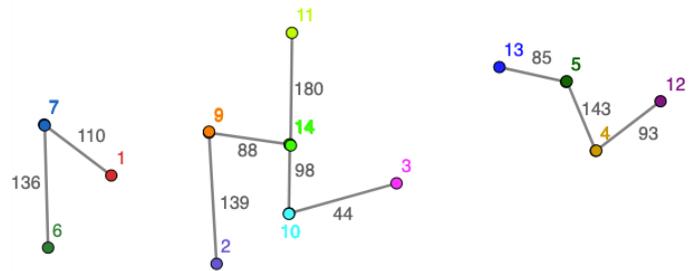
Gambar 31 Langkah Kelima Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



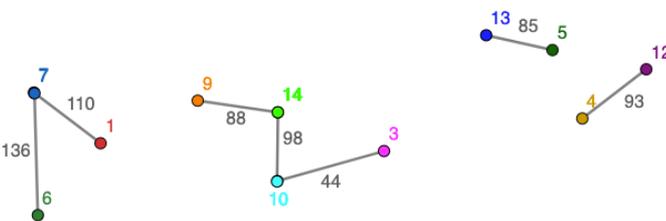
Gambar 35 Langkah Kesembilan Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



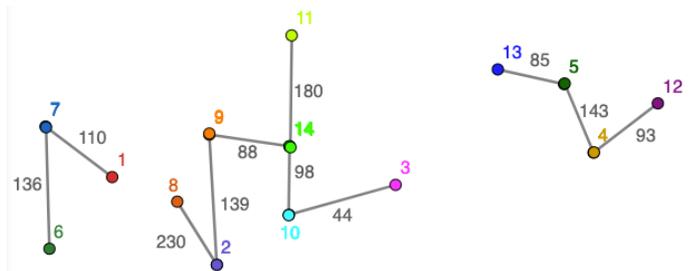
Gambar 32 Langkah Keenam Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



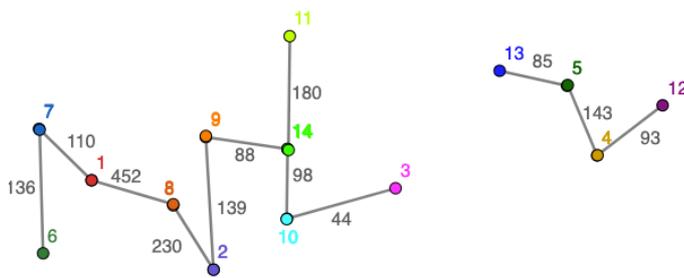
Gambar 36 Langkah Kesepuluh Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



Gambar 33 Langkah Ketujuh Algoritma Kruskal  
 Sumber: Dokumentasi Penulis

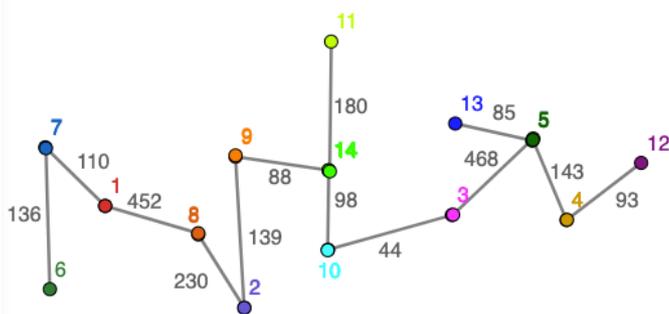


Gambar 37 Langkah Kesebelas Algoritma Kruskal  
 Sumber: Dokumentasi Penulis



Gambar 38 Langkah Keduabelas Algoritma Kruskal

Sumber: Dokumentasi Penulis



Gambar 39 Langkah Ketigabelas Algoritma Kruskal

Sumber: Dokumentasi Penulis

Dapat kita lihat pada gambar 28 yang merupakan bentuk akhir dari pohon merentang minimum rute jalan tol di Provinsi Kalimantan Tengah yang diselesaikan dengan algoritma kruskal. Apabila kita bandingkan dengan gambar 15 yang merupakan bentuk akhir dari pohon merentang minimum rute jalan tol di Provinsi Kalimantan Tengah yang diselesaikan dengan algoritma Prim, kita memperoleh hasil yang sama. Node dalam hal ini merupakan ibukota kabupaten/kota yang dilalui dan sisi berbobot yang dalam hal ini merupakan jalur antar ibukota kabupaten/kota pada pohon merentang minimum dengan menggunakan algoritma prim dan algoritma kruskal hasilnya sama. Total bobot dari pohon merentang minimum yang tersebut adalah 2266 dalam satuan kilometer.

#### IV. BEBERAPA KESALAHAN UMUM

Terdapat beberapa kendala yang penulis hadapi selama penulisan makalah ini. Dalam hal pencarian jarak antar ibukota kabupaten/kota, penulis mengambil jarak terpendek antar ibukota kabupaten/kota, sama halnya dengan menarik sebuah garis lurus yang memiliki bobot berupa jarak antar ibukota kabupaten/kota dalam satuan kilometer. Pada kenyataannya, belum tentu pembangunan tol dapat efisien jika menarik garis lurus, karena mungkin saja terdapat sungai atau rumah penduduk. Apabila pemerintah hendak membangun tol melewati sebuah sungai, pemerintah harus menyiapkan biaya ekstra untuk membangun jembatan. Selain itu, apabila tol yang hendak dibangun melewati rumah penduduk, pemerintah harus menyiapkan kompensasi dana dan melakukan relokasi untuk

penduduk yang terdampak pembangunan dari jalan tol itu sendiri. Dalam hal ini, kedua hal tersebut menambah anggaran pembangunan, sehingga pembangunan tol menjadi tidak efisien yang mengakibatkan pembangunan jalan tol menjadi tidak optimal.

#### V. KESIMPULAN

Beberapa permasalahan dapat direpresentasikan dalam bentuk graf. Salah satunya seperti yang penulis sampaikan dalam makalah ini, yaitu mencari sebuah rute paling minimum yang menghubungkan semua titik. Dengan memodelkan peta Provinsi Kalimantan Tengah dengan sebuah graf, kita bisa menemukan rute tol paling minimum yang menghubungkan semua titik dengan menggunakan algoritma prim dan kruskal. Rute tol yang terpendek ini, diharap bisa memberikan manfaat bagi banyak pihak, dengan rute tol yang terpendek ini tentu pemerintah juga akan mengeluarkan biaya yang lebih sedikit untuk membangun jalan tol. Selain itu, dengan rute terpendek ini, akan memberikan efisiensi bagi pengguna jalan tol, dalam hal ini rute terpendek juga dapat menghemat bahan bakar kendaraan bermotor, karena semakin pendek rute semakin cepat pula pengguna jalan sampai ke tempat tujuan.

#### VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak yang sudah membantu dalam menyelesaikan makalah ini. Khususnya kepada Allah Subhanahu wa ta'ala yang sudah memberikan kesehatan dan kesempatan kepada penulis untuk menulis makalah ini. Selain itu, penulis secara khusus mengucapkan terima kasih kepada keluarga dan Bapak Dr. Ir. Rinaldi Munir, MT. selaku dosen pengampu mata kuliah IF2120 Matematika Diskrit kelas 01. Terima kasih kepada pembuat referensi yang sudah berkenan untuk berbagi ilmu.

#### DAFTAR PUSTAKA

- [1] Munir, Rinaldi. 2021. Pohon (Bag. 1), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>, diakses pada 14 Desember 2021.
- [2] Munir, Rinaldi. 2021. Pohon (Bag. 2), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>, diakses pada 14 Desember 2021.
- [3] Munir, Rinaldi. 2021. Pohon (Bag. 2), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bag2.pdf>, diakses pada 14 Desember 2021.
- [4] Munir, Rinaldi. 2021. Pohon (Bag. 2), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bag2.pdf>, diakses pada 14 Desember 2021.
- [5] Anonim, 2021. <http://teorigraf-teknik-informatika.blogspot.com/2017/11/teori-graf.html>, diakses pada 14 Desember 2021
- [6] Badan Pemeriksa Keuangan. 2021. <https://kalteng.bpk.go.id/peta-administrasi/>, diakses pada 12 Desember 2021.
- [7] Website Pemerintah Daerah Kalimantan Tengah. 2021. <https://kalteng.go.id/page/106/website-kab-kota>, diakses pada 12 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

A handwritten signature in black ink, appearing to read 'Fikri' with a stylized flourish below it.

Fikri Khoiron Fadhila 13520056