

**PENERAPAN ALGORITMA SOLLIN PADA JARINGAN KABEL
TELKOM TANJUNG SELOR BERBANTU MATLAB**

SKRIPSI



Oleh :

**AHYAREFENDY
NPM. 201521008**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS KALTARA
TANJUNG SELOR
2021**

**PENERAPAN ALGORITMA SOLLIN PADA JARINGAN KABEL
TELKOM TANJUNG SELOR BERBANTU MATLAB**

SKRIPSI

Oleh

**AHYAR EFENDY
NPM. 201521008**



**UNIVERSITAS KALTARA TANJUNG SELOR
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
JURUSAN MATEMATIKA**

2021

**PENERAPAN ALGORITMA SOLLIN PADA JARINGAN KABEL
TELKOM TANJUNG SELOR BERBANTU MATLAB**

SKRIPSI

Oleh

**AHYAR EFENDY
NPM. 201521008**

*Diajukan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana (S1)
Matematika pada Jurusan Matematika Fakultas Matematika dan Ilmu
Pengetahuan Alam Universitas Kaltara Tanjung Selor*

**UNIVERSITAS KALTARA TANJUNG SELOR
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
JURUSAN MATEMATIKA**

2021

PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan dibawah ini:

Nama : Ahyar Efendy
NPM : 201521008
Jurusan : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai hasil tulisan atau pemikiran saya sendiri. Apabila kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Tanjung Selor, Juli 2021

Yang membuat pernyataan,

Ahyar Efendy
NPM.201521008

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS KALTARA
TANJUNG SELOR

Kami dengan ini menyatakan bahwa skripsi yang ditulis oleh :

AHYAR EFENDY

NPM.201521008

Dengan Judul Penelitian:

**“PENERAPAN ALGORITMA *SOLLIN* PADA JARINGAN KABEL
TELKOM TANJUNG SELOR BERBANTU MATLAB”**

Diterima sebagai salah satu syarat untuk memperoleh gelar sarjana

Menyetujui:

Pembimbing

St. Syahdan, M.Pd
NIDN. 11.080989.01

Dekan Fakultas MIPA,

Ketua Jurusan Matematika

Ratna Dwi Christyanti, M.Si
NIDN. 11.110388.02

St. Syahdan, M.Pd
NIDN. 11.080989.01

PENGESAHAN SKRIPSI

Skripsi ini telah dipertahankan di hadapan sidang Panitia Ujian Skripsi Fakultas
MIPA Universitas Kaltara pada tanggal 22 Juli 2021

Dewan Penguji

Penguji I,

Penguji II,

Dr, Siti Aisyah, M.Si
NIDN. 11.130684.01

Ratna Dwi Christyanti, M.Si
NIDN. 11.110388.02

Mengesahkan,
Dekan Fakultas MIPA,

Mengetahui,
Ketua Jurusan Matematika

Ratna Dwi Christyanti, M.Si
NIDN. 11.110388.02

St. Syahdan, M.Pd
NIDN. 11.080989.01

ABSTRAK

AHYAR EFNDY. *Penerapan Algoritma Sollin Pada Jaringan Kabel Telkom Tanjung Selor Berbantu MATLAB*, (dibimbing oleh St. Syahdan)

Rumusan masalah dalam penelitian ini adalah bagaimana penerapan algoritma *sollin* pada jaringan kabel Telkom Tanjung Selor berbantu matlab. Adapun tujuan yang diharapkan dalam penelitian ini adalah untuk mengetahui Penerapan algoritma *sollin* pada jaringan kabel Telkom Tanjung Selor berbantu matlab.

Teori graf merupakan salah satu cabang matematika sebenarnya sudah ada sejak lebih dari dua ratus tahun yang silam. Jurnal pertama tentang teori graf muncul pada tahun 1736, oleh matematikawan terkenal dari Swiss yang bernama Euler. Puluhan tahun terakhir ini teori graf mengalami perkembangan yang sangat pesat. Dalam kehidupan sehari-hari terdapat permasalahan mengenai optimasi yang dapat diselesaikan menggunakan pohon rentang minimum, atau dikenal dengan istilah *Minimum Spanning Tree*.

Pohon merentang minimum (*minimum spanning tree*) merupakan metode pencarian jalur optimum kesemua titik graf, dari titik awal hingga akhir dengan titik membentuk sirkuit. Saat ini, pencarian jalur optimum merupakan permasalahan yang semakin penting di kehidupan sehari-hari karena dianggap efisien, efektif dan menguntungkan untuk mencapai tujuan yang diinginkan. Terdapat beberapa algoritma untuk menyelesaikan masalah pohon merentang minimum. Pada penelitian ini akan menggunakan Algoritma *Sollin*.

Algoritma *Sollin* merupakan suatu metode yang digunakan untuk mencari dan membentuk pohon merentang minimum yang mana tahap demi tahap begitu juga dengan tujuan dari Algoritma *Sollin* adalah untuk mendapatkan nilai yang minimum dari suatu jaringan kabel di PT. Telkom Tanjung Selor.

Dari data yang diperoleh dibuat dalam bentuk tabel, kemudian dari tabel tersebut akan dibuat gambar jalur dengan titik sumber PT. Telkom sampai ke semua ujung titik. Dari gambar jaringan kabel Telkom tersebut diketahui adanya 17 titik dan 16 sisi yang menghubungkan setiap titik. Berdasarkan dari gambar jaringan kabel Telkom tersebut akan dicari pohon rentang minimum menggunakan Algoritma *Sollin*.

Hasil dari perhitungan menggunakan Algoritma *Sollin* berbantu Maple menghasilkan nilai minimum yang sama dengan bobot 5.731 m dengan kata lain pengujian ini telah menghemat kabel sepanjang 2.952 m dari total jarak yaitu 8.683 m.

Kata kunci: *Graf, Pohon Rentang Minimum, Algoritma Sollin*

MOTTO

“Masa depan kita, ada ditangan kita sendiri. Jika masa muda, Anda isi hanya dengan malas – malasan tidak menutup kemungkinan masa tua Anda juga akan terisi kemalasan saja. Sebaliknya, jika Anda memulainya sekarang tanpa peduli rasa malas menerpa, esok saat tua Anda akan menang”

-B.J Habibie

“Sesungguhnya ada ilmu yang akan membuatmu menjadi seseorang yang pandai berhitung jika dikaji sampai puncaknya.

Dan ilmu itu adalah ilmu faraid”

“Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap”.

(QS. Al-Insyirah,6-8).

Berusaha dan berdo'a adalah kunci suatu kesuksesan

KATA PENGANTAR

Puji dan syukur selalu dipanjatkan Kehadirat Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai tugas akhir dengan baik yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Sains.

Penulis menyadari bahwa selama proses penyusunan ini tidak terlepas dari bimbingan, arahan, bantuan serta banyak pihak yang telah membantu karena itu penulis menyampaikan terima kasih kepada :

1. Ibu Ratna Dwi Christiyanti, M.Si selaku Dekan Fakultas MIPA Universitas Kaltara.
2. Ibu St. Syahdan, M.Pd selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Kaltara dan selaku Dosen Pembimbing saya atas bimbingan dan saran yang telah diberikan kepada penulis.
3. Semua Dosen Jurusan Matematika Fakultas MIPA Universitas Kaltara yang telah memberikan ilmunya kepada penulis.
4. Keluarga besar Universitas Kaltara, khususnya teman seperjuangan di Fakultas MIPA Jurusan Matematika terima kasih atas dukungan, semangat dan kerjasamanya.
5. Kedua orang tua serta saudara-saudara yang telah memberikan dorongan moral dan doa selama ini.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, maka dari itu penulis mengharapkan kritik dan saran demi kesempurnaan penulisan ini sehingga dapat memberikan manfaat bagi pembaca dan dapat diterapkan dilapangan serta dikembangkan lebih lanjut.

Tanjung Selor, Juli 2021

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
LEMBAR PERSETUJUAN	iv
LEMBAR PENGESAHAN	v
ABSTRAK	vi
MOTTO	vii
PERSEMBAHAN.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penulisan	3
BAB II KAJIAN PUSTAKA	
2.1 Definisi Graf.....	4
2.2 Komponen Graf.....	5
2.3 Jenis-Jenis Graf	5
2.4 Pohon.....	8
2.5 Pohon Rentang	9
2.6 Pohon Rentang Minimum	10
2.7 Definisi Algoritma.....	12
2.8 Algoritma Sollin.....	13
2.9 Runing Time.....	16
2.10 Kompleksitas.....	16
2.11 Matlab.....	17

BAB III METODE PENELITIAN

3.1 Jenis Penelitian	18
3.2 Lokasi Penelitian	18
3.3 Prosedur Penelitian.....	18
3.4 Flowchart Algoritma Sollin	19

BAB IV PEMBAHASAN

4.1 Hasil Penelitian	22
4.2 Algoritma <i>Sollin</i> Dalam Menentukan Pohon Rentang Minimum	24

BAB V PENUTUP

5.1 Kesimpulan.....	36
5.2 Saran.....	36

DAFTAR PUSTAKA	37
-----------------------------	-----------

DAFTAR TABEL

	Halaman
Tabel 4.1 Keterangan Graf.....	24
Tabel 4.2 Kerterangan Graf.....	25

DAFTAR GAMBAR

	Halaman
Gambar 4.1 Graf STO, ODC, ODP	24
Gambar 4.2 Algoritma <i>Sollin</i>	30
Gambar 4.3 Algoritma <i>Sollin</i>	31
Gambar 4.4 Algoritma <i>Sollin</i>	32
Gambar 4.5 Algoritma <i>Sollin</i>	33
Gambar 4.6 Pohon Rentang Minimum Dengan Algoritma <i>Sollin</i>	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pohon merentang minimum (*minimum spanning tree*) merupakan metode pencarian jalur optimum kesemua titik graf, dari titik awal hingga akhir dengan titik membentuk sirkuit. Saat ini, pencarian jalur optimum merupakan permasalahan yang semakin penting di kehidupan sehari-hari karena dianggap efisien, efektif dan menguntungkan untuk mencapai tujuan yang diinginkan. Terdapat beberapa algoritma untuk menyelesaikan masalah pohon merentang minimum, diantaranya yaitu Algoritma Sollin. Dapat digunakan untuk mencari total jarak atau panjang minimum.

Algoritma sollin adalah algoritma yang digunakan untuk pencarian pohon merentang minimum (*minimum spanning tree*) sebuah graf berbobot dengan berorientasi pada bobot maksimum graf. Cara kerja algoritma solin adalah dengan memanfaatkan cutset optimal. Kita dapat mengatakan bahwa algoritma solin adalah perpaduan antara algoritma prim dan kruskal. Pada setiap iterasi algoritma solin akan menghasilkan jarak minimum antara dua buah titik yang menggunakan konsep nearest-neighbbor. Jadi pada setiap node kita mencari titik manakah dari titik tersebut yang menghasilkan jarak terendah. Setelah setiap titik memiliki pasangannya akan diperiksa apakah yang diperoleh connected, jika tidak pilih cutset yang memiliki bobot terendah. Berikut adalah algoritma solin [ahuja dkk, 1993]

Munculnya jaringan kerja pada sejumlah perencanaan diberbagai macam bidang baik itu dalam jaringan komunikasi, jaringan transportasi, jaringan listrik dan sebagainya merupakan sesuatu yang dapat ditemui dalam kehidupan sehari-hari. Masalah pengiriman barang yang berkaitan dengan persoalan transportasi atau distribusi dari suatu sumber ke suatu tujuan dengan biaya transportasi minimum, maka model transportasi ini dapat pula diselesaikan dengan suatu jaringan.

Pada penelitian ini penulis akan menggunakan graf tak berarah berbobot. Graf tersebut tersusun dari banyak titik dan garis, titik merepresntasikan rumah pelanggan, dan sisi adalah panjang pipa yang menghubungkan satu rumah kerumah yang lain. Teori graf merupakan suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Graf digunakan untuk menggambarkan berbagai macam struktur dan sebagai visualisasi objek-objek agar lebih mudah dimengerti. Beberapa contoh graf yang sering dijumpai, antara lain struktur organisasi, bagan alir, pengambilan mata kuliah, peta, rangkaian listrik, dan lain-lain. (Rusli, 2018)

Dari sekian banyak konsep yang ada didalam teori graf, konsep pohonlah yang akan digunakan karena sangat mudah dan luas jika diterapkan diberbagai bidang ilmu. Dalam kehidupan sehari-hari banyak masalah yang ditemui, sehingga dapat diselesaikan dengan menggunakan pohon rentang minimum atau dikenal dengan (*minimum spanning tree*). Seperti menentukan biaya minimum (termurah), jarak terpendek, maupun jaringan listrik dan sebagainya.

Penelitian ini menggunakan algoritma untuk menentukan pohon rentang minimum (*minimum spanning tree*) pada teori graf. Ada beberapa macam algoritma yang bisa digunakan untuk menyelesaikan pohon rentang minimum (*minimum spanning tree*), antara lain algoritma *Prim*, algoritma *Kruskal*, algoritma *Sollin*, algoritma genetika dan lain-lain. Algoritma itu tergolong dalam algoritma *Greedy*, kita mencari nilai minimum semua dengan mencari nilai minimum perbagian, dengan harapan merupakan nilai minimum total.

Dari berbagai metode di atas, Algoritma Sollin merupakan salah satu metode yang digunakan oleh beberapa peneliti dalam menyelesaikan pohon rentang minimum seperti penelitian sebelumnya yang dilakukan oleh.....

Berdasarkan uraian tersebut, penulis mengambil judul “**Penerapan Algoritma Sollin pada Jaringan Kabel Telkom Tanjung Selor Berbantu Matlab**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang dapat diangkat dalam penelitian ini adalah “Bagaimana penerapan Algoritma *Sollin* pada Jaringan Kabel Telkom Tanjung Selor Berbantu Matlab”?

1.3 Tujuan Penelitian

Adapun tujuan yang diharapkan dalam penelitian ini adalah untuk mengetahui Penerapan Algoritma *sollin* pada Jaringan Kabel Telkom Tanjung Selor Berbantu Matlab

1.4 Manfaat Penelitian

Manfaat bagi peneliti adalah peneliti dapat menerapkan ilmu yang sudah dipelajari tentang pohon rentang minimum sehingga dapat diterapkan pula dalam kehidupan nyata. Adapun manfaat bagi pembaca ialah terkhusus buat yang memiliki usaha dibidang ini agar dapat menjadikannya sebagai bahan pertimbangan

BAB II

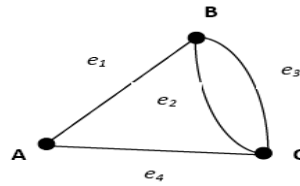
KAJIAN PUSTAKA

2.1 Definisi Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E) , yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertex* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul. (Munir, 2010)

Graf ditulis dengan notasi $G = (V, E)$ yang mana V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi yang menghubungkan sepasang simpul. Simpul pada graf dapat dinomori dengan huruf a, b, c dan seterusnya dengan bilangan asli $1, 2, 3$ dan seterusnya atau gabungan keduanya. Secara geometri graf dapat digambarkan sebagai sekumpulan simpul di dalam bidang dwimatra yang dihubungkan dengan sekumpulan garis (sisi).

Graf merupakan salah satu cabang dari ilmu dalam komputer yang memiliki peruntukan yang cukup luas dalam kehidupan nyata, diantaranya adalah penggunaan graf dalam melakukan perutean. Perutean adalah suatu kegiatan pembuatan jalur atau rute dengan tujuan tertentu. Penggunaan graf akan didapatkan lintasan yang memiliki keunggulan-keunggulan tertentu misalnya lintasan dengan biaya paling murah, lintasan dengan waktu tempuh paling cepat, lintasan dengan jarak paling pendek, dan lintasan dengan tingkat efisiensi paling tinggi. Pada bidang yang lebih luas, apabila lintasan yang dibuat dianalogikan dengan alur kerja yang harus dilakukan, maka akan didapatkan hasil yang optimal.



Gambar 2.1 Graf G

2.2 Komponen Graf

Menurut Prasetyo didalam Ikhsan, (2013) ada beberapa terminologi dari teori graf yang digunakan untuk menjelaskan komponen-komponen penyusunnya, ketika melihat suatu graf.

1) Titik (*verteks*)

Titik (*verteks*) adalah himpunan titik yang tidak kosong dan terbatas yang disimbolkan dengan v , dimana jumlah titik pada graf dapat dinyatakan dengan $n = |v|$

2) Sisi (*edge*)

Sisi (*edge*) adalah himpunan yang sisinya menghubungkan sepasang titik yang disimbolkan dengan e .

3) Derajat (*degree*)

Derajat (*degree*) adalah jumlah sisi yang berada pada titik tertentu yang mana disimbolkan dengan $d(v)$.

4) Ukuran (*size*)

Ukuran (*size*) adalah banyaknya titik yang dimiliki dari suatu graf.

2.3 Jenis-Jenis Graf

Beberapa jenis graf dapat dikelompokkan tergantung dari sudut pandang pengelompokannya, dimana dari pengelompokan tersebut

berdasarkan ada atau tidaknya sisi ganda, berdasarkan jumlah simpul dan berdasarkan orientasi arah pada sisi. Berdasarkan ada atau tidaknya sisi ganda atau *Loop* pada suatu graf, maka secara umum ada 2 jenis graf yang dapat digolongkan, yaitu :

1. Graf sederhana (*simple graph*)

Graf sederhana adalah graf yang tidak mengandung sisi ganda maupun *Loop*

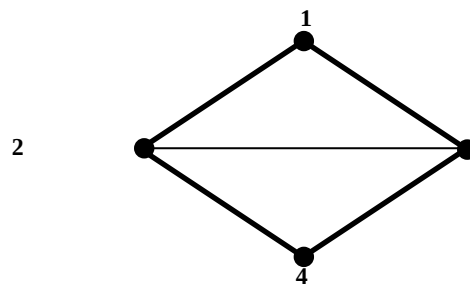
2. Graf tak sederhana (*unsimple graph*)

Graf tak sederhana adalah graf yang mengandung sisi ganda atau *Loop*.

Berdasarkan sisi pada graf yang memiliki orientasi arah, maka secara umum graf dapat dibedakan menjadi 2 jenis, yaitu :

- a. Graf tak-berarah (*undirected graph*)

Graf tak-berarah adalah graf yang sisinya tidak memiliki orientasi arah dan urutan dari pasangan simpul yang dihubungkan oleh sisi tersebut tidak diperhatikan. Berarti $(u, v) = (v, u)$ adalah sisi yang sama.

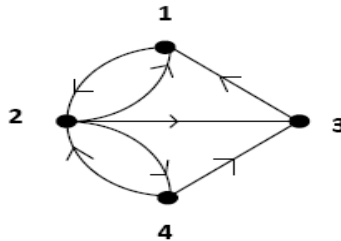


Gambar 2.2 Graf tak berarah

- b. Graf berarah (*directed graph* atau *digraph*)

Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. Pada graf berarah (u, v) dan (v, u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u, v) \neq (v, u)$. Untuk busur (u, v) , simpul (*initial vertex*) asal

diberi nama simpul u dan simpul terminal (*terminal vertex*) dinamakan simpul v .



Gambar 2.3 Graf berarah

Menurut Munir didalam Umi Latifah, (2012) adanya beberapa sifat yang berkaitan dengan graf yang sering digunakan, yaitu:

1) Bertetangga (*adjacency*)

Dua buah simpul pada graf tak-berarah G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain, u bertetangga dengan v jika (u, v) adalah sebuah sisi pada graf G .

2) Bersisian (*incident*)

Untuk sembarang sisi $e = (u, v)$, sisi e dikatakan bersisian dengan simpul u dan simpul v .

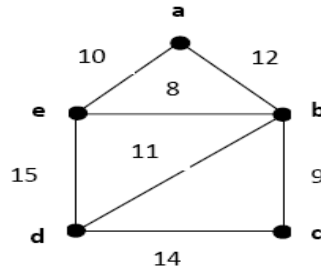
3) Graf Terhubung (*connected graph*)

Di dalam graf keterhubungan antar dua buah simpul itu sangat penting. Dua buah simpul u dan simpul v dikatakan terhubung jika terdapat lintasan dari u ke v . Jika simpul yang pertama dapat dicapai dari simpul kedua maka pasti dua buah simpul tersebut terhubung. Apabila dua simpul terminal pada jaringan komputer terhubung maka dapat berkomunikasi.

Graf tak-berarah G disebut graf terhubung (*connected graph*) jika untuk setiap pasangan simpul u dan v di dalam himpunan V terdapat lintasan dari u ke v (yang juga harus berarti ada lintasan dari v ke u). Jika tidak, maka graf G disebut graf tak terhubung (*disconnected graph*).

4) Graf Berbobot (*weighted graph*)

Graf yang setiap sisinya diberi harga (bobot) disebut graf berbobot. Sisi yang memiliki bobot berbeda-beda tergantung dari masalah yang dimodelkan pada graf. Bobot dapat dinyatakan dengan jarak antara dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain (dalam jaringan komputer), biaya produksi, dan lain sebagainya. Contoh gambar graf berbobot:



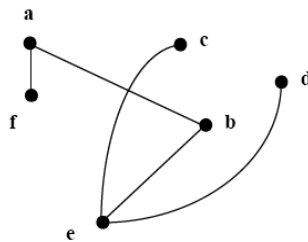
Gambar 2.4 Graf berbobot

2.4 Pohon

Pohon (*tree*) merupakan salah satu bentuk khusus dari struktur suatu graf. Pohon adalah suatu graf terhubung yang tidak mempunyai subgraf yang memuat sirkuit. Sedangkan sirkuit dalam graf adalah lintasan dalam graf dimana simpul awal dan simpul akhirnya sama. (Angga, 2013)

Menurut Ayu didalam Riska, (2012) pohon (*tree*) didefinisikan sebagai graf terhubung yang tidak mengandung sirkuit. Karena merupakan graf terhubung, maka pohon selalu terdapat jalur (*path*) yang menghubungkan setiap dua simpul dalam pohon.

Pohon memiliki sifat yang penting yaitu terhubung dan tidak mengandung sirkuit. Pohon dinotasikan sama dengan, $T = (V, E)$ Keterangan: T : *Tree* V : *Vertices* atau *node* atau *vertex* atau simpul, V merupakan himpunan tidak kosong. $V = \{v_1, v_2, \dots, v_n\}$ E : *Edges* atau area atau sisi yang menghubungkan simpul $E = \{e_1, e_2, \dots, e_n\}$.



Gambar 2.6 Gambar pohon G_1

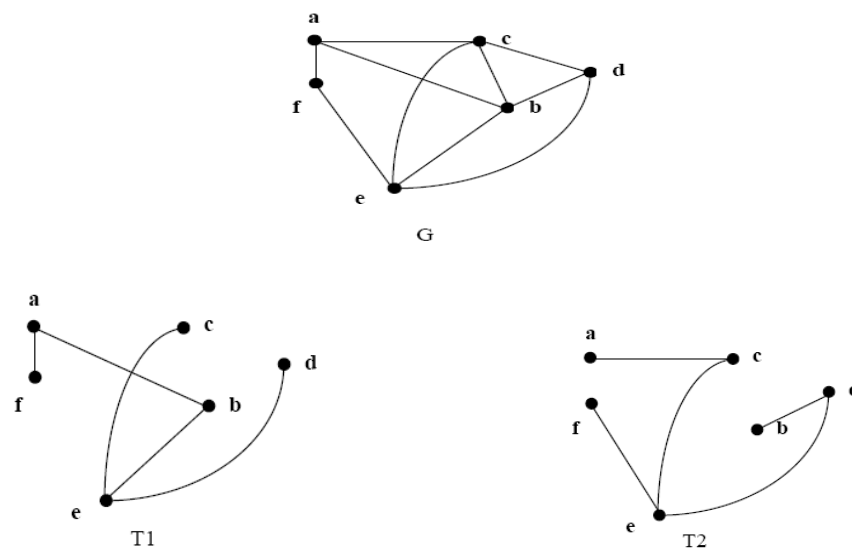
2.5 Pohon Rentang

Menurut Munir didalam Marwan, (2012) misalkan $G = (V, E)$ adalah graf tak berarah terhubung yang bukan pohon, yang berarti di G terdapat satu atau lebih sirkuit. G dapat diubah menjadi pohon $T = (V_1, E_1)$ dengan cara memutuskan sirkuit-sirkuit yang ada.

Disebut pohon merentang karena semua simpul pada pohon T sama dengan semua simpul pada graf G , dan sisi-sisi pada pohon $T \subseteq$ sisi-sisi pada graf G . Dengan kata lain $V_1 = V$ dan $E_1 \subseteq E$. Apabila proses ini dilakukan

secara berulang-ulang sampai semua sirkuit di G hilang, maka G akan menjadi sebuah pohon T , yang dinamakan pohon merentang (*spanning tree*).

Dari setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang (*spanning tree*). Suatu pohon T yang berasal dari sebuah graf G disebut pohon rentang. Graf G adalah graf tak berarah terhubung yang bukan merupakan pohon (memiliki siklus), sedangkan pohon T adalah graf yang merupakan graf pohon yang didapat dengan cara memutuskan sirkuit-sirkuit yang terdapat pada graf G . Maka, graf pohon yang semua himpunan simpulnya merupakan *Improper Subset* dari himpunan simpul yang terdapat pada graf G disebut pohon rentang, sedangkan *Proper Subset* dari himpunan semua sisi pada graf G disebut himpunan semua sisi.



Gambar 2.7 G adalah Graf, T_1 dan T_2 adalah pohon rentang dari Graf G

2.6 Pohon Rentang Minimum

Menurut Munir didalam Deny, (2009) apabila G adalah graf berbobot, maka bobot pohon merentang T dari G didefinisikan sebagai jumlah bobot semua sisi di

T . Diantara semua pohon merentang dalam graf G , pohon merentang yang berbobot minimum dinamakan pohon merentang minimum.

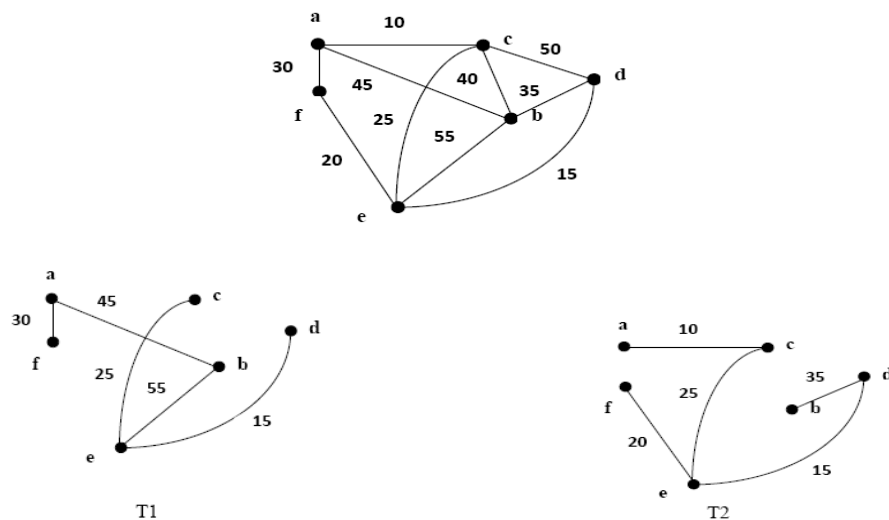
Pohon merentang yang berbeda mempunyai bobot yang berbeda pula. Pohon merentang minimum ini mempunyai terapan yang luas dalam masalah riil. Dikatakan pohon rentang minimum apabila diantara semua pohon merentang dalam graf G merupakan pohon yang memiliki bobot minimum.

Deny, (2011) mengatakan ada beberapa langkah-langkah untuk menghitung total jarak minimum dari suatu graf sebagai berikut :

1. Dari suatu graf yang terbentuk, perhatikan apakah memenuhi kriteria suatu pohon merentang.
2. Lakukan pelacakan secara berurutan mulai dari simpul pertama sampai simpul terakhir.
3. Pada setiap simpulnya perhatikan nilai (bobot) tiap-tiap sisinya.
4. Ambil nilai yang paling kecil artinya jarak terpendek dari setiap sisi simpul.
5. Lanjutkan sampai seluruh simpul tergambar pada pohon merentang.
6. Jumlahkan nilai yang telah dipilih atau jarak minimum yang menghubungkan simpul-simpul tersebut.

Jika graf berbobot adalah G , maka bobot dari pohon rentang T dari G didefinisikan sebagai jumlah bobot pada semua sisi di T . Pohon rentang yang berbeda memiliki bobot yang berbeda pula. Dinamakan pohon rentang minimum apabila diantara semua pohon rentang di G memiliki bobot paling minimum. Menyangkut pemilihan seperangkat penghubung yang merupakan persoalan dari

pohon rentang minimum yang menghubungkan semua *node* dalam suatu jaringan, sedemikian rupa sehingga jumlah panjang dari semua penghubung terpilih tadi adalah sekecil mungkin.



Gambar 2.8 G adalah Graf Berbobot T_1 dan T_2 adalah pohon rentang Berbobot

2.7 Definisi Algoritma

Algoritma adalah susunan yang logis dan sistematis dalam memecahkan suatu masalah untuk mencapai tujuan tertentu. Dalam menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka, algoritma berperan penting dalam penyusunan program atau syntax tersebut. (Gun, 2017)

Algoritma merupakan salah satu cabang ilmu komputer yang membahas tentang prosedur penyelesaian dari suatu permasalahan. Komputer bisa

menyelesaikan perhitungan dengan cepat dan benar apabila dengan menggunakan algoritma yang baik. Sebaliknya jika algoritmanya kurang baik maka penyelesaiannya akan lambat dan bahkan tidak didapatkan solusi yang diinginkan.

Suatu algoritma akan memerlukan masukan (*input*) tertentu untuk memulainya, dan akan menghasilkan keluaran (*output*) tertentu pada akhirnya. Dalam algoritma adapun yang perlu diperhatikan dalam mencari langkah-langkah yang paling sesuai untuk penyelesaian suatu masalah, karena setiap algoritma memiliki karakteristik tertentu yang memiliki kelebihan dan kekurangan.

Beberapa hal yang harus dipahami dalam mencari algoritma antara lain:

1. Masalah seperti apa yang hendak diselesaikan.
2. Gagasan apa yang ada pada algoritma tersebut.
3. Berapa lama yang diperlukan untuk menyelesaikan masalah.
4. Berapa jumlah data yang dapat ditangani oleh suatu algoritma.

2.8 Algoritma Sollin

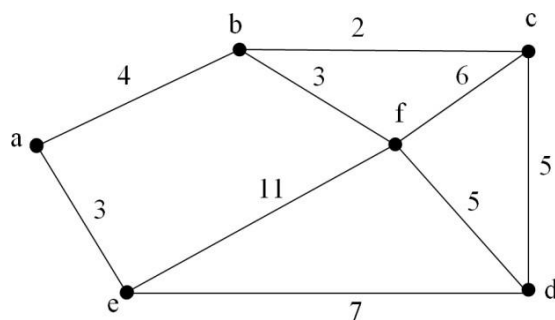
Algoritma Sollin adalah pencarian pohon merentang minimum (*minimum spannin tree*) berdasarkan pada sisi atau busur dengan bobot terbesar yang dimiliki suatu graf, dari sisi-sisi graf yang ada, akan dipilih sisi yang memiliki bobot terbesar, lalu diperiksa apabila sisi tersebut dihapus akan memutus graf atau tidak, jika tidak, maka sisi dihapus jika ya, maka pilih sisi bobot terbesar lainnya dengan tidak menyebabkan graf menjadi tidak terhubung atau terputus dan membentuk sirkuit. Penghapusan dapat berhenti

apabila tidak ada lagi sisi yang dapat dihapus (Khoiroh, 2010). Secara terurut algoritma Sollin dapat dituliskan sebagai berikut :

1. Hapus sisi dengan bobot paling besar, dengan tidak memutus graf.
2. Lakukan langkah pertama sampai tidak terdapat lagi sisi bobot terbesar yang bisa dihapus. Dalam artian jika sisi terhapus akan menyebabkan graf menjadi tidak terhubung.

Contoh kasus :

Pada gambar 2.13 merupakan contoh soal penerapan dari algoritma Sollin, sebagai berikut:

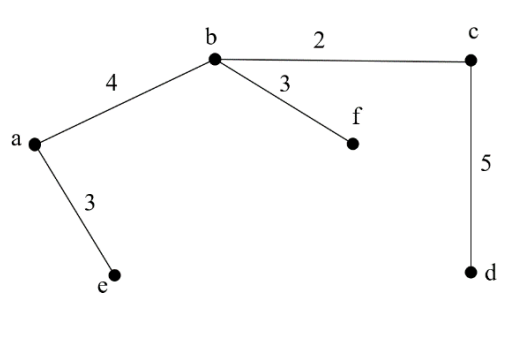


Gambar 2.13 Soal Penerapan Algoritma Sollin

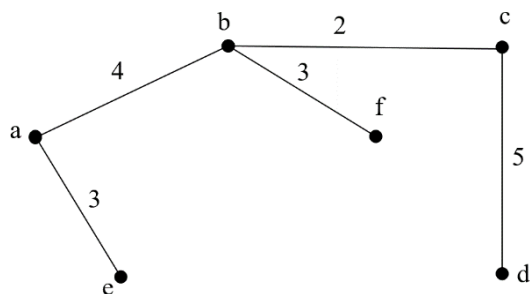
Dengan menggunakan algoritma Sollin, langkah pertama yang diambil adalah menghapus sisi dengan bobot paling besar, lalu lakukan pengecekan apakah dengan menghapus sisi tersebut akan memutuskan graf, jika tidak, hapus sisi tersebut. Langkah langkah penyelesaian algoritma Sollin dapat dilihat pada Tabel 2.2.

Tabel 2.2 Algoritma Sollin

Iterasi	Langkah	Graf (T)
1	Hapus ef dengan bobot 11	
2	Hapus ed dengan bobot 7	
3	Hapus cf dengan bobot 6	

4	Hapus df dengan bobot 5	
---	-------------------------	--

Karena sudah diperoleh pohon merentang minimum yang sesuai dengan syarat dan ketentuan algoritma Sollin, maka langkah diberhentikan. Hasil pohon merentang minimum yang didapatkan dari perhitungan manual menggunakan algoritma Sollin pada gambar 2.14 dibawah ini :



Gambar 2.14 Pohon Merentang Minimum Algoritma Sollin

Jadi, diperoleh pohon merentang minimum yang diperoleh adalah $V(T) = \{a,b,c,d,e,f\}$ dan $E(T)=\{ab,ae,bc,bf,cd\}$ dengan total bobot 17.

2.9 Running Time

Running time adalah waktu yang dibutuhkan oleh sebuah algoritma untuk mengeksekusi suatu masalah pada sebuah komputer yang dimulai dari saat *user* menjalankan algoritma hingga saat algoritma berhenti (Siahaan, 2017).

2.10 Kompleksitas

Kompleksitas merupakan ukuran besaran yang digunakan untuk model abstrak pengukuran waktu dan ruang sebuah algoritma. Kompleksitas algoritma terdiri dari dua macam yaitu kompleksitas ruang yang ditulis dengan notasi $S(n)$, diukur dari jumlah memori yang terdapat di dalam algoritma dan Kompleksitas waktu yang ditulis dengan notasi $T(n)$, diukur dari jumlah proses yang dibutuhkan algoritma.

Dalam penelitian ini yang dibahas hanya kompleksitas waktu (Siahaan, 2017).

1 Kompleksitas Waktu

Kompleksitas waktu ditulis dengan notasi $T(n)$, ukuran jumlah langkah komputasi dihitung dari berapa kali processor melakukan komputasi dalam sebuah algoritma dengan input atau masukan berukuran tertentu, dimana input atau masukan merupakan jumlah data yang diproses oleh sebuah algoritma.

2 Kompleksitas Waktu Asimptotik

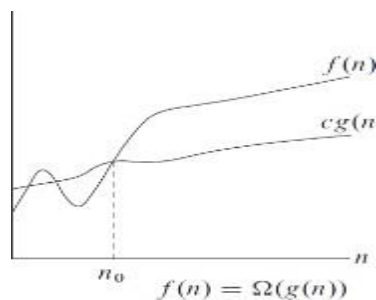
Kompleksitas waktu asimptotik digunakan untuk memperkirakan efisiensi algoritma berdasarkan $T_{min}(n)$ dan $T_{max}(n)$ bersamaan dengan besaran masukan. Kompleksitas waktu dibedakan atas tiga macam.

Kompleksitas waktu untuk kasus terbaik (*best case*) dinotasikan dengan $\Omega(g(n))$ (*Big-Omega*). Kompleksitas waktu untuk kasus rata-rata (*average case*) dilambangkan dengan notasi $\Theta(g(n))$ (*Big-Theta*). Dan kompleksitas waktu untuk kasus terburuk (*worst case*) dilambangkan dengan $O(g(n))$ (*Big-O*).

1. Notasi Ω (*Big-Omega*)

Fungsi $T(n)$ didefinisikan dalam $\Omega(g(n))$, dilambangkan $T(n) \in \Omega(g(n))$, jika $T(n)$ diberi batas bawah oleh beberapa kelipatan konstanta nilai positif dari $g(n)$ untuk semua n bernilai besar, yaitu jika ada beberapa konstanta positif c dan beberapa bilangan bulat bukan negatif n_0 (Levitin, 2009) sehingga $T(n) \geq cg(n)$ untuk semua $n \geq n_0$

Grafik fungsi *big-omega* dapat dilihat pada gambar 2.15



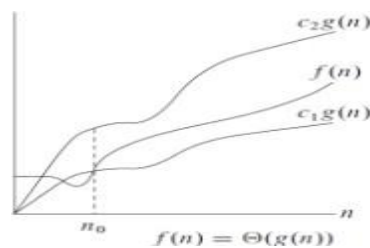
Gambar 2.15 Grafik Notasi *Big-Omega* (Cormen,

2. Notasi Θ (*Big-Theta*).

Fungsi $T(n)$ didefinisikan dalam $\Theta(g(n))$ disebut juga batas ketat, dilambangkan $T(n) \in \Theta(g(n))$, jika $T(n)$ diberi batas atas dan bawah oleh

beberapa kelipatan konstanta nilai positif dari $g(n)$ untuk semua n bernilai besar, yaitu jika ada beberapa konstanta positif c_1 dan c_2 dan beberapa bilangan bulat bukan negatif n_0 (Levitin, 2009) sehingga $c_2g(n) \leq f(n) \leq c_1g(n)$ untuk semua $n \geq n_0$

Grafik fungsi *big-theta* dapat dilihat pada gambar 2.16. .

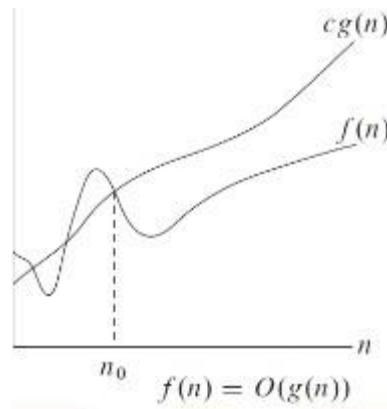


Gambar2.16 Grafik Notasi *Big-Theta* (Cormen, 2009)

3. Notasi O (Big-Oh).

Fungsi $T(n)$ didefinisikan dalam $O(g(n))$, dilambangkan $T(n) \in O(g(n))$, jika $T(n)$ diberi batas atas oleh beberapa kelipatan konstanta dari $g(n)$ untuk semua n bernilai besar, yaitu jika ada beberapa konstanta positif c dan beberapa bilangan bulat bukan negatif n_0 (Levitin, 2009) sehingga $T(n) \leq cg(n)$ untuk semua $n \geq n_0$

Grafik fungsi *big-oh* dapat dilihat pada gambar 2.17.



Gambar 2.17 Grafik Notasi *Big-Oh* (Cormen, 2009)

2.11 MATLAB (Matrix Laboratory)

Matlab merupakan sebuah singkatan dari Matrix Laboratory, yang pertama kali dikenalkan oleh University of New Mexico dan University of Stanford pada tahun 1970. software ini pertama kali memang digunakan untuk keperluan analisis numerik, aljabar linier dan teori tentang matriks. Saat ini, kemampuan dan fitur yang dimiliki oleh Matlab sudah jauh lebih lengkap dengan ditambahkannya toolbox-toolbox yang sangat luar biasa.

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic maupun C++. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan

dan grafik-grafik perhitungan. Matlab hadir dengan membawa warna yang berbeda. Hal ini karena matlab membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan visualisasi. Matlab dikembangkan oleh Math Works, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matrik pada proyek LINPACK dan EISPACK. Saat ini matlab memiliki ratusan fungsi yang dapat digunakan sebagai problem solver mulai dari simple sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu.

BAB III

METODE PENELITIAN

3.1 Jenis penelitian

Penelitian yang akan dilakukan adalah dengan mengumpulkan data-data yang ada di Kantor PT. Telkom yang mana data tersebut meliputi aspek fisik dengan tujuan untuk dapat memperoleh pemahaman yang lebih mendalam.

3.2 Sumber Data

sumber penelitian ini akan dilakukan di Kantor PT. Telkom tepatnya di wilayah Tanjung Selor Kabupaten Bulungan Provinsi Kalimantan Utara dan sumber data yang diperoleh adalah dari juarita.

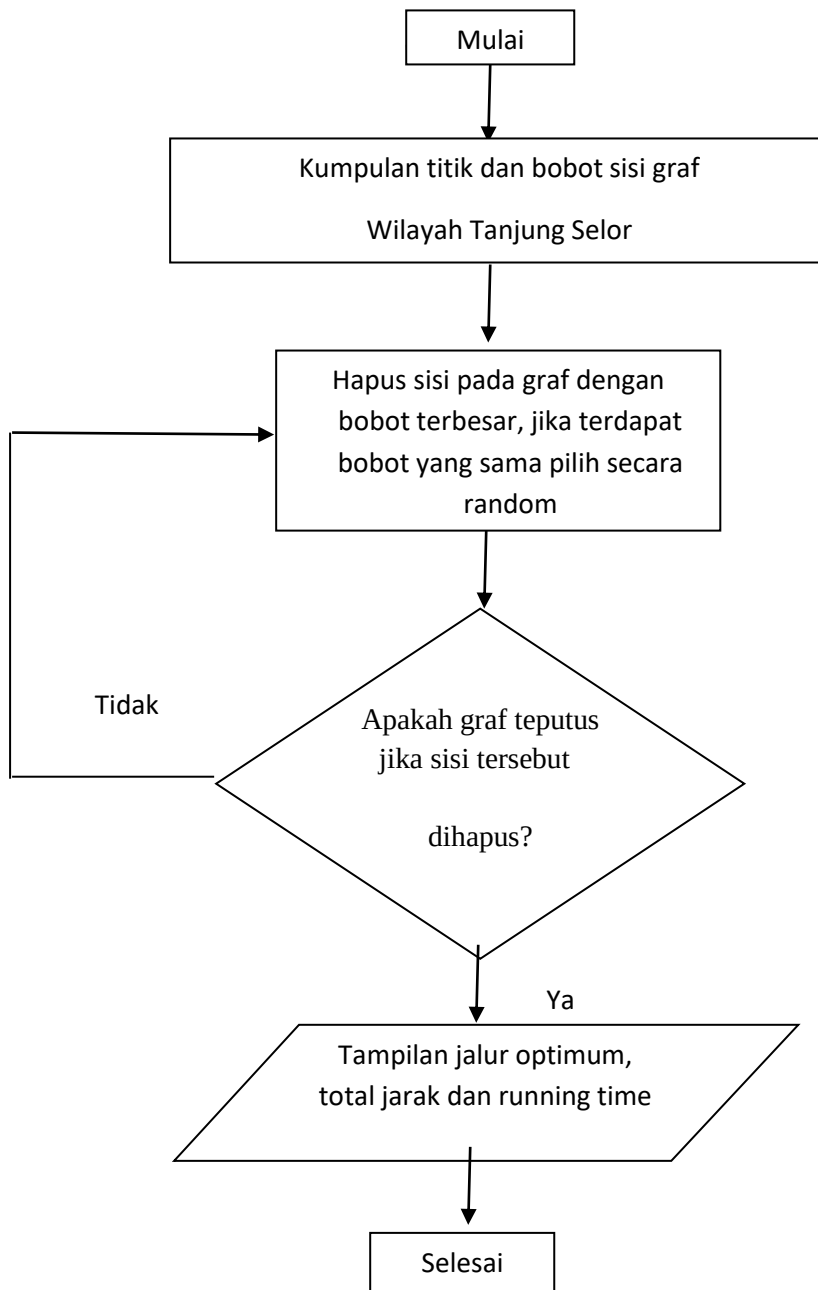
3.3 Prosedur penelitian

Pada prosedur penelitian akan dilakukan suatu kajian pustaka, yaitu dengan mengkaji permasalahan berdasarkan sumber pustaka yang ada. Adapun langkah-langkah yang akan dilakukan dalam tahapan pemecahan masalah ini antara lain:

1. Menjelaskan tentang apa itu Pohon Rentang Minimum (*minimum spanning tree*), Algoritma *Sollin*.
2. Menerapkan langkah-langkah Algoritma *Sollin* dalam penyelesaian pohon rentang minimum

Langkah ini merupakan langkah terakhir dari suatu penelitian, dimana akan dilakukan suatu penarikan kesimpulan berdasarkan penelitian dengan cara perhitungan antara Algoritma *Sollin* untuk menentukan pohon rentang minimum (*minimum spanning tree*).

3.4 Flowchart Algoritma Sollin



Gambar 3.8 Flowchart Algoritma Sollin

BAB IV

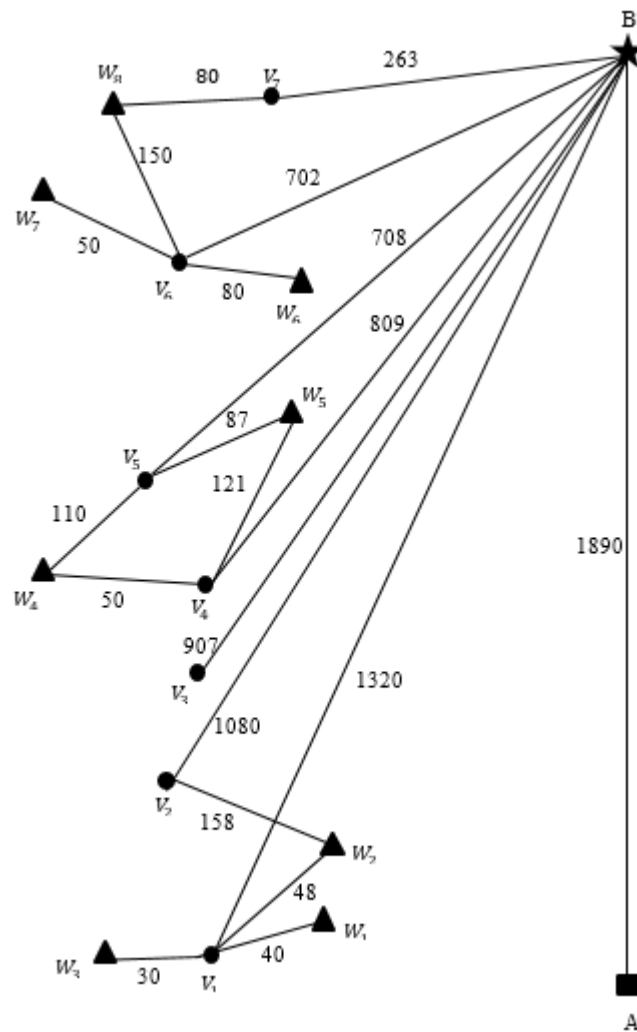
PEMBAHASAN

4.3 Hasil Penelitian

Data yang digunakan dalam penelitian ini adalah data yang diperoleh dari kantor Telkom Tanjung Selor Kabupaten Bulungan yaitu berupa data jarak antara titik-titik kabel dari titik sumber sampai ke tujuan. Data tersebut kemudian ditampilkan dalam bentuk gambar jalur dengan titik sumber PT. Telkom sampai ke semua ujung titik.

Data yang di peroleh adalah data STO (Sentral Telepon Otomat), ODC (*Optical Distribution Cabinet*), ODP (*Optical Distribution Point*) dan Pelangan. Lokasi data STO di kantor PT.Telkom jalan Agatis Tanjung Selor dan ODC di Jalan Skip I di depan Kantor Pos Tanjung Selor. Sedangkan lokasi setiap titik ODP dan Pelanggan di Jalan Manggis.

Dari data yang diperoleh, diketahui bahwa terdapat 17 titik dan 20 sisi yang menghubungkan setiap titik, sehingga berdasarkan data tersebut dapat dimodelkan dalam bentuk gambar jaringan (graf) seperti pada gambar 4.1 di bawah ini.



Gambar 4.1 Graf STO, ODC, ODP dan Pelanggan yang saling berhubungan

Keterangan :

1. STO (*sentral telepon otomatis*) = A
2. ODC (*optical distribution cabinet*) = B
3. ODP (*optical distribution point*) = (V_{1-n})
4. Pelanggan = (w_{i-n})

Tabel 4.1 Keterangan Graf

No	Sisi	Bobot (m)
1	(A, B)	1890
2	(B, V_1)	1320
3	(B, V_2)	1080
4	(B, V_3)	907
5	(B, V_4)	809
6	(B, V_5)	708
7	(B, V_6)	702
8	(B, V_7)	263
9	(V_1, W_1)	40
10	(V_1, W_2)	48
11	(V_1, W_3)	30
12	(V_2, W_2)	158
13	(V_4, W_4)	50
14	(V_4, W_5)	121
15	(V_5, W_4)	110
16	(V_5, W_5)	87
17	(V_6, W_6)	80
18	(V_6, W_7)	50
19	(V_6, W_8)	150
20	(V_7, W_8)	80
Jumlah		8683

Sumber: Juarita (2017)

4.4 Algoritma Sollin Dalam Menentukan Pohon Rentang Minimum

Algoritma Sollin merupakan algoritma pencarian pohon merentang minimum dengan cara menghapus sisi dengan bobot paling maksimum. Untuk menentukan pohon merentang minimum dengan menggunakan Algoritma Sollin dapat dilakukan dengan mengikuti beberapa prosedur. Pertama urutkan data berdasarkan sisi dengan nilai bobot yang terurut dari besar ke kecil seperti yang terlihat pada Tabel 4.2 berikut.

Tabel 4.2 Keterangan Graf

No	Sisi	Bobot (m)
1	(A, B)	1890
2	(B, V_1)	1320
3	(B, V_2)	1080
4	(B, V_3)	907
5	(B, V_4)	809
6	(B, V_5)	708
7	(B, V_6)	702
8	(B, V_7)	263
9	(V_2, W_2)	158
10	(V_6, W_8)	150
11	(V_4, W_5)	121
12	(V_5, W_4)	110
13	(V_5, W_5)	87
14	(V_6, W_6)	80
15	(V_7, W_8)	80
16	(V_4, W_4)	50
17	(V_6, W_7)	50
18	(V_1, W_2)	48
19	(V_1, W_1)	40
20	(V_1, W_3)	30

Kemudian kita lakukan penghapusan sisi dimulai dari yang memiliki bobot terbesar dengan syarat tidak membuat graf menjadi tidak terhubung.

1. Sisi 1 :

Sisi AB adalah sisi dengan nilai bobot paling maksimum yaitu 1890. Meski demikian pada sisi AB proses penghapusan tidak dapat dilakukan, hal ini dikarenakan akan menyebabkan graf menjadi tidak terhubung dimana titik A akan menjadi titik terpencil. Oleh karena itu, sisi AB tetap dipertahankan.

2. Sisi 2

Sisi BV_1 adalah sisi maksimum kedua setelah sisi AB dengan bobot 1320.

Pada sisi ini proses penghapusan dapat dilakukan karena graf masih terhubung.

3. Sisi 3

Sisi BV_2 adalah sisi maksimum berikutnya dengan bobot 1080. Pada sisi BV_2 proses penghapusan juga tidak bisa dilakukan, hal ini dikarenakan akan menyebabkan graf menjadi tidak terhubung dimana titik V_2 akan menjadi titik terpencil. Oleh karena itu, sisi BV_2 tetap dipertahankan.

4. Sisi 4

Hal yang sama juga berlaku pada sisi BV_3 dengan bobot 907 dimana titik V_3 akan menjadi titik terpencil jika proses penghapusan dilakukan. Oleh karena itu, sisi BV_3 tetap dipertahankan.

5. Sisi 5

Sisi BV_4 adalah sisi maksimum selanjutnya dengan bobot 809. Pada sisi ini proses penghapusan dapat dilakukan karena graf masih terhubung.

6. Sisi 6

Sisi BV_5 adalah sisi maksimum berikutnya dengan bobot 708, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik V_5 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

7. Sisi 7

Sisi BV_6 adalah sisi maksimum selanjutnya dengan bobot 702. Pada sisi ini proses penghapusan dapat dilakukan karena graf masih terhubung.

8. Sisi 8

Sisi BV_7 adalah sisi maksimum berikutnya dengan bobot 263, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik V_7 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

9. Sisi 9

Sisi V_2W_2 adalah sisi maksimum berikutnya dengan bobot 158, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_2 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

10. Sisi 10

Sisi V_6W_8 adalah sisi maksimum berikutnya dengan bobot 150, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik V_6 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

11. Sisi 11

Sisi V_4W_5 adalah sisi maksimum berikutnya dengan bobot 121. Pada sisi ini proses penghapusan dapat dilakukan karena graf masih terhubung.

12. Sisi 12

Sisi V_5, W_4 adalah sisi maksimum berikutnya dengan bobot 110, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_4 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

13. Sisi 13

Sisi V_5W_5 adalah sisi maksimum berikutnya dengan bobot 87, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_5 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

14. Sisi 14

Sisi V_6, W_6 adalah sisi maksimum berikutnya dengan bobot 80, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_6 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

15. Sisi 15

Sisi V_7, W_8 adalah sisi maksimum berikutnya dengan bobot 80, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_8 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

16. Sisi 16

Sisi V_4, W_4 adalah sisi maksimum berikutnya dengan bobot 50, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik V_4 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

17. Sisi 17

Sisi V_6, W_7 adalah sisi maksimum berikutnya dengan bobot 50, tetapi tidak bisa dihapus karena akan menyebabkan
an graf menjadi tidak terhubung dimana titik W_7 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

18. Sisi 18

Sisi V_1, W_2 adalah sisi maksimum berikutnya dengan bobot 48, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik V_1 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

19. Sisi 19

Sisi V_1, W_1 adalah sisi maksimum berikutnya dengan bobot 40, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_1 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

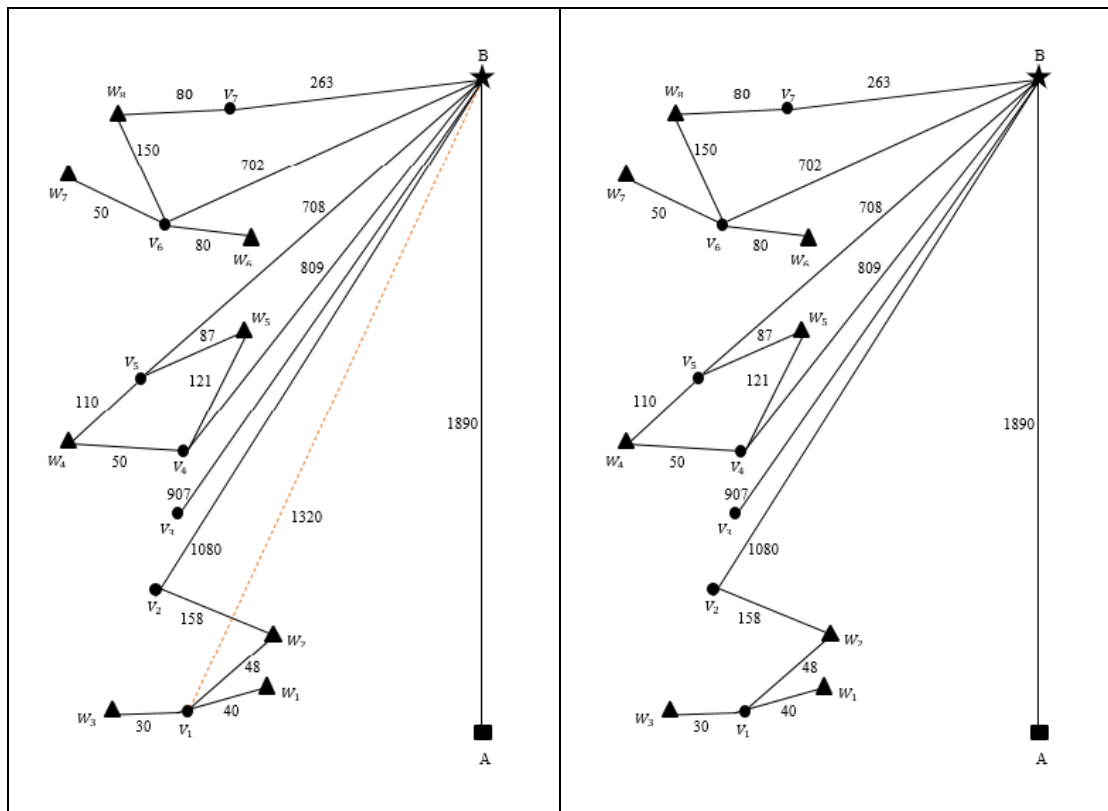
20. Sisi 20

Sisi V_1, W_3 adalah sisi maksimum berikutnya dengan bobot 30, tetapi tidak bisa dihapus karena akan menyebabkan graf menjadi tidak terhubung dimana titik W_3 akan menjadi titik terpencil sehingga sisi ini tetap dipertahankan.

Dari proses penghapusan sisi-sisi di atas terlihat bahwa tahap untuk menentukan pohon merentang minimum dengan menggunakan Algoritma Sollin dapat dilakukan dengan 4 (empat) langkah penghapusan yaitu:

1. Langkah 1

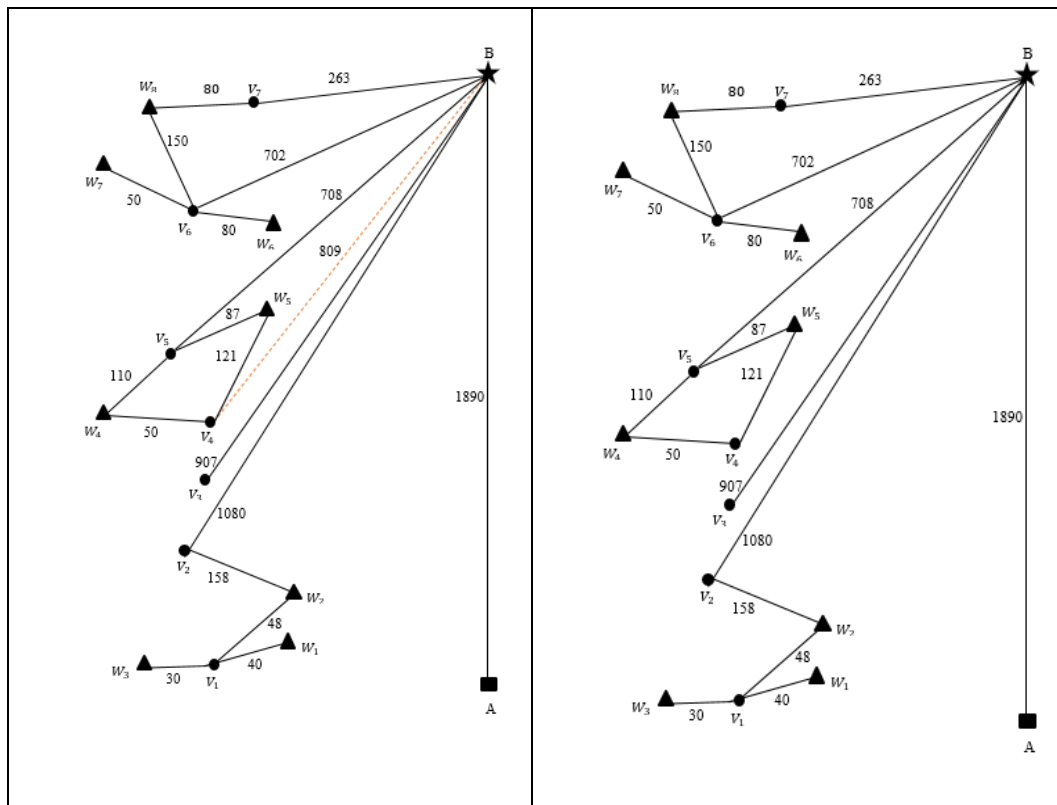
Menghapus sisi kedua yaitu sisi BV_1 dengan bobot 1320, hal ini dikarenakan sisi pertama (AB) tidak dapat dihapus karena akan menyebabkan graf menjadi tidak terhubung sehingga graf menjadi seperti gambar 4.2 berikut



Gambar 4.2 Algoritma Sollin Langkah 1

2. Langkah 2

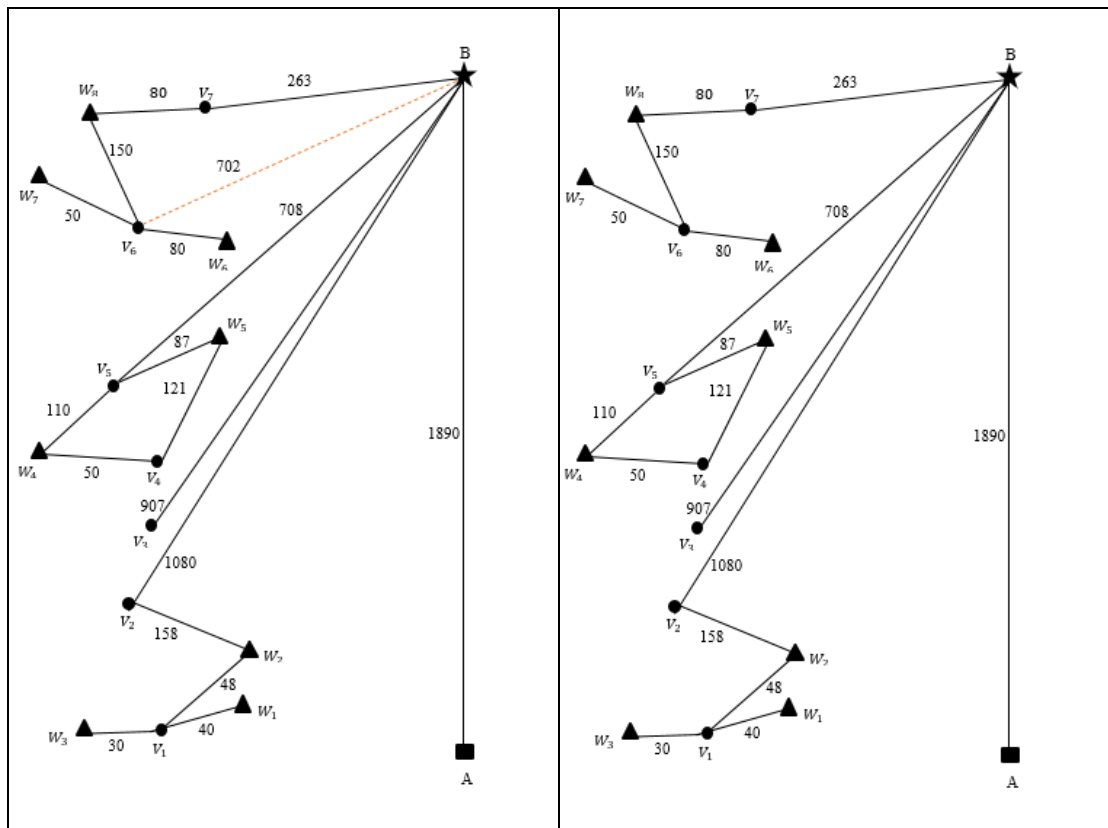
Menghapus sisi kelima yaitu sisi BV_4 dengan bobot 809, hal ini dikarenakan sisi ketiga dan keempat tidak dapat dihapus karena akan menyebabkan graf menjadi tidak terhubung sehingga graf menjadi seperti gambar 4.3 berikut.



Gambar 4.3 Algoritma Sollin Langkah 2

3. Langkah 3

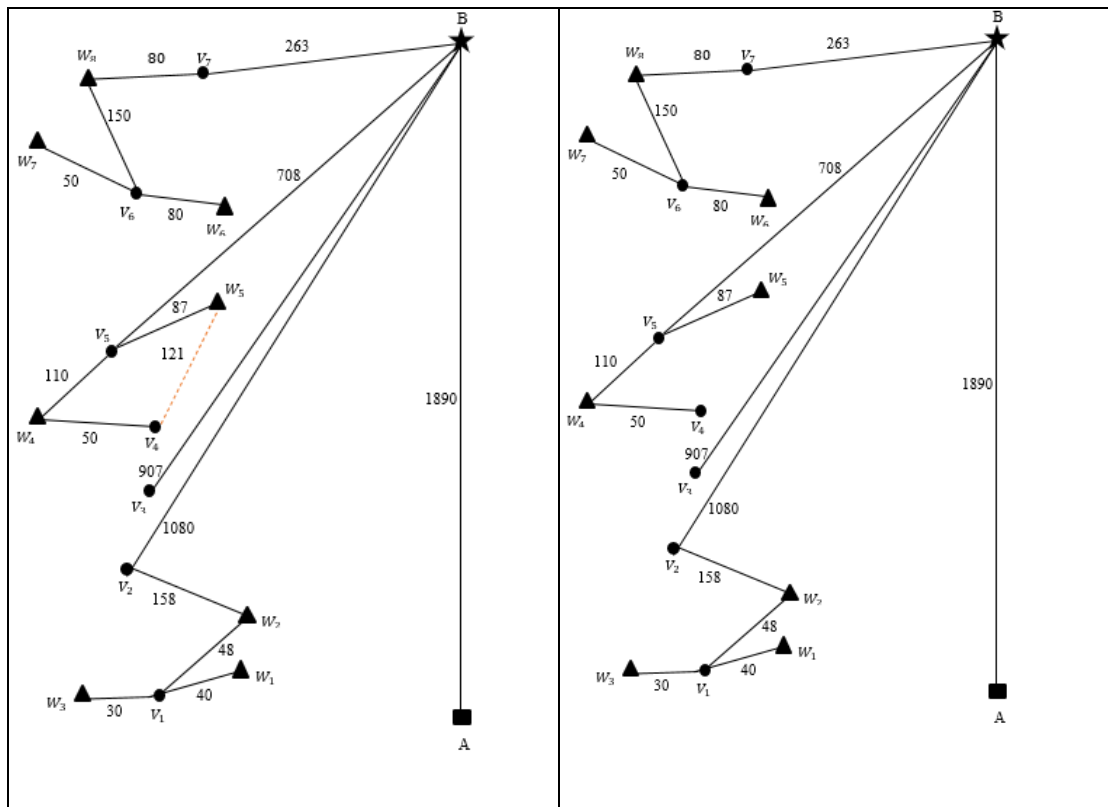
Menghapus sisi ketujuh yaitu sisi BV_6 dengan bobot 702, hal ini dikarenakan sisi keenam tidak dapat dihapus karena akan menyebabkan graf menjadi tidak terhubung sehingga graf menjadi seperti gambar 4.4 berikut.



Gambar 4.4 Algoritma *Sollin* Langkah 3

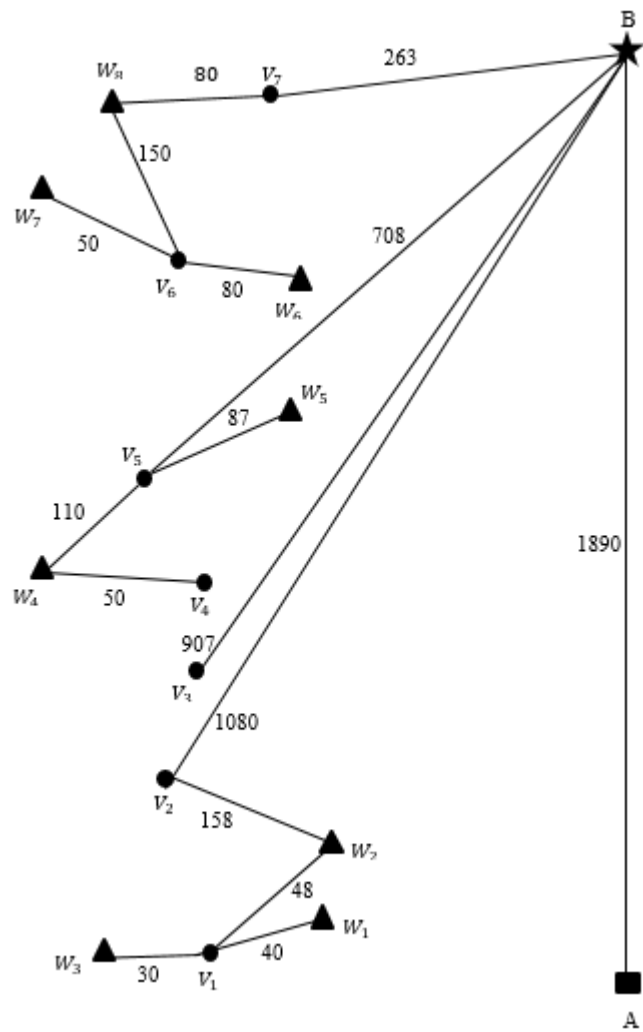
4. Langkah 4

Menghapus sisi sebelas yaitu sisi V_4W_5 dengan bobot 121. Sisi ini merupakan sisi terakhir yang dapat dihapus karena penghapusan pada sisi-sisi yang lain akan menyebabkan graf menjadi tidak terhubung sehingga graf menjadi seperti gambar 4.5 berikut



Gambar 4.5 Algoritma Sollin Langkah 4

Karena sudah tidak terdapat lagi sisi yang dapat dihapus dan itu artinya semua titik sudah terhubung dan sudah diperoleh pohon merentang minimumnya, maka langkah dapat dihentikan. Hasil pohon merentang minimum yang diperoleh dari perhitungan menggunakan Algoritma Sollin dapat dilihat pada gambar 4.6 dibawah ini:



Gambar 4.6 Pohon merentang minimum dengan Algoritma Sollin

Berdasarkan perhitungan manual diperoleh bahwa untuk menentukan jarak (panjang kabel) minimum PT Telkom dengan menggunakan Algoritma *Sollin* membutuhkan 4 langkah untuk dapat menemukan pohon merentang minimum graf dengan 17 titik dan 16 sisi dengan total bobot sebagai berikut:

$$\begin{aligned}
 &= (A, B) + (B, V_2) + (B, V_3) + (B, V_5) + (B, V_7) + (V_2, W_2) + (V_6, W_8) + \\
 &\quad (V_5, W_4) + (V_5, W_5) + (V_6, W_6) + (V_7, W_8) + (V_4, W_4) + (V_6, W_7) \\
 &\quad + (V_1, W_2) + (V_1, W_1) + (V_1, W_3) \\
 &= 1890 + 1080 + 907 + 708 + 263 + 158 + 150 + 110 + 87 + 80 + 80 \\
 &\quad + 50 + 50 + 48 + 40 + 30 \\
 &= 5731
 \end{aligned}$$

BAB V

PENUTUP

1. Kesimpulan

Berdasarkan dari hasil analisis Algoritma *Sollin* pada pemasangan kabel PT. Telkom Tanjung Selor, maka dapat disimpulkan bahwa perhitungan manual Algoritma *Sollin* hanya memerlukan 4 langkah untuk dapat menyelesaikan pohon merentang minimum, hal ini dikarenakan cara kerja Algoritma *Sollin* hanya menghapus sisi terbesar dari sebuah graf. Dari hasil pengujian dalam menyelesaikan pohon merentang minimum menghasilkan total jarak minimum atau panjang kabel terpasang yaitu 5.731 m dengan kata lain pengujian ini telah menghemat kabel sepanjang 2.952 m dari total jarak yaitu 8.683 m.

2. Saran

Berikut ini beberapa saran yang dapat dipertimbangkan untuk memperbaiki penelitian ini dan untuk penelitian kedepannya, yaitu:

1. Untuk kedepannya diharapkan algoritma dapat dikembangkan pada studi kasus yang lainnya
2. Untuk selanjutnya, sistem dapat dikembangkan dengan menggunakan *google maps* sehingga dapat memberikan bentuk jalur peta yang lebih baik.
3. Penelitian selanjutnya bisa membandingkan Algoritma *Sollin* dengan algoritma yang lain dalam menyelesaikan pohon merentang minimum seperti Algoritma *Prim*, Algoritma *Kruskal*, dan Algoritma *Boruvka*.

DAFTAR PUSTAKA

- Munir, Rinaldi. 2010. *Matematika Diskrit*. Informatika. Bandung.
- Prasetyo. 2013. *Penerapan algoritma Krusal dan Sollin pada Pendistribusian*. Air PDAM, Vol. IX, No. 2 Desember 2018.
- Siahaan, R. 2017. Perbandingan Kompleksitas Waktu Teoritis dan Real Time Algoritma Strand sort, Gnome Sort. Tersedia dari Repository USU Database.
- Latifah, Umi, Endang Sugiharti. 2012. *Penerapan Algoritma Prim dan Kruskal pada Jaringan Distribusi Air PDAM Tirta Moedal Cabang Semarang Utara*. UNNES Journal of Mathematic. FMIPA. Universitas Negeri Semarang.
- Khoiroh, Mufidatul. 2010. Keefektian Penggunaan Algoritmat Boruvka, Algoritma Prim, dan Algoritma Sollin dalam Menentukan Pohon Merentang Minimum. Universitas Islam Negeri (UIN) Maulana Malik Ibrahim. Malang
- Maulana, Gun Gun. 2017. Pemebelajaran Dasar Algoritma dan Pemograman Menggunakan El-Goritma Berbasis Web. Jurnal Teknik Mesin.
- Riska. 2012. *Perbandingan Algoritma Kruskal dengan Algoritma Genetika dalam Penyelesaian Masalah Minimum Spanning Tree (MST)*. Skripsi. Fakultas Teknik Universitas Bengkulu.
- Wiria, Deny Nugraha. 2011. *Aplikasi Algoritma Prim untuk Menentukan Minimum Spanning Tree suatu Graf Berbobot dengan Menggunakan Pemrograman Berorientasi Objek*. Jurnal Ilmiah Foristek Vol.1, No.2, September 2011.

