The Euclidean Steiner Tree Problem

Yan Arie Motingo 13518129
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518129@std.stei.kuliah.itb.ac.id

Masalah *The Euclidean Steiner Tree* diselesaikan dengan menemukan tree dengan penjang euclidean terpendek, dengan penambahan titik steiner. Titik steiner digunakan secara luas untuk mendesain banyak hal, seperti jalan tol, dan pipa dalam pertambangan. Tetapi The Euclidean Steiner Tree Problem merupakan NP-Hard problem (non-deterministic polynomial-time hardness), yang berarti tidak ada solusi untuk algoritma waktu polinomnya. Makalah ini meyajikan sejarah singkat dan dua teknik untuk menemukan *Steiner Tree.*

Keywords— Steiner point, Topology, minimal spanning tree, Steiner minimal tree.

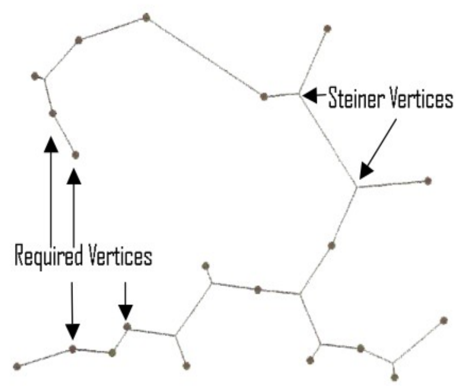
I. PENDAHULUAN

The steiner tree problem, atau *minimum steiner tree problem*, adalah istilah umum untuk kelas masalah dalam hal optimasi pada kombinasi. Sementara *steiner tree problem* dapat dirumuskan dalam sejumlah pengaturan, yang memerlukan interkoneksi yang optimal untuk sebuah set objek dan fungsi tujuan yang telah ditentukan sebelumnya. steiner tree problem dalam grafik dapat dilihat sebagai generalisasi dari dua masalahoptimasi kombinatorial yang terkenal; masalah jalur terpendek dan rentang minimum masalah pohon steiner. Pohon Steiner serupa dengan pohon merentang, tetapi ada suatu perbedaan mendasar antara pohon Steiner dengan pohon merentang. Pada pohon Steiner, titik dan garis dapat ditambahkan pada graf untuk mengurangi panjang sisi dari pohon merentang. Titiktitik yang ditambahkan untuk mengurangi panjang sisi pada graf tersebut disebut titik Steiner. Untuk setiap kumpulan titik yang sama, dapat diperoleh pohon Steiner yang berbeda.

Permasalahan ini meminta kemungkinan jaaringan terpendek untuk n node pada siatu bidang. Ini adalah masalah klasik yang mudah untuk dinyatakan dan dipahami, tetapi sulit untuk dipecahkan. Sejak tahun 1960-an teori jaringan minimal yang rumit semakin banyak dikembangkan, dengan teknik kombinasi, kombinatorik, dan analsisi geometri.

Masalah *The Euclidean Steiner tree* diselesaikan dengan menemukan pohon panjang minimal yang membangun satu set simpul pada bidang sambil memungkinkan untuk penambahan simpul bantu. *The euclidean steiner tree problem* adalah masalah yang berakar sejak abad ke-17 ketika ilmuwan Pierre Fermat mengusulkan masalah berikut; Temukan di bidang itu sebuah titik, dimana jumlah dari jaraknya terhadap ketiga titik yang diberikan minimal. Heinen mengusulkan solusi lengkapnya pada 1834. Butuh sekitar 100 tahun lagi bagi permasalah Fermat ini untuk mendapatkan kepopuleran diantara matematikawan. Pada 1941 setelah Courant dan Robbins mempublikasi “*What is Mathematics?*” nama permasalahannya diganti menjadi *Steiner tree problem*. Masalah ini dinamakan sebagai penghargaan untuk Jacob Steiner, seorang profesor dari Universitas Berlin yang memberikan kontribusi yang besar di bidang matematika. meskipun hubungan antara Jacob Steiner dan permasalahan fermat tidak berhubungan, tetapi pengaplikasian dari S*teiner tree* berdampak besar. seperti pada pemetaan pipa pemanas dalam pembangunan suatu gedung, melacak tata letak antara gerbang logika di sirkuit untuk meminimalkan waktu propagasi, menentukan jalur untuk jalur pipa minyak atau gas alam yang sesingkat mungkin sementara mempertimbangkan medan yang mereka lewati atau hindari, dan jaringan minimal lainnya adalah beberapa dari sekian banyak contoh.

The Euclidean Steiner tree adalah masalah khusus dari steiner tree. graf pohon steiner didefinisikan secara formal: Diberikan graf sederhana G=(V,E) dengan simpulnya yang dipartisi menjadi dua set. tentukan panjang minimum di G, yang mengandung simpul yang dibutuhkan dan adalah subset dari simpul steiner. Tidak seperti Masalah Steiner tree ,Masalah *the Euclidean Steiner tree* tidak menerima *Steiner Vertices* sebagai input, tujuannya adalah bagaimana cara menghubungkan set simpul pada sebuah bidang dengan cara mengenalkan Steiner point untuk minimisasi panjang dari sisi graf.



Saat ini, terdapat dua algoritma pohon merentang minimum yang kerap digunakan, yaitu algoritma Prim dan algoritma Kruskal. Masing – masing algoritma memiliki ciri khas tersendiri tetapi memiliki kesamaan, yaitu menggunakan teknik greedy dalam prosesnya. Perbedaan mendasar dari kedua algoritma ini terletak pada proses pengambilan sisi untuk dimasukkan ke dalam pohon T. Algoritma Kruskal adalah algoritma yang digunakan dalam lingkup kajian teori graf yang berfungsi untuk mencari pohon merentang minimum untuk graf terhubung berbobot G. Artinya, algoritma ini akan mencari himpunan bagian dari sisi yang membentuk graf Gdi mana himpunan bagian ini akan membentuk sebuah pohon yang melingkupi semua simpul yang terkandung pada graf G dengan jumlah bobot sisi yang ada di dalam himpunan bagian tersebut adalah minimum. Algoritma ini direka cipta oleh Joseph B Kruskal

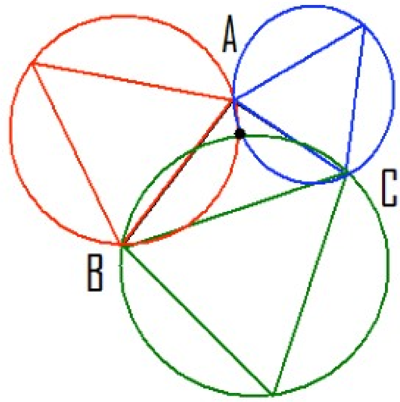
II. LATAR BELAKANG

Pertama-tama kita harus meninjau ulang pekerjaan Torricelli, Simpson, dan Heinen, untuk menetapkan latar belakang dan properti dari *Euclidean Steiner tree problem*. Cara terbaik untuk memahami permasalahan ini dengan memulai dari kecil dan menambahkan node nya satu per satu. Pada untuk kasus n=1 dimana n adalah jumlah terminal, pada kasus ini jarak terminal ke dirinya sendiri adalah nol. Sekarang pada kasus n=2, dimana terminal diberi label A dan B. jarak terpendek dari dua titik adalah garis lurus yang menghubungkannya, maka titik untuk meminimasi jarak dari kedua node berada di tengah garis yang menghubungkan kedua node itu, meskipun pada akhirnya titik ini tidak memberi efek apapun pada jarak minimum dari A ke B atau sebaliknya.

Pada tahun 1640 Torricelli menemukan solusi untuk n=3, dengan node nya yang diberi label A, B, dan C. solusi nya melibatkan pembentukan segitiga dari ketiga node tersebut dan pada ujung dari setiap segitiga tersebut dibentuk tiga segitiga baru dengan lingkaran untuk melingkupi segitiganya dengan ketiga sudut dari segitiga bersinggungan dengan lingkaran. Perpotongan dari ketiga lingkaran tersebut disebut dengan titik Torricelli.

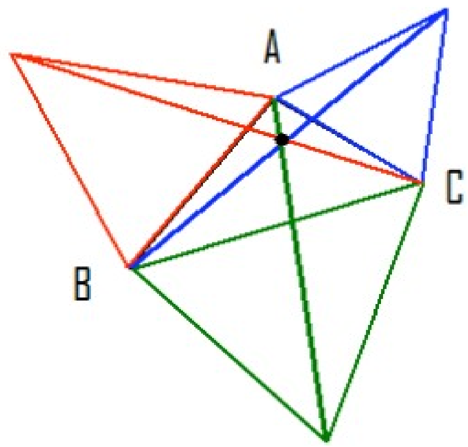
Simpson merancang alternatif untuk titik Torricelli pada tahun 1750, dengan cara membuat tiga segitiga sama sisi, dimana sisi segitiga yang terbentuk dari tiga node tersebut sebagai alasnya, lalu garis Simpson ditarik dari tiga segitiga yang bukan segitiga yang dibentuk dari ketiga node, perpotongan dari ketiga garis itu akan membentuk Torricelli point. Ketiga node tersebut akan membentuk sudut 120° yang membagi sisi dalam dari segitiga yang dibentuk ketiga node, jika salah satu dari sudut tersebut lebih besar dari yang lain maka metode Torricelli tidak akan bekerja.

DIbutuhkan waktu 84 tahun untuk Heinen untuk menyelesaikan permasalahan Fermat. Solusinya mengandung



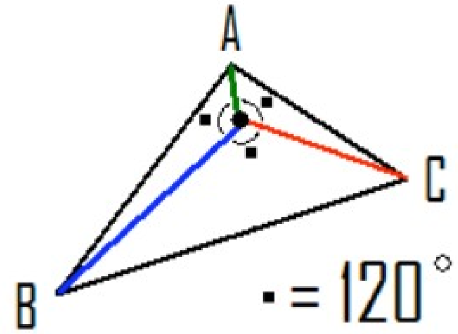
dua kasus:

1. jika salah satu dari sudut dalam yang terbentuk dari ketiga titik A,B, dan C lebih besar dari 120°, maka titik yang yang didefinisikan oleh masalah fermat bertepatan dengan titik yang sudutnya lebih besar dari 120°.
2. jika semua sudut interior yang terbentuk dari tiga titik A, B, dan C kurang dari 120 °, maka titik yang ditentukan oleh masalah Fermat adalah persimpangan garis Simpson atau titik Torricelli.



Setelah penamaan ulang Permasalahan Fermat, titik hasil nya sekarang disebut dengan Steiner point. sebuah titik steiner dengan n=3 dengan tidak ada sudut yang melebihi 120 °, akan membagi setiap sudut dengan sudut 120 °. properti ini disebut dengan angle condition dan sangat penting untuk pembuatan kasus dengan n>3.

Tujuan utama dari *Euclidean Steiner tree problem* adalah menghubungkan set terminal n dengan jarak/jaringan yang minimum. jaringan minimal ini disebut dengan *Steiner Minimal Tree* (SMT), dan ekivalen dengan Steiner tree. pada semua kasus SMT mengandung paling banyak n-2 *Steiner points*. Sebelum mencari SMT untuk suatu graf, terdapat beberapa terminologi yang penting. *Topologies* adalah graf yang menunjukkan koneksi antara Steiner point dengan terminal.

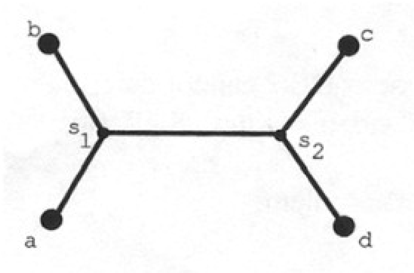


Steiner Topology adalah topologi dimana semua titik steiner memiliki derajat tiga, yang artinya bahwa mungkin untuk membuat panjang sisi nol dimana titik steiner nya ditimpa pada terminal lain atau dirinya sendiri. *Full Steiner Topology* adalah topologi steiner dimana terdapat n-2 titik steiner.

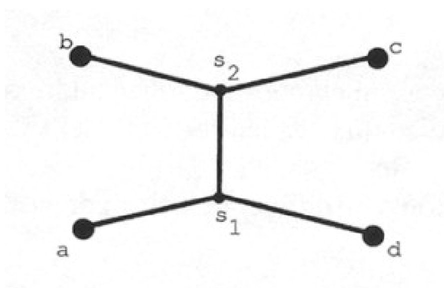
Jumlah full Steiner Topologies untuk graf dengan terminal n diberikan dengan fungsi berikut

$$f(n)=\frac{(2n-4)!}{2^{(n-2)}(n-2)!}$$

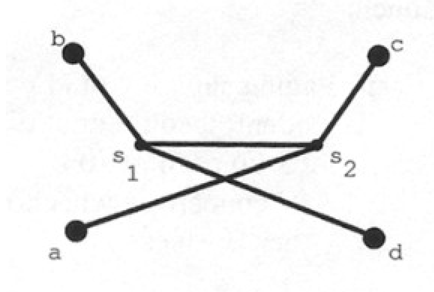
Perhatikan kalau f adalah super exponensial, yang artinya fungsi f akan bertambah dengan lebih cepat daripada fungsi exponensial pada umumnya. f(4)=3, f(5)= 15,f(6)=105, f(7)= 945, f(9)=135135.



(a)



(b)



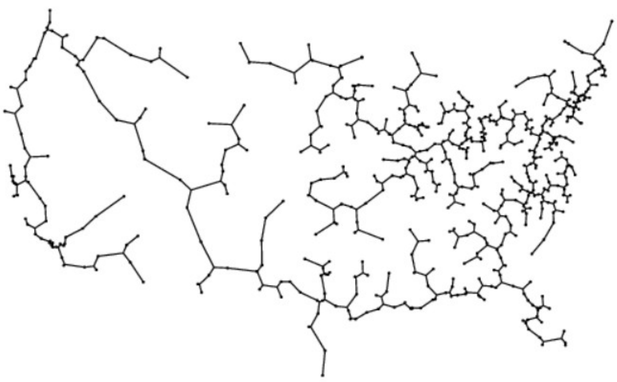
(c)

Pada contoh diatas terdapat 4 terminal dan hanya tiga kemungkinan untuk *Full Steiner Topologies*. Perhatikan bahwa ketiganya memiliki dia titik steiner tetapi topologi masing-masing memiliki total panjang yang berbeda. Contoh (a) merupakan global minimum sehingga merepresentasikan SMT untuk graf dengan n=4. Dikarenakan untuk mencari semua kemungkinan topologi diperlukan perhitungan super exponensial maka semua implementasinya berdasarkan prinsip exponensial.

III. Algoritma Presisi

Masalah Euclidean Steiner tree dapat diselesaikan, tetapi cara pengerjaannya tidak memiliki jawaban yang dapat dicari begitu saja, seperti titik steiner yang dapat diletakkan dimana saja pada bidang. Melzak adalah orang pertama yang menemukan solusi untuk masalah ini pada tahun 1961. pendekatannya adalah dengan mengenumerasi semua Steiner Topologies dan menentukan panjang masing-masing, dan yang dari semua topologi itu yang paling pendek adalah solusi global. dari fungsi super exponensial di atas, proses pencarian secara enumerasi juga memerlukan waktu super exponensial, sehingga metode ini tidak dapat digunakan untuk menyelesaikan masalah dengan terminal yang banyak.

Metode terbaik untuk menyelesaikan masalah ini saat ini adalah dengan metode GeoSteiner algorithm. algoritma ini diciptakan oleh Warme, Winter, dan Zachariasen. Algoritma ini memiliki kemampuan untuk mengatasi masalah hingga 2000 terminal. Gambar dibawah merupakan salah satu contoh dari algoritma GeoSteiner, gambar ini adalah gambar peta Amerika Serikat, yang hanya membutuhkan waktu sekitar beberapa jam pada tahun 1998.



1. Algoritma Parameter

Pertamakali diperkenalkan oleh Downey pada 1990, algoritma ini bertujuan untuk mencari solusi dengan *running time* yang wajar, tetapi dengan input yang kecil. Algoritma ini juga dikenal dengan nama *"fixed parameter tractable"* (FPT), Downey menunjukkan bahwa STP adalah parameter yang pasti dan dapat diselesaikan dengan wakti O(f(k)g(n)), dimana *f* adalah fungsi exponensial, *g* adalah fungsi polinomial, *n* adalah banyaknya node, dan *k* adalah nilai yang lebih dari *n* (banyaknya terminal atau parameter lainnya). Kompleksitasnya selalu ditulis dengan O*(f(k)), dikarenakan algoritmanya memperhitungkan pertumbuhan exponensial dari f(k).

2. Algoritma Dreyfus-Wagner
   Algoritma ini, merupakan algoritma dinamik, yang mengkomputasikan MST untuk terminal yang diinput, dengan fungsi waktu $O*\left(3^k\right)$

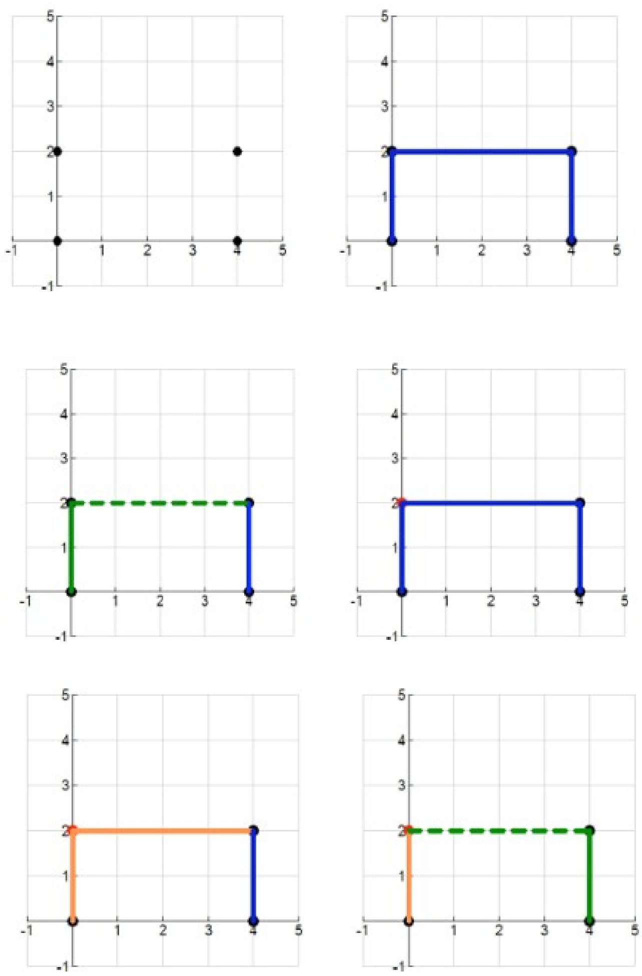
IV. *HEURISTIC AND APPROXIMATION ALGORITHM*

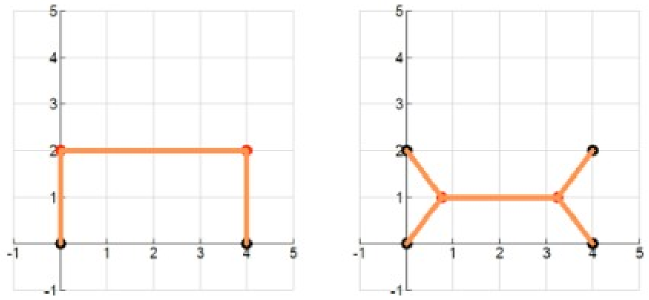
Terdapat banyak cara untuk mendekati sebuah masalah yang rumit untuk menemukan set jawaban yang sesuai, cara paling sederhana biasanya melibatkan pemakaian komponen yang telah ada, termasuk menyelesaikan permasalahan Euclidean Steiner tree. Karena algortima pasti untuk menyelesaikan permasalah ini membutuhkan waktu yang exponensial, maka diperlukan suatu algoritma yang dapat belajar sendiri dan melakukan pendekatan terhadap suatu masalah. metode pertama dengan menggunakan *minimal spanning tree* (MST). metode ini tidak menghasilkan titik steiner, tetapi membangun semua simpul yang dibutuhkan melalui algoritma MST seperti Kruskal atau Prim.

Untuk mengukur kinerja dari algoritma ini, maka perlu dikenalkan Steiner ratio, yang didefinisikan sebagai jumlah panjang MST dari semua simpul dibagi dengan panjang SMT. Gilbert dan Pollack pada 1968 memperkirakan rasio terbaik adalah 1.1547. Untuk meningkatkan akurasi dari algoritma yang dapat meningkatkan kinerja dari stenier ratio, maka diperlukan penambahan titik steiner. Dreyer dan Overton mengembangkan metode ini. idenya adalah untuk mencoba memastikan bahwa angle condition terpenuhi untuk semua titik steiner yang dimasukkan pada graf. berikut adalah pseudo-code , tx,ty,tz melambangkan terminal, sedangkan Sn adalah simpul Steiner:

1. Temukan MST.
2. FOR setiap sisi yang menghubungkan simpul (tx, ty) DO
   a. tentukan sisi terkecil (tx, ty) dimana tz dapat merupakan simpul tetap atau titik steiner
   b. IF sudut nya lebih kecil dari 120° THEN
      I. Letakkan titik steiner yang baru Sn diatas ty.
      II. Singkirkan sisi (tx, ty) dan (ty, tz). sisi ini tidak akan dianggap loop untuk step 2
      III. tambahkan sisi baru (tx, Sn), (ty, Sn), dan (tz, Sn).
3. Jalankan algoritma optimasi lokal pada tree dengan topologi yang baru

Untuk melihat kinerja dari algoritma di atas, anggap kita memiliki 4 terminal t1=(0,0), t2=(0,2), t3=(4,2), t4=(4,0) pada bidang euclidean. langkah pertama adalah menemukan MST antara terminal, yang dapat dilakukan dengan membuat graf yang saling menghubungkan antara keempat terminal dengan Euclidean weights dan menjalankan algoritma Kruskal. pada kasus ini MST nya mengandung sisi (t1, t2), (t2, t3), dan (t3, t4). untuk setiap sisi yang menghubungkan dua terminal kita menentukan sisi mana yang memiliki sudut terkecil. dimulai dari sisi (t1,t2), dapat dilihat kalau sisi (t2,t3), berhubungan dengan sisi (t1,t2) dengan sudut 90°. karena 90° lebih kecil dari 120°, sesuai dengan algoritma di atas, maka perlu ada penambahan simpul steiner diatas terminal t2. Selanjutnya sisi (t1,t2) dan (t2,t3) dihapus dan diganti dengan sisi (t1,s1),(t2,s1) {yang dikenal sebagai zero weight edge karena titik nya ditimpa pada titik lain}, dan (t3,s1). Algoritma ini akan diulang untuk sisi (t3,t4), yang akan menghasilkan titik steiner kedua (pada t3). Pada tahap ini telah terbentuk dua titik steiner dan lima sisi, yang dimana dua sisi tersebut merupakan sisi zero weight edge. pada langkah ketiga topologi yang baru diserahkan ke algoritma yang mengoptimasi topologi berdasarkan terminal dan simpul steiner nya.





Algoritma steiner Insertion heuristic tidak tahu dimana titik steiner akan ditempatkan pada posisi final, tetapi menentukan jika/dimana titik steiner dibutuhkan. Optimasi lokal yang menentukan posisi final titik steiner, yang dapat berubah yang pada akhirnya akan mempengaruhi running time dari algoritma. Sehingga kompleksitas waktu dari algoritma ini adalah O(n3) (Tanpa Memasukkan faktor optimasi), fungsi waktu tersebut adalah fungsi polinomial, tetapi running time nya bergantung pada fungsi optimasi lokal.

Penulis	Rasio	*Running Time*
Kou, Markowsky, Berman ('81)	2	$O(n\log(n)+m)$
Zelikovsky ('93)	1.84	$O(n^3)$
Berman, Ramaiyer ('94)	1.734	$O(n^3)$
Zelikovsky ('95)	1.69	n^k
Karpinski, Zelikovsky ('95)	1.644	n^k
Primel, St ('97)	1.667	$O\left(\left(\frac{\log(1/e)}{e}\right)n^{14}\log(n)\right)$
Hougardy, Promel ('99)	1.59	n^k
Robbins, Zelikovsky ('00)	1.55	n^k

Permasalahan yang ditemukan dalam teori pohon Steiner adalah untuk mencari pohon Steiner dari suatu graf, digunakan algoritma yang memiliki kompleksitas cukup besar untuk jumlah titik masukan yang besar (> 30). Selain itu, ditemukan beberapa hal lain yang menjadi kendala dalam mencari dan mengaplikasikan pohon Steiner dalam kehidupan sehari-hari. Beberapa hal yang menjadi hambatan dalam mengaplikasikan pohon Steiner, Banyaknya titik yang akan dihubungkan dengan pohon Steiner. Pohon Steiner antara 2 titik dapat dicari dengan mudah, sama seperti mencari jalur terpendek antara 2 titik tersebut, sedangkan pohon Steiner untuk setiap titik pada graf G, sama dengan mencari pohon merentang minimum pada graf tersebut. Namun, algoritma pencarian pohon Steiner termasuk kategori NP (non-polynomial) complete. Dengan demikian, pertambahan banyak titik yang akan dihubungkan menjadikan komplesitas algoritma semakin besar dan tidak dapat ditentukan besaran pertambahannya. Adanya aturan penggunaan garis/sisi. Pada sebagian besar permasalahan dalam kehidupan nyata, pembentukan pohon Steiner mendapat aturan tambahan, yaitu garis/sisi yang boleh ditambahkan hanya berupa garis horizontal/vertikal saja, seperti pada pembuatan desain chip. Permasalahan ini dikenal dengan istilah rectilinear Steiner. Permasalahan ini dapat menambah kompleksitas dari algoritma yang digunakan. Adanya titik Steiner yang harus dibuat. Dalam hampir semua kasus pembuatan pohon Steiner, diperlukan adanya titik Steiner untuk menghasilkan jalur yang terpendek. Namun, permasalahan yang dihadapi adalah titik-titik Steiner tersebut tidak diketahui berapa banyak dan dimana harus diletakan. Hal tersebut menjadi salah satu hal yang menyebabkan kompleksitas algoritma tersebut meningkat.

V. Kesimpulan

The Euclidean Steiner Tree Problem merupakan permasalah yang menarik, hanya dengan menambah titik steiner pada set terminal yang dibutuhkan maka permasalahan nya menjadi NP-Hard (non-deterministic polynomial-time hardness). Maka dari itu, algoritma untuk permasalahan ini terbagi atas dua kelas; algoritma presisi, dan heuristic or approximation altorithm. Algoritma prasisi, memiliki kelebihan dimana perhitungan jarak terpendek dapat dicari secara presisi, tetapi running time yang dibutuhkan sangat lama, algoritma presisi terbaik untuk saat ini adalah GeoSteiner versi 3.1, dan diimplementasikan oleh Warme, winter, dan Zacharisen. Algoritma aproksimasi, didasarkan pada MST, pada dasarnya MST adalah pohon steiner tanpa titik steiner. Seperti yang ditunjukkan pada Steiner Insertion Algorithm, kecepatan, ukuran input dan keuntungan utama dari algoritma ini.

Walaupun banyak terdapat kendala dalam penerapan teori pohon Steiner ini, namun terdapat beberapa cara untuk mengoptimalkan pencarian pohon Steiner dari suatu graf. Caranya, buat graf yang menggambarkan titik-titik masukan, dengan panjang sisi (i, j) sama dengan jarak antara titik i dan j. Kemudian cari pohon merentang minimum dari graf ini. Maka akan ditemukan hasil yang merupakan pendekatan baik untuk pohon Steiner biasa maupun rectilinear Steiner. Kasus terburuk untuk cara pendekatan seperti ini adalah jika 3 titik membentuk segitiga sama sisi. Pohon merentang minimun akan mengandung 2 sisi dengan panjang = 2, namun seharusnya panjang pohon Steiner terkecil yang dapat dicapai dengan menggunakan bantuan titik Steiner di tengah-tengah ketiga titik tersebut adalah $\sqrt{3}$Perbandingan perbedaan hasil ini dan hasil pendekatan adalah $\frac{\sqrt{3}}{2}=0,866$Angka perbandingan perbedaan hasil ini selalu diperoleh untuk setiap hasil pendekatan dan hasil optimal pada pohon Steiner. Pada rectilinear Steiner, perbandingan perbedaan hasil pendekatan dengan hasil optimal selalu . $\geq \frac{2}{3}=0,6667$

REFERENCES

[1] Zachariasen, M. Algorithms for Plane Steiner Tree Problems. Ph.D. Thesis, Department of Computer Science, University of Copenhagen, 1998.

[2] Vijay V. Varzirani. Approximation Algorithms. Springer, 2001.

[3] Hochbaum, D. Approximation Algorithms for NP-Hard Problems. PWS Publishing 1997.

[4] P. Winter, M. Zachariasen. Large Euclidean Steiner Minimum Trees in an Hour. ISMP 1997.

[5] H. Zhou. Efficient Steiner tree construction based on spanning graphs. ISPD 2003.

[6] http://personal.denison.edu/~havill/272S04/papers/steiner.pdf

[7] Courant R, Robbins H (1941) What is Mathematics? Oxford University Press, London

[8] Dahan-Dalmedico A (1986) Un texte de philosophie mathematique de Gergonne. ´ Revue d'histoire des sciences 39:97–126

[9] A.R. Conn and M.L. Overton. A primal-dual interior point method for minimizing a sum of Euclidean vector norms, July 1994. Draft copy of incomplete manuscript.

[10] M.R. Garey, R.L. Graham, and D.S. Johnson. The complexity of computing steiner minimal trees. SIAM Journal on Applied Mathematics, 32:835–859, 1977.

[11] F.K. Hwang, D.S. Richards, and P. Winter. The Steiner Tree Problem. Annals of Discrete Mathematics. Elsevier Science Publishers B.V., Amsterdam, 1992.

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, 5 Desember 2019

Yan Arie Motinggo 13518129