

Penyelesaian Traveling Salesman Problem dengan Algoritma Heuristik

Filman Ferdian - 13507091

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung, e-mail: if17091@students.if.itb.ac.id

Abstrak – *Traveling salesman problem (TSP) adalah permasalahan yang sangat terkenal dalam Teori Graf. Permasalahan pada TSP adalah mencari sirkuit terpendek pada suatu graf tidak berarah yang berasal dari suatu simpul dengan melewati seluruh simpul dan kembali ke simpul asal. Sirkuit ini disebut juga sirkuit Hamilton. Makalah ini membahas, penyelesaian TSP menggunakan algoritma heuristik. Algoritma heuristik adalah algoritma yang menghasilkan solusi terbaik pada permasalahan yang merupakan bagian atau beririsan dengan suatu persoalan yang lebih kompleks. Pada penyelesaian TSP, algoritma ini mencari minimum spanning tree dari seluruh simpul. Kemudian, algoritma ini membentuk sirkuit euler untuk aproksimasi solusi lalu menangani simpul yang terlalui lebih dari satu kali sehingga diharapkan dapat memberikan solusi optimal dalam menyelesaikan persoalan TSP.*

Kata kunci: graf ,traveling salesman problem, algoritma heuristik.

1. PENDAHULUAN

Permasalahan *Traveling salesman problem* (TSP) merupakan salah satu persoalan yang terkenal dalam Teori Graf. Nama teori ini diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota. Inti permasalahan ini adalah menemukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang tersebut berangkat dari satu kota dan menyinggahi setiap kota tepat satu kali dan kembali ke kota asal.

Permasalahan ini dapat dimodelkan dengan baik melalui teori graf. Pada TSP, graf yang digunakan adalah graf tidak berarah dengan bobot pada setiap sisi. Solusi dari permasalahan ini dapat membantu dalam mencari beragam permasalahan lain seperti, efisiensi pengiriman surat atau barang, perancangan pemasangan pipa saluran, proses pembuatan PCB (*Printed Circuit Board*) dan lain-lain.

Penentuan permasalahan ini juga tidak hanya terbatas pada masalah jarak. TSP dapat digunakan untuk memodelkan permasalahan waktu yang dibutuhkan untuk melalui beberapa simpul.

Permasalahan ini terlihat sepele untuk jumlah simpul yang sedikit. Namun, TSP akan menjadi permasalahan yang sangat kompleks jika jumlah simpul mencapai ratusan, ribuan bahkan jutaan. Oleh karena itu, solusi dalam penyelesaian masalah ini sangat menarik. Apalagi untuk membentuk suatu algoritma yang dapat menyelesaikan TSP dengan

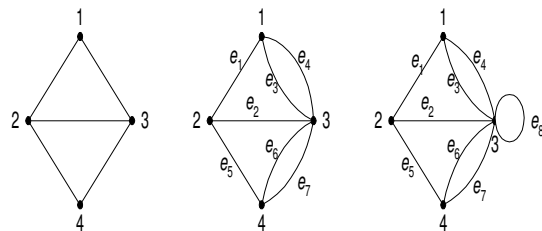
efisiensi tinggi sangat sulit.

Terdapat beragam teori algoritma yang dapat digunakan untuk menyelesaikan permasalahan ini seperti, algoritma *brute force*, algoritma *greedy*, algoritma genetik, dll. Algoritma heuristik adalah salah satu algoritma yang dapat menangani permasalahan ini. Algoritma heuristik adalah algoritma yang mencari solusi terbaik untuk kasus yang merupakan bagian atau irisan dari permasalahan total dengan harapan dapat menghasilkan solusi optimal untuk keseluruhan kasus dengan melalui proses tambahan.

2. GRAF

Graf merupakan suatu alat bantu untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.

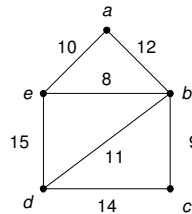
Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$. V (*vertex*) merupakan himpunan tidak kosong dari simpul. E (*edge*) merupakan himpunan sisi yang menghubungkan sepasang simpul. Simpul pada graf umumnya dinomori dengan huruf atau angka. Sedangkan sisi pada graf umumnya dinamai dengan himpunan simpul yang dihubungkan oleh sisi tersebut.



(Contoh Graf)

Graf terdiri dari berbagai jenis diantaranya graf sederhana graf tak sederhana, selain itu juga ada graf berarah. Graf sederhana adalah graf yang tidak memiliki sisi ganda dan juga gelang. Sisi ganda merupakan kondisi ketika dua buah simpul memiliki lebih dari satu sisi. Sisi gelang adalah ketika ada sisi yang berasal dari satu simpul dan kembali pada simpul tersebut. Sedangkan graf tak sederhana adalah graf yang memiliki sisi ganda dan/atau gelang. Graf berarah merupakan graf yang setiap sisinya memiliki orientasi arah dari suatu simpul ke simpul lainnya.

Graf memiliki beberapa terminologi (istilah) dasar diantaranya, bersisian, bertetangga, simpul terpercil, graf kosong, derajat, lintasan, siklus, sirkuit, terhubung, upagraf, komplemen upagraf, upagraf merentang, cut-set, dan graf berbobot. Graf berbobot adalah graf yang memiliki nilai pada setiap sisinya. Istilah lain dari graf berbobot adalah graf berlabel.

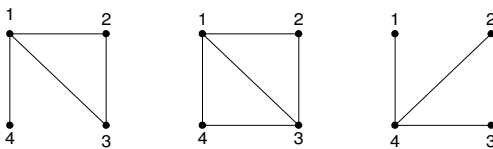


(Contoh Graf Bobot)

Graf banyak digunakan dalam beragam bidang pengetahuan sebagai alat untuk memodelkan suatu permasalahan. Contoh terapan dari graf dalam bidang pemodelan dapat berupa model Rangkaian listrik, model isomer senyawa karbon dan juga model transaksi konkuren pada basis data terpusat.

3. SIRKUIT DAN LINTASAN HAMILTON

Lintasan Hamilton ialah lintasan yang melalui tiap simpul di dalam graf tepat satu kali. Sirkuit Hamilton ialah sirkuit yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul asal (sekali-gus simpul akhir) yang dilalui dua kali. Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi-Hamilton.



(Contoh graf Hamilton)

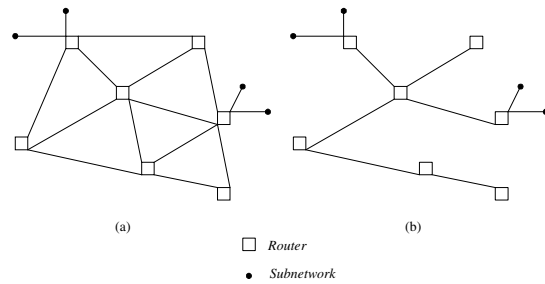
Beberapa teorema tentang sirkuit Hamilton:

- Syarat cukup (jadi bukan syarat perlu) supaya graf sederhana G dengan $n (\geq 3)$ buah simpul adalah graf Hamilton ialah bila derajat tiap simpul paling sedikit $n/2$ (yaitu, $d(v) \geq n/2$ untuk setiap simpul v di G).
- Setiap graf lengkap adalah graf Hamilton.
- Di dalam graf lengkap G dengan n buah simpul ($n \geq 3$), terdapat $(n - 1)!/2$ buah sirkuit Hamilton.
- Di dalam graf lengkap G dengan n buah simpul ($n \geq 3$ dan n ganjil), terdapat $(n - 1)/2$

buah sirkuit Hamilton yang saling lepas (tidak ada sisi yang beririsan). Jika n genap dan $n \geq 4$, maka di dalam G terdapat $(n - 2)/2$ buah sirkuit Hamilton yang saling lepas.

4. POHON MERENTANG MINIMUM

Setiap graf terhubung memiliki pohon merentang. Pohon merentang dapat digunakan dalam menentukan jumlah ruas jalan seminimum mungkin yang menghubungkan semua kota sehingga setiap kota tetap terhubung satu sama lain atau perutean (routing) pesan pada jaringan komputer.



(a) Jaringan komputer, (b) Pohon merentang *multicast*

Graf terhubung berbobot memiliki beberapa pohon merentang. Pohon merentang adalah pohon yang dibangun dengan semua simpul dari suatu graf. Pohon merentang yang memiliki total bobot minimum disebut pohon merentang minimum.

Pada proses penentuan pohon merentang minimum terdapat dua algoritma yang sangat populer yaitu algoritma prim dan algoritma kruskal.

Algoritma Prim

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.

Algoritma Kruskal

1. sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya – dari bobot kecil ke bobot 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.

5. TRAVELING SALESMAN PROBLEM

Traveling salesman problem adalah suatu permasalahan dalam bidang diskrit dan optimasi kombinatorial. Sebagai permasalahan kombinatorial, persoalan ini tergolong memiliki kemungkinan jawaban yang sangat banyak. Permasalahan ini diilhami oleh masalah seorang pedagang yang mengelilingi beberapa kota.

Permasalahan matematik yang berkaitan dengan *Traveling Salesman Problem* mulai muncul sekitar tahun 1800-an. Masalah ini dikemukakan oleh dua orang matematikawan, yaitu Sir William Rowan Hamilton yang berasal dari Irlandia dan Thomas Penyngton Kirkman yang berasal dari Inggris. Bentuk umum dari persoalan TSP pertama kali dipelajari oleh para matematikawan mulai tahun 1930-an oleh Karl Menger di Vienna dan Harvard. Persoalan tersebut kemudian dikembangkan oleh Hassler Whitney dan Merrill Flood di Princeton.

Berdasarkan kesesuaian dengan nama, deskripsi persoalan adalah sebagai berikut: diberikan sejumlah kota, tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan.

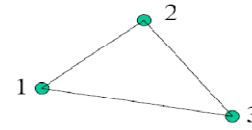
Kota dapat dinyatakan sebagai sebuah simpul graf, sedangkan sisi menyatakan jalan yang menghubungkan antara dua kota. Bobot pada sisi menyatakan jumlah antara dua buah kota. Persoalan ini adalah persoalan yang menentukan sirkuit Hamilton dengan sisi memiliki bobot minimum pada suatu graf terhubung. Selain permasalahan pedagang keliling, terdapat beberapa kasus yang merupakan penerapan dari pencarian sirkuit Hamilton dengan bobot minimum.

Jika setiap simpul pada graf bobot mempunyai sisi ke simpul lain maka, graf tersebut adalah graf lengkap berbobot. Pada sembarang graf lengkap dengan n buah simpul ($n > 2$), jumlah sirkuit Hamilton yang berbeda adalah $(n-1)!/2$.

Persoalan TSP adalah persoalan yang sulit jika dipandang dari segi komputasi. Secara teoritis, TSP dapat diselesaikan dengan mengenumerasi seluruh kemungkinan sirkuit Hamilton yang ada lalu menghitung bobot total dari seluruh sirkuit Hamilton.

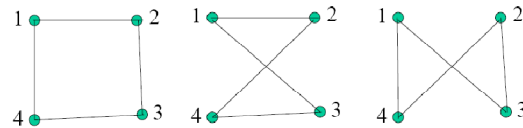
Berikut beberapa contoh penanganan permasalahan Traveling Salesman Problem:

TSP dengan 3 kota:



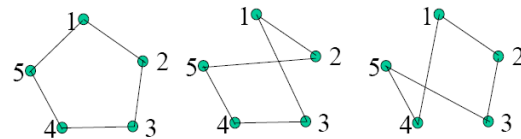
Permasalahan 3 kota tidak memerlukan komputasi karena jumlah kemungkinan solusi hanya 1.

TSP dengan 4 kota:



Permasalahan 4 kota tidak memerlukan komputasi karena jumlah kemungkinan hanya $(4-1)! / 2 = 3$.

TSP dengan 5 kota:



Permasalahan 5 kota mulai memerlukan komputasi karena jumlah kemungkinan $(5-1)! / 2 = 12$

Permasalahan dengan jumlah kota yang semakin banyak akan menghasilkan semakin banyak kemungkinan. Untuk 20 kota akan menghasilkan $19!/2$ kemungkinan atau sekitar $6,08 \cdot 10^{16}$. Namun, tidak semua traveling salesman problem melibatkan graf lengkap. Pada pemodelan kota, kasus ini dapat terjadi ketika ada kota yang tidak memiliki jalan penghubung langsung.

Sampai saat ini, belum ada algoritma yang dapat menghasilkan solusi terbaik dengan waktu yang lebih cepat untuk persoalan TSP. Algoritma yang ditemukan sampai saat ini belum dapat melakukan komputasi dengan banyak input untuk menentukan solusi TSP. Hal yang dapat dilakukan adalah mencari solusi paling optimal yaitu solusi yang memiliki kesesuaian antara waktu pemrosesan dengan solusi. Mengingat waktu yang digunakan oleh persoalan dengan komputasi tingkat tinggi seperti TSP, maka kita harus puas dengan hasil paling optimal bukan paling benar.

6. ALGORITMA HEURISTIK

Algoritma Heuristik adalah algoritma yang mencari nilai optimal dari bagian atau irisan suatu permasalahan utama lalu melakukan proses penyempurnaan sehingga mencapai solusi terbaik yang diinginkan.

Algoritma heuristik adalah salah satu algoritma yang bisa diterapkan dalam menangani permasalahan *Traveling Salesman Problem*. Terdapat beberapa kategori dari algoritma heuristik yang dapat menangani kasus TSP. Diantaranya:

- Algoritma tetangga terdekat: Prosesnya adalah dengan mengunjungi satu kota terdekat yang belum dikunjungi hingga kembali ke tempat asal.
- *2-opt heuristic*: Prosesnya adalah dengan mengganti 2 sisi dengan 2 sisi baru yang menghubungkan bagian yang sama sehingga menghasilkan tur yang lebih optimal.
- *K-opt heuristic*: Prosesnya mirip dengan algoritma sebelumnya namun dengan mengganti sisi sebanyak k dengan sisi baru yang menghubungkan simpul semula sehingga menghasilkan tur yang lebih optimal.
- *V'-opt heuristic*: Prosesnya juga mirip dengan *k-opt heuristic*. Perbedaannya adalah 2-opt merubah sisi yang telah menjadi bagian dari sisi optimal, sedangkan *V'-opt* mengubah sisi yang belum masuk ke bagian dari tur yang belum optimal.

Algoritma yang akan dibahas dalam makalah ini adalah algoritma heuristik yang mencari nilai minimum bobot dengan menggunakan *spanning tree* sehingga menghasilkan irisan dari graf yang memiliki nilai optimal. Proses selanjutnya adalah membentuk sirkuit euler yang lebih mudah dibentuk daripada sirkuit Hamilton sehingga dapat menjadi aproksimasi dari solusi *Traveling Salesman Problem*. Setelah itu, proses dilanjutkan dengan perbaikan simpul yang dilalui lebih dari 1 kali sehingga menghasilkan solusi paling optimal.

7. ANALISIS PENYELESAIAN TSP DENGAN ALGORITMA HEURISTIK

Algoritma untuk menyelesaikan permasalahan TSP untuk kondisi diatas adalah sebagai berikut:

1. Cari *minimum spanning tree* yang menghubungkan tiap n simpul dari graf. Hasil pencarian *minimum spanning tree* ini dinamakan A.
2. Tentukan simpul graph yang berderajat ganjil, jika k merupakan jumlah simpul graph berderajat ganjil dari n simpul maka k pasti

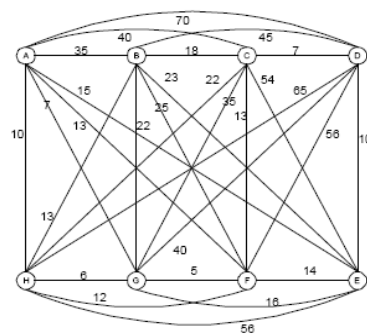
bilangan genap. Kita pasang k simpul sehingga panjang dari cabang yang menghubungkan simpul-simpul tersebut minimum. K simpul dengan tiap cabangnya yang diperoleh dari memasang masing-masing simpul dari k simpul tersebut membentuk jaringan yang dinamakan B. Jaringan A dan B yang sudah terbentuk kita gabungkan menjadi jaringan C.

3. Sekarang jaringan C tidak mempunyai simpul berderajat ganjil. Kita dapat menggambarkan sirkuit Euler pada jaringan C. Sirkuit Euler merupakan aproksimasi solusi dari *Traveling Salesman Problem*.
4. Periksa tiap nodes pada jaringan C yang dikunjungi lebih dari satu kali dan perbaiki solusi *Traveling Salesman Problem* dari langkah 3 dengan menerapkan ketidak-samaan dibawah ini:

$$l(a,b) < l(a,c) + l(c,b)$$

Berikut akan diberikan suatu kasus untuk dijadikan permasalahan untuk ditentukan sirkuit minimalnya. Misal, diberikan graf lengkap berbobot dengan 8 simpul yang saling terhubung.

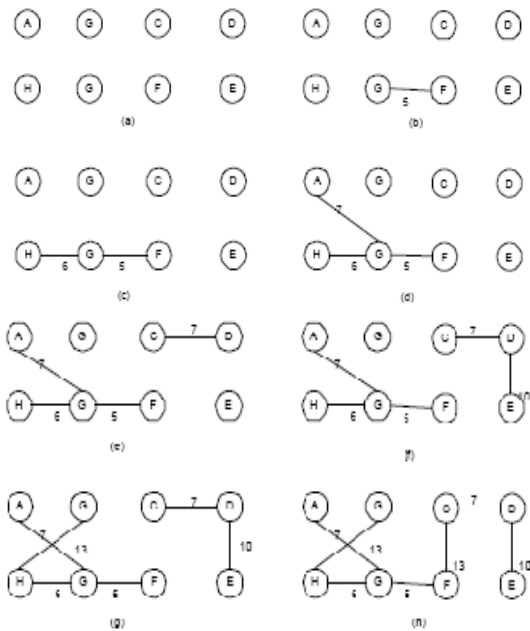
	A	B	C	D	E	F	G	H
A	0	35	40	70	15	13	7	10
B	35	0	18	45	23	25	22	13
C	40	18	0	7	54	13	35	22
D	70	45	7	0	10	56	40	65
E	15	23	54	10	0	14	16	56
F	13	25	13	56	14	0	5	12
G	7	22	35	40	16	5	0	6
H	10	13	22	65	56	12	6	0



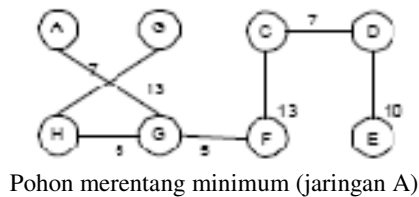
Graf lengkap dari permasalahan yang dicantumkan pada tabel

Jumlah kemungkinan pada penentuan TSP dengan 8 input ini ada $(8-1)! / 2 = 2520$ kemungkinan sehingga proses enumerasi lalu membandingkan seluruh total bobot akan menjadi sangat rumit. Metode Heuristik menjadi salah satu solusi yang dapat diterapkan pada kasus ini.

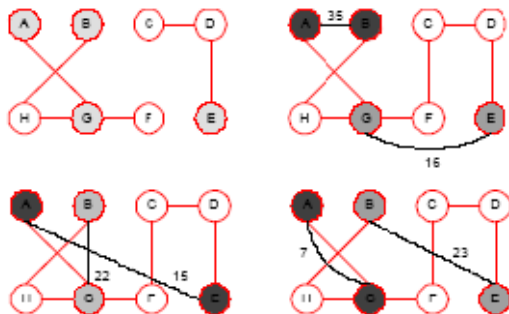
Proses awal dari penyelesaian kasus ini adalah dengan membentuk pohon merentang minimum dari seluruh simpul. Seperti pada gambar berikut:



Setelah proses algoritma kruskal dijalankan maka akan terbentuk pohon merentang minimum. Pohon merentang minimum telah memberikan jalur yang akan mengunjungi seluruh kota dengan jumlah jarak jalan total paling minimal. Namun, hasil algoritma ini belum membentuk sirkuit.

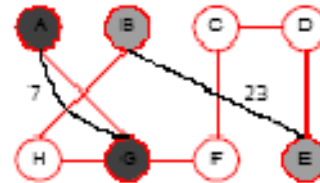


Setelah terbentuk pohon merentang minimum, proses dilanjutkan dengan mencari simpul yang berderajat ganjil. Simpul berderajat ganjil akan dihubungkan sehingga menjadi berderajat genap dengan bobot sisi tambahan minimum.



Setelah proses ini selesai maka terbentuk suatu sirkuit euler. Sirkuit ini merupakan aproksimasi dari solusi TSP. Pada hasil proses ini sisi dari A dan G, lalu B dan E menjadi jaringan B.

Proses ketiga, jaringan C terbentuk dengan seluruh simpul berderajat genap yang merupakan sirkuit euler. Sirkuit ini merupakan aproksimasi dari solusi TSP yang diinginkan.

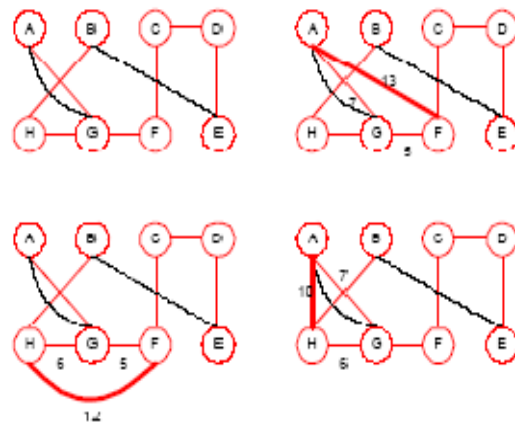


Jaringan C yang membentuk sirkuit euler

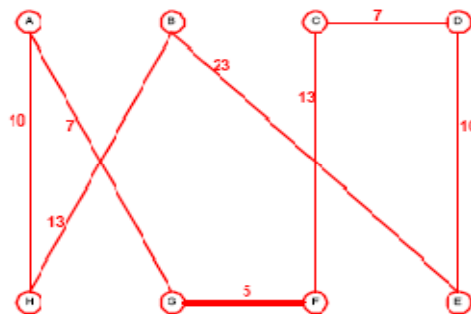
Langkah berikutnya adalah memeriksa apakah ada simpul yang dikunjungi lebih dari satu kali lalu kita lakukan proses perbaikan perjalanan traveling salesman problem dengan memeriksa ketidak samaan:

$$l(a,b) < l(a,c) + l(c,b)$$

Pemeriksaan pada graf mengimplikasikan pemeriksaan pada perjalanan dari simpul A ke simpul H.



Setelah proses berakhir, maka terbentuk suatu sirkuit Hamilton minimum atau solusi dari permasalahan *traveling salesman problem*. Sehingga diperoleh gambar berikut.



8. KESIMPULAN

1. *Traveling salesman problem* adalah suatu permasalahan dalam menentukan sirkuit terpendek dari suatu simpul ke seluruh simpul lain tepat satu kali dan kembali ke simpul asal.
2. Algoritma heuristik adalah algoritma yang mencari nilai optimal dari suatu bagian permasalahan utama atau beririsan dari permasalahan utama dengan harapan dapat mencapai solusi optimal secara keseluruhan.
3. Algoritma heuristik dapat digunakan dalam menangani permasalahan *Traveling salesman problem*.
4. Proses penyelesaian melalui algoritma heuristik adalah menentukan *minimum spanning tree*, menyusun sirkuit euler sebagai aproksimasi solusi lalu mengganti lintasan bagi simpul yang dilalui lebih dari satu kali.
5. Penyelesaian dengan algoritma heuristik menghabiskan waktu proses cukup besar namun jauh lebih cepat daripada proses enumerasi. Namun, hasil perhitungan dari algoritma ini menghasilkan hasil minimum yang optimal
6. Penyelesaian permasalahan *Traveling Salesman Problem* adalah permasalahan yang memiliki kompleksitas komputasi tinggi dan belum ada algoritma yang mangkus untuk menangani kasus ini dengan jumlah simpul yang banyak dalam waktu relative cepat dan hasil sangat optimal.

DAFTAR REFERENSI

- [1] Munir, Rinaldi. 2005, Matematika Diskrit. Bandung: Penerbit Informatika.
- [2] Algoritma Heuristik untuk Traveling salesman Problem.
“http://kur2003.if.itb.ac.id/file/TSP_heuristic_.pdf”. Tanggal akses: 29 Desember 2008
- [3] Basuki, Achmad. 2004. Penyelesaian TSP (Traveling salesman Problem menggunakan Monte Carlo. “<http://lecturer.eepis-its.edu/~basuki/lecture/TSPMonteCarlo.pdf>”. Tanggal akses: 29 Desember 2008
- [4] Wikipedia. 2009. Traveling Salesman Problem. “http://en.wikipedia.org/wiki/Traveling_salesman_problem”. Tanggal akses: 5 Januari 2009.
- [5] Wikipedia. 2009. Heuristic Algorithm. “[http://en.wikipedia.org/wiki/Heuristic_\(computer_science\)](http://en.wikipedia.org/wiki/Heuristic_(computer_science))”. Tanggal akses: 5 Januari 2009