

Aplikasi Pohon Merentang Minimum Dalam Pembuatan Jalan Tol di Kalimantan Timur

Jordan Daniel Joshua – 13519098¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13519098@std.stei.itb.ac.id

Abstract—Kalimantan Timur merupakan provinsi yang terpilih menjadi Ibu Kota Indonesia yang baru pada tahun 2019. Bisa dipastikan pembangunan infrastruktur di Kalimantan Timur akan menjadi target utama dari pemerintah dalam 10 tahun ke depan. Salah satu infrastruktur yang menjadi target utama pemerintah adalah jalan tol. Pembangunan jalan tol ini bertujuan untuk memudahkan mobilitas antar kabupaten atau antar kota di Kalimantan Timur. Adapun pembangun jalan tol di Kalimantan Timur dapat diaplikasikan menggunakan pohon merentang minimum untuk membuat jalan tol yang menghubungkan semua daerah di Kalimantan Timur dan efisien.

Keywords—Jalan Tol, Jalur Kereta Api, Kalimantan Timur, Pohon Merentang Minimum

I. PENDAHULUAN

Pembangunan infrastruktur merupakan fokus utama dari pemerintahan Indonesia selama beberapa tahun terakhir. Salah satu infrastruktur yang pembangunannya paling diincar oleh pemerintah adalah jalan tol. Hal tersebut dikarenakan jalan tol sangat penting dalam memudahkan mobilisasi dari 1 daerah ke daerah lainnya yang memudahkan persebaran sumber daya alam maupun sumber daya manusia. Pembangunan jalan tol ini sangat penting untuk berbagai daerah, terutama daerah Kalimantan Timur yang akan menjadi Ibu Kota baru dari Indonesia menggantikan DKI Jakarta. Diharapkan pembangunan jalan tol di Kalimantan Timur dapat memudahkan transportasi antar daerah yang selama ini selalu menjadi masalah di Kalimantan Timur mengingat hanya ada 1 jalan tol di Kalimantan Timur yaitu jalan tol yang menghubungkan kota Balikpapan dengan kota Samarinda.



Gambar 1 Jalan Tol Balikpapan-Samarinda

Sumber: <https://kaltimtoday.co/pertama-di-kalimantan-ini-5-fakta-tol-balikpapan-samarinda/>

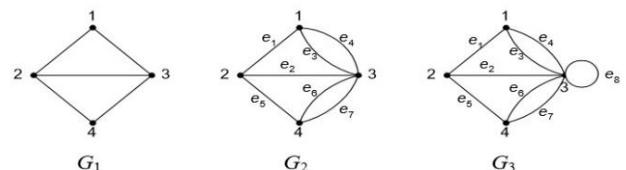
Pembangunan jalan tol di Kalimantan Timur ini diharapkan dapat menghubungkan semua kabupaten dan kota di Kalimantan Timur yaitu Kabupaten Berau, Kabupaten Kutai Barat, Kabupaten Kutai Kartanegara, Kabupaten Kutai Timur, Kabupaten Mahakam Ulu, Kabupaten Paser, Kabupaten Penajam Paser Utara, Kota Balikpapan, Kota Bontang, dan Kota Samarinda. Untuk membuat sebuah jalan tol yang efisien (jalur terpendek) dan menghubungkan semua kabupaten dan kota tersebut, dibutuhkan pengaplikasian dari Graf dan Pohon. Penyelesaian dilakukan dengan membuat sebuah graf terhubung berbobot yang simpul-simpulnya berupa kota dan kabupaten yang ada dengan jarak antardaerah sebagai bobot dari graf. Graf tersebut kemudian dibuat menjadi pohon merentang minimum dengan menggunakan algoritma Prim atau algoritma Kruskal.

II. LANDASAN TEORI

A. Graf

Graf adalah representasi visual dari objek-objek diskrit dan hubungan antara objek-objek tersebut. Adapun bentuk matematis dari sebuah graf adalah $G = (V, E)$ dengan V sebagai himpunan dari simpul (*vertices*) Graf dan E sebagai himpunan sisi (*edges*) yang menghubungkan sepasang simpul Graf.

Berdasarkan keberadaan dari gelang atau sisi ganda pada graf, graf dapat digolongkan menjadi 2 jenis yaitu graf sederhana dan graf tak-sederhana. Graf sederhana adalah graf yang tidak mengandung gelang ataupun sisi ganda. Graf tak-sederhana adalah graf yang mengandung gelang atau sisi ganda. Graf tak-sederhana pun dapat digolongkan menjadi 2 yaitu graf ganda (mengandung sisi ganda) dan graf semu (mengandung gelang).

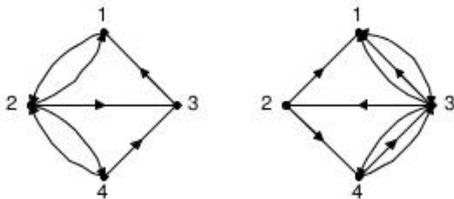


Gambar 2. (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Gambar 2 Graf Sederhana, Ganda, dan Semu

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

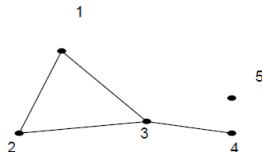
Berdasarkan orientasi pada sisinya, graf dibedakan menjadi 2 jenis yaitu graf tak-berarah dan graf berarah. Graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah sedangkan graf berarah adalah graf yang sisinya mempunyai orientasi arah. Graf berarah terbagi menjadi 2 yaitu graf berarah (tidak memiliki sisi ganda) dan graf-ganda berarah (memiliki sisi ganda).



Gambar 3 Graf Berarah dan Graf-Ganda Berarah
 Sumber: <http://septian-sisinfak.blogspot.com/2011/10/jenis-jenis-graf.html>

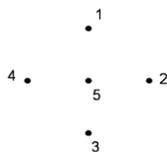
Adapun terminologi-terminologi yang sering digunakan pada graf sebagai berikut:

- Ketetanggaan (Adjacent)**
 Dua buah simpul dikatakan bertetangga apabila kedua simpul tersebut terhubung langsung.
- Bersisian (Incidency)**
 Sebuah sisi $e = (v_1, v_2)$ dapat dikatakan bersisian dengan simpul v_1 dan v_2 .
- Simpul Terpencil (Isolated Vertex)**
 Istilah untuk simpul yang tidak memiliki sisi yang bersisian dengan simpul tersebut. Pada gambar di bawah, simpul 5 merupakan simpul terpencil karena tidak memiliki sisi yang bersisian dengannya.



Gambar 4 Simpul Terpencil
 Sumber: <http://rabbittjeyek.blogspot.com/2011/12/teori-graf-6.html>

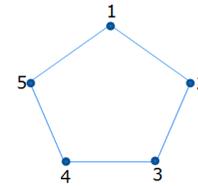
- Graf Kosong (Null/Empty Graph)**
 Sebuah graf dikatakan kosong apabila tidak memiliki himpunan sisi atau himpunan sisinya kosong.



Gambar 5 Graf Kosong
 Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

- Derajat (Degree)**
 Derajat sebuah simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Adapun jumlah derajat pada sebuah graf haruslah genap. Pada gambar di bawah, masing-masing simpul memiliki derajat 2. Karena jumlah derajat pada gambar di bawah adalah 10 yang berupa genap maka gambar di bawah adalah

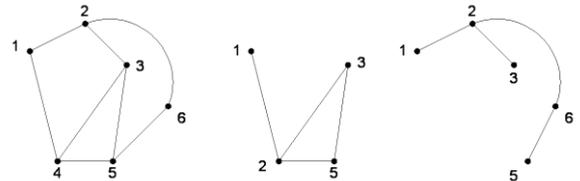
sebuah graf.



Gambar 6 Sembarang Graf

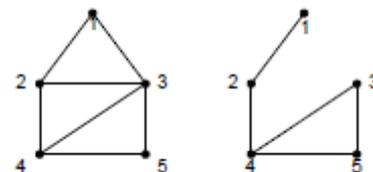
Sumber: <https://bermatematika.net/2016/05/20/graf-dan-komplemennya/>

- Lintasan (Path)**
 Lintasan yang panjangnya n dari awal simpul v_0 sampai ke simpul tujuan v_n di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang terbentuk. Panjang dari lintasan sendiri adalah jumlah sisi dalam lintasan tersebut. Lintasan 1,2,3,4 pada gambar 6 memiliki panjang 4.
- Siklus (Cycle) atau Sirkuit (Circuit)**
 Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Pada gambar 6, 1,2,3,4,5,1 adalah contoh sirkuit dengan panjang 5.
- Upagraf dan Komplemen Upagraf**
 Misalkan $G = (V,E)$ adalah sebuah graf. $G_1 = (V_1,E_1)$ adalah upagraf dari G jika V_1 tergabung dalam V dan E_1 tergabung dalam E . Komplemen dari upagraf G_1 terhadap graf G adalah $G_2 = (V_2,E_2)$ sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang anggota-anggotanya bersisian dengannya.



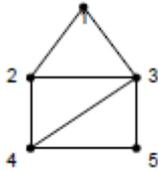
Gambar 7 Graf G , Upagraf G , dan Komplemennya
 Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

- Upagraf Merentang (Spanning Subgraph)**
 Upagraf $G_1 = (V_1,E_1)$ dari $G = (V,E)$ dikatakan upagraf merentang jika G_1 mengandung semua simpul dari G .



Gambar 8 Graf G dan Upagraf Merentangnya
 Sumber: <https://ayobelajarbos.blogspot.com/2019/06/graf-dan-pohon.html>

- Cut-Set**
 Cut set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung atau terbagi menjadi 2 komponen. Pada graf G di gambar 9, himpunan $\{(4,5), (3,5)\}$ adalah cut-set. Terdapat banyak cut-set dalam sebuah graf.

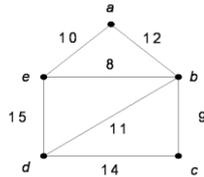


Gambar 9 Graf G

Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

11. Graf Berbobot (Weighted Graph)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah nilai atau bobot.



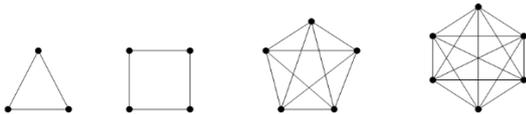
Gambar 10 Graf Berbobot

Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

Ada beberapa jenis graf khusus yang sering digunakan atau sering muncul seperti:

1. Graf Lengkap (Complete Graph)

Graf lengkap adalah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf lengkap dengan n buah simpul dilambangkan dengan K_n . Jumlah sisi pada graf lengkap adalah $n(n-1)/2$



Gambar 11 Graf Lengkap

Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

2. Graf Lingkaran

Graf lingkaran adalah graf sederhana yang setiap simpulnya berderajat dua. Graf lingkaran dengan jumlah simpul n dilambangkan dengan C_n .

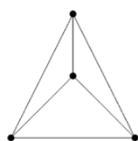


Gambar 12 Graf Lingkaran

Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

3. Graf Teratur (Regular Graph)

Graf teratur adalah graf yang setiap simpulnya memiliki derajat yang sama antara yang satu dengan yang lain. Jumlah simpul pada graf teratur dengan n buah simpul dan derajat r adalah $nr/2$.



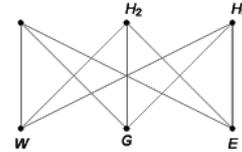
Gambar 13 Graf Teratur

Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

[graph_21.html](http://sha-essa.blogspot.com/2011/12/teori-graph_21.html)

4. Graf Bipartite (Bipartite Graph)

Graf bipartite adalah graf yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian. Graf bipartite yang setiap sisi pada G menghubungkan antara simpul di V_1 ke simpul di V_2 disebut graf bipartite dan dinyatakan sebagai $G(V_1, V_2)$.



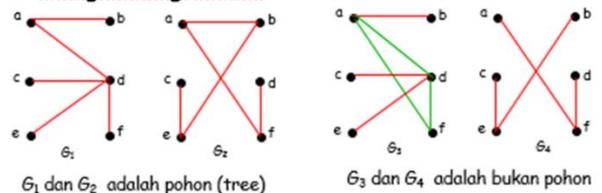
Gambar 14 Graf Bipartite

Sumber: http://sha-essa.blogspot.com/2011/12/teori-graph_21.html

B. Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Adapun sifat-sifat atau properti dari pohon dengan simpul n sebagai berikut:

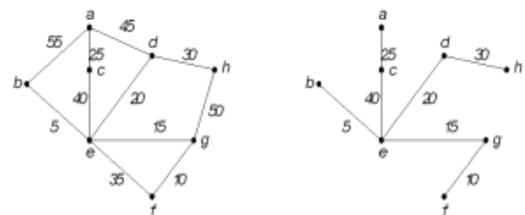
1. Graf tak-berarah sederhana $G = (V, E)$ adalah sebuah pohon
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal
3. G terhubung dan memiliki $m = n - 1$ buah sisi
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi
5. G tidak mengandung sirkuit dan penambahan 1 sisi pada pohon akan mengakibatkan 1 sirkuit pada pohon
6. G terhubung dan semua sisinya adalah jembatan



Gambar 15 Contoh Pohon

Sumber: <http://poetra70.blogspot.com/2015/09/pohon-matematika-diskrit.html>

Pohon merentang dari graf terhubung adalah upagraf merentang yang berupa pohon. Pohon merentang diperoleh dengan memotong sirkuit di dalam graf. Sebuah graf terhubung berbobot bisa memiliki lebih dari 1 pohon. Pohon merentang yang berbobot minimum yang dibuat dari graf tersebut dinamakan pohon merentang minimum. Ada dua cara untuk mendapatkan pohon merentang minimum dari sebuah graf. Cara yang pertama adalah Algoritma Prim dan cara yang kedua adalah Algoritma Kruskal



Gambar 16 Pohon Merentang Minimum

Sumber: <http://student.blog.dinus.ac.id/jonathanph/2019/01/01/alg>

C. Algoritma Prim

Algoritma Prim adalah salah satu cara untuk mengubah graf-terhubung berbobot menjadi sebuah pohon merentang minimum. Adapun langkah-langkah dalam Algoritma Prim 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.

D. Algoritma Kruskal

Algoritma Kruskal adalah cara selain Algoritma Prim untuk mengubah graf-terhubung berbobot menjadi sebuah pohon merentang minimum. Adapun langkah-langkah dalam Algoritma Prim sebagai berikut:

1. Sisi-sisi dari graf sudah diurutkan menaik berdasarkan bobotnya dari kecil ke besar
2. T masih kosong
3. Pilih sisi (u,v) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (u,v) ke dalam T.
4. Ulangi langkah 2 sebanyak n-1 kali.



Gambar 17 Peta Kalimantan Timur

Sumber: <https://bkd.kaltimprov.go.id/profil/bkd-kabupaten-kota-sekaltim>

Adapun table jarak antara daerah yang 1 dengan daerah yang lain berdasarkan list kabupaten dan kota di atas adalah sebagai berikut:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 1 | | | | | | | | | | |
| 2 | 198 | | | | | | | | | |
| 3 | 328 | 146 | | | | | | | | |
| 4 | 315 | 123 | 79 | | | | | | | |
| 5 | 341 | 145 | 114 | 38 | | | | | | |
| 6 | 314 | 142 | 190 | 113 | 89 | | | | | |
| 7 | 243 | 89 | 191 | 129 | 126 | 64 | | | | |
| 8 | 321 | 190 | 257 | 190 | 166 | 78 | 141 | | | |
| 9 | 252 | 196 | 318 | 254 | 243 | 161 | 222 | 111 | | |
| 10 | 318 | 320 | 447 | 383 | 370 | 284 | 347 | 222 | 129 | |

Tabel 1 Jarak Antardaerah di Kalimantan Timur (Km)

Sumber: Google Map dan Topovani

III. PENGUMPULAN DATA

Kalimantan Timur memiliki 7 kabupaten dan 3 kota. Daftar kabupaten dan kota tersebut sebagai berikut:

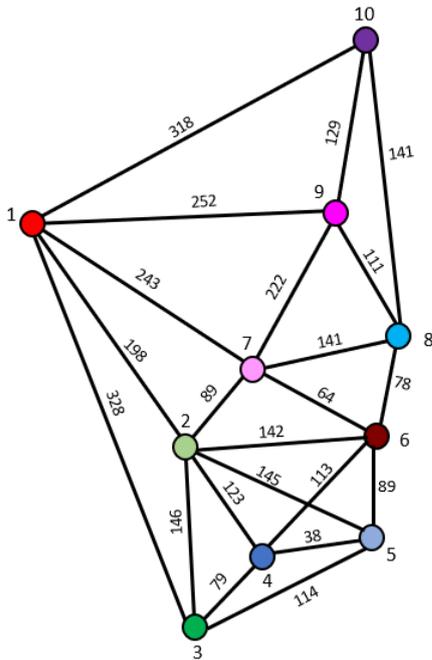
1. Kabupaten Mahakam Ulu
2. Kabupaten Kutai Barat
3. Kabupaten Paser
4. Kabupaten Penajam Paser Utara
5. Kota Balikpapan
6. Kota Samarinda
7. Kabupaten Kutai Kartanegara
8. Kota Bontang
9. Kabupaten Kutai Timur
10. Kabupaten Berau

Adapun letak dari kabupaten dan kota tersebut di peta sebagai berikut:

IV. POHON MERENTANG MINIMUM DALAM PEMBUATAN JALAN TOL

A. Pembuatan Graf Jalan Tol

Dari tabel 1, dapat dibuat sebuah graf berbobot tak berarah dengan simpul berupa kota atau kabupaten dan bobot dari sisi-sisi yang menghubungkan simpul-simpul tersebut berupa jarak antardaerah dalam satuan km. Untuk mempermudah pembacaan dan pembuatan graf, dihilangkan beberapa sisi yang tidak akan mempengaruhi pembuatan pohon merentang. Adapun graf yang dibuat sebagai berikut.



Gambar 18 Graf Berbobot Jarak Kabupaten dan Kota di Kalimantan Timur

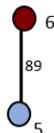
Sumber: dokumentasi pribadi

Keterangan:

- | | |
|-----------------------------|---------------------------|
| 1. Kab. Mahakam Ulu | 6. Kota Samarinda |
| 2. Kab. Kutai Barat | 7. Kab. Kutai Kartanegara |
| 3. Kab. Paser | 8. Kota Bontang |
| 4. Kab. Penajam Paser Utara | 9. Kab. Kutai Timur |
| 5. Kota Balikpapan | 10. Kab. Berau |

B. Pembuatan Pohon Merentang Minimum Menggunakan Algoritma Prim

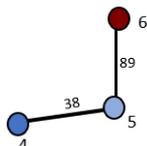
Pembuatan pohon merentang minimum dari sebuah graf dapat menggunakan cara Algoritma Prim. Adapun beberapa langkah yang harus dilakukan dalam membuat pohon merentang minimum. Pertama-tama kita ambil sisi yang menghubungkan Kota Balikpapan dengan Kota Samarinda karena sudah ada jalan tol yang menghubungkan kedua daerah tersebut.



Gambar 19 Langkah Pertama Algoritma Prim

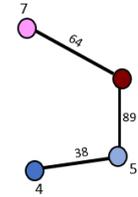
Sumber: dokumentasi pribadi

Setelah itu kita ambil sisi dengan bobot terkecil yang bersisian dengan salah satu dari simpul yang sudah ada di dalam pohon tersebut dan kita tambahkan ke dalam pohon merentang. Langkah di atas kita ulang sampai semua simpul yang ada di graf sudah terhubung.



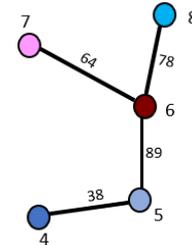
Gambar 20 Langkah Kedua Algoritma Prim

Sumber: dokumentasi pribadi



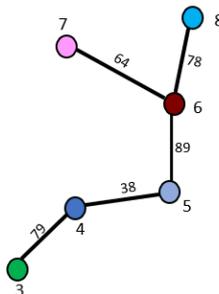
Gambar 21 Langkah Ketiga Algoritma Prim

Sumber: dokumentasi pribadi



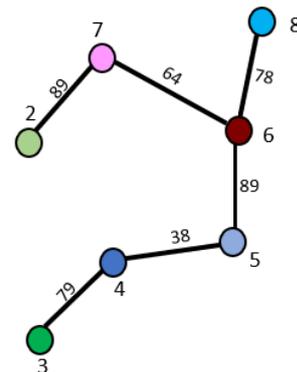
Gambar 22 Langkah Keempat Algoritma Prim

Sumber: dokumentasi pribadi



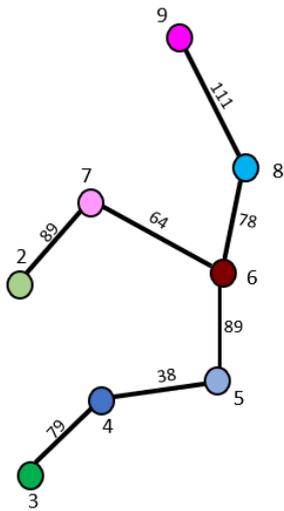
Gambar 23 Langkah Kelima Algoritma Prim

Sumber: dokumentasi pribadi

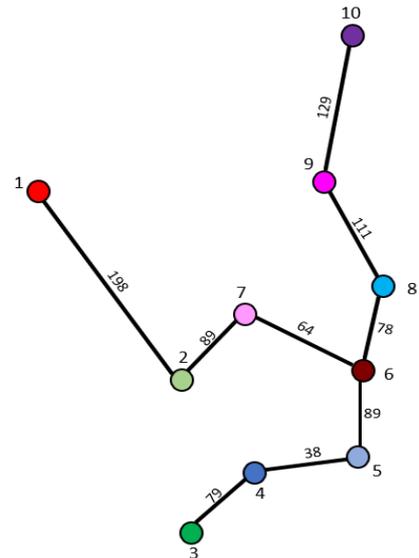


Gambar 24 Langkah Keenam Algoritma Prim

Sumber: dokumentasi pribadi



Gambar 24 Langkah Ketujuh Algoritma Prim
 Sumber: dokumentasi pribadi



Gambar 27 Langkah Kesembilan Algoritma Prim
 Sumber: dokumentasi pribadi

Gambar 27 merupakan gambar pohon merentang minimum yang dibentuk dari Algoritma Prim. Total langkah yang diperlukan dalam membuat pohon merentang minimum menggunakan algoritma prim sebanyak 9 langkah dengan bobot total dari pohon merentang minimum sebesar 875 km.

C. Pembuatan Pohon Merentang Minimum Menggunakan Algoritma Kruskal

Pembuatan pohon merentang minimum dapat juga menggunakan Algoritma Kruskal. Pembuatannya menyerupai Algoritma Prim dengan beberapa perbedaan. Adapun langkah-langkah dalam Algoritma Kruskal sebagai berikut. Pertama-tama, kita urutkan sisi-sisi pada graf dari bobot terkecil ke bobot terbesar.

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Sisi | (4,5) | (6,7) | (6,8) | (3,4) | (2,7) | (5,6) | (8,9) | (4,6) |
| Bobot | 38 | 64 | 78 | 79 | 89 | 89 | 111 | 113 |

Tabel 2 Sisi 1-8 Graf Berdasarkan Bobotnya
 Sumber: dokumentasi pribadi

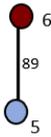
| | | | | | | | |
|-------|-------|-------|--------|-------|--------|-------|-------|
| Sisi | (3,5) | (2,4) | (9,10) | (7,8) | (8,10) | (2,6) | (2,5) |
| Bobot | 114 | 123 | 129 | 141 | 141 | 142 | 145 |

Tabel 3 Sisi 1-8 Graf Berdasarkan Bobotnya
 Sumber: dokumentasi pribadi

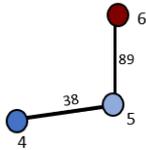
| | | | | | | | |
|-------|-------|-------|-------|-------|-------|--------|-------|
| Sisi | (2,3) | (1,2) | (7,9) | (1,7) | (1,9) | (1,10) | (1,3) |
| Bobot | 146 | 198 | 222 | 243 | 252 | 318 | 328 |

Tabel 2 Sisi 1-8 Graf Berdasarkan Bobotnya
 Sumber: dokumentasi pribadi

Kondisi awal pohon merentang tidak kosong karena sudah ada jalan tol dari Kota Balikpapan ke kota Samarinda sehingga pohon merentang minimum berisi sisi (5,6). Setelah itu kita ambil sisi dengan bobot terkecil. Apabila sisi tersebut membuat sebuah sirkuit di dalam pohon merentang minimum maka sisi tersebut kita tolak dan tidak kita masukkan ke dalam pohon merentang. Tinjau sisi (4,5) dengan bobot 3. Sisi (4,5) diterima karena tidak membentuk sirkuit di dalam pohon merentang minimum.

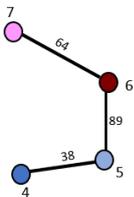


Gambar 28 Kondisi Awal Pohon Merentang
 Sumber: dokumentasi pribadi

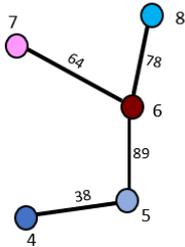


Gambar 29 Sisi (4,5) Diterima
 Sumber: dokumentasi pribadi

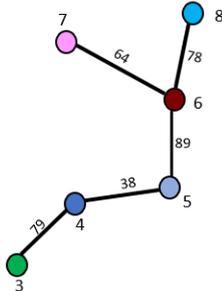
Setelah itu kita tinjau lagi sisi-sisi berikutnya apakah sisi tersebut diterima atau tidak. Setelah kita tinjau, didapatkan bahwa sisi (6,7), (6,8), (3,4), (2,7), (8,9) diterima. Sedangkan sisi (5,6) sudah ada di pohon dan sisi (4,6) mengakibatkan terjadinya sirkuit dengan lintasan 4,5,6,4 sehingga titik (4,6) ditolak.



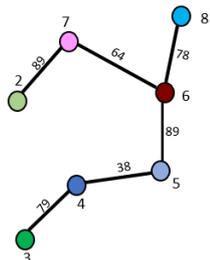
Gambar 30 Sisi (6,7) Diterima
 Sumber: dokumentasi pribadi



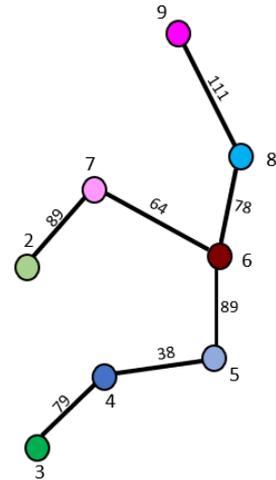
Gambar 31 Sisi (6,8) Diterima
 Sumber: dokumentasi pribadi



Gambar 32 Sisi (3,4) Diterima
 Sumber: dokumentasi pribadi

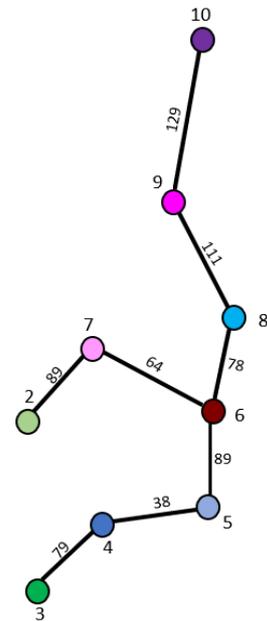


Gambar 33 Sisi (2,7) Diterima
 Sumber: dokumentasi pribadi



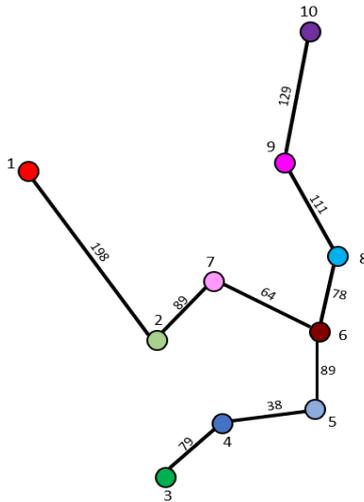
Gambar 34 Sisi (8,9) Diterima
 Sumber: dokumentasi pribadi

Berikutnya kita tinjau sisi-sisi di tabel 3. Pada tabel 3, sisi (3,5), (2,4) ditolak karena kedua sisi tersebut mengakibatkan terbentuknya sirkuit di dalam pohon. Sisi (9,10) diterima karena adanya sisi tersebut tidak mengakibatkan terbentuknya sirkuit di dalam pohon. Sedangkan sisi (7,8), (8,10), (2,6), (2,5) kita tolak karena sisi-sisi tersebut menyebabkan sirkuit di pohon.



Gambar 35 Sisi (9,10) Diterima
 Sumber: dokumentasi pribadi

Setelah itu, kita tinjau sisi-sisi pada tabel 4. Sisi (2,3) ditolak karena mengakibatkan sirkuit pada pohon. Akan tetapi, sisi (1,2) diterima oleh pohon karena tidak mengakibatkan sirkuit sehingga semua simpul pada pohon sudah terhubung yang menandakan berakhirnya proses Algoritma Kruskal



Gambar 36 Sisi (1,2) Diterima
Sumber: dokumentasi pribadi

Gambar 36 merupakan bentuk akhir dari pohon merentang minimum menggunakan Algoritma Kruskal. Adapun jumlah langkah yang dibutuhkan untuk membuat pohon merentang minimum ini sebanyak 17 langkah dengan total bobot dari pohon merentang minimum sebesar 875 km.

V. KESIMPULAN

Pembangunan infrastruktur di Indonesia merupakan hal yang sangat penting. Oleh karena itu ketertinggalan Kalimantan Timur, calon ibu kota baru, dalam bidang pembangunan menjadi prihatin bagi pemerintah. Salah satu infrastruktur yang perlu dibangun Kalimantan Timur tidak lain adalah jalan tol. Meskipun sangat penting, pembangunan jalan tol memerlukan biaya yang sangat besar. Oleh karena itu, untuk menghemat biaya rancangan jalan tol harus efisien atau yang jaraknya paling pendek namun dapat menghubungkan semua daerah di Kalimantan Timur.

Aplikasi pohon merentang dapat digunakan untuk menyelesaikan permasalahan ini dengan dua jenis pendekatan yaitu pendekatan menggunakan Algoritma Prim dan pendekatan menggunakan Algoritma Kruskal. Kedua cara menghasilkan hasil yang sama dengan cara yang berbeda. Algoritma Prim memerlukan 9 langkah dalam membuat pohon merentang minimum sedangkan Algoritma Kruskal membutuhkan 17 langkah. Hasil dari kedua algoritma berupa sebuah rancangan jalan tol yang menghubungkan semua daerah di Kalimantan Timur dengan total jarak sebesar 875 km. Sehingga dapat ditarik kesimpulan bahwa penggunaan pohon merentang minimum melalui Algoritma Prim ataupun Algoritma Kruskal dalam pembuatan jalan tol di Kalimantan Timur sukses dan efisien.

VI. UCAPAN TERIMA KASIH

Pertama-tama penulis menaikan pujian dan syukur kepada Tuhan Yang Maha Esa atas berkat rahmat-Nya, penulis dapat menyelesaikan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada dosen-dosen dari mata kuliah IF2120 tahun 2020 yang telah mengajar dan menyediakan bahan ajaran kepada penulis sehingga penulis dapat memahami materi yang ada dan dapat membuat makalah ini. Yang terakhir penulis juga

mengucapkan terima kasih kepada orang tua penulis dan semua orang yang telah mendukung penulis dalam pembuatan makalah ini.

DAFTAR PUSTAKA

- [1] <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>, diakses pada 7 Desember 2020.
- [2] Daftar Kabupaten, <https://kaltimprov.go.id/daftar-kabupaten>, diakses pada 8 Desember 2020.
- [3] Prim's Minimum Spanning Tree, <https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/>, diakses pada 11 Desember 2020.
- [4] Kruskal's Minimum Spanning Tree Algorithm, <https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/>, diakses pada 11 Desember 2020.
- [5] <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>, diakses pada 11 Desember 2020.

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, 3 Desember 2020

Jordan Daniel Joshua / 13519098