

# STUDI DAN IMPLEMENTASI GRAF DALAM PENENTUAN RUTE

Wahyu Adhi A – NIM : 13505024

Program Studi Teknik Informatika, Institut Teknologi Bandung

Jl. Ganesha 10, Bandung

E-mail : [if15024@students.if.itb.ac.id](mailto:if15024@students.if.itb.ac.id)

## Abstrak

Graf merupakan salah satu cabang ilmu matematika yang merepresentasikan objek – objek diskrit dan hubungan antara objek – objek tersebut. Representasi visual dari graf adalah dengan menyatakan obyek dengan noktah dan hubungan antara objeknya dengan garis. Untuk selanjutnya kita sebut noktah pada graf sebagai simpul (vertex) dan garis pada graf sebagai sisi (edge). Pada saat membicarakan graf tentu kita tidak boleh lupa pada salah satu struktur graf yang cukup penting yaitu pohon (beberapa referensi menetapkan pokok bahasan “pohon” pada bab yang berbeda karena luasnya bahasan tentang bab ini.). Pohon adalah graf berarah / tak berarah yang tidak membentuk sirkuit. Penggunaan struktur graf ini sangat penting dalam bidang informatika dan ilmu komputer.

Salah satu kegunaan graf yang cukup penting adalah dalam hal pengrutean. Perutean yaitu kegiatan membuat rute dengan tujuan tertentu. Dengan penggunaan graf kita akan mendapatkan rute dengan keunggulan – keunggulan tertentu misalnya : lintasan dengan biaya paling murah, lintasan dengan waktu tempuh paling cepat, lintasan dengan jarak paling pendek, lintasan dengan tingkat efisiensi paling tinggi.

Adapun hal – hal yang akan kita bahas lebih lanjut dalam *paper* ini diantaranya adalah : lintasan hamilton, tujuh jembatan di konisberg, pohon merentang minimum, pohon steiner, masalah lintasan terpendek, masalah penyelidikan rute, dan permasalahan pedagang keliling.

*Kata kunci:* graf, pohon, aplikasi graf, pengrutean.

## 1. Lintasan Hamilton

Menurut teori graf pada bidang matematika, masalah lintasan hamilton dan sirkuit hamilton adalah masalah menentukan lintasan hamilton atau sirkuit hamilton yang mungkin dalam suatu graf (baik berarah maupun tidak berarah). Kedua masalah itu adalah *NP-complete*. Masalah mencari lintasan / sirkuit hamilton adalah FNP.

Ada kaitan yang erat antara 2 masalah itu. Masalah lintasan hamilton untuk graf  $G$  ekuivalen dengan sirkuit hamilton di graf  $H$  yang didapatkan dari graf  $G$  dengan menambahkan sebuah simpul baru dan menghubungkannya dengan semua simpul di graf  $G$ .

Sirkuit hamilton merupakan kasus khusus dari masalah persoalan pedagang keliling / Traveling Salesman Problem (TSP), yang diperoleh dengan menetapkan jarak tetap antara 2 buah kota jika mereka berdekatan.

Masalah sirkuit berarah ataupun tidak berarah merupakan *Karp's 21 NP-complete problems*. Pada sekitar tahun 1974 Garey dan Johnson menunjukkan dengan singkat bahwa sirkuit hamilton berarah adalah *NP-complete* untuk graf planar dan sirkuit hamilton tidak berarah untuk graf planar kubik.

### **1.1 Algoritma kuadratik untuk graf yang padat**

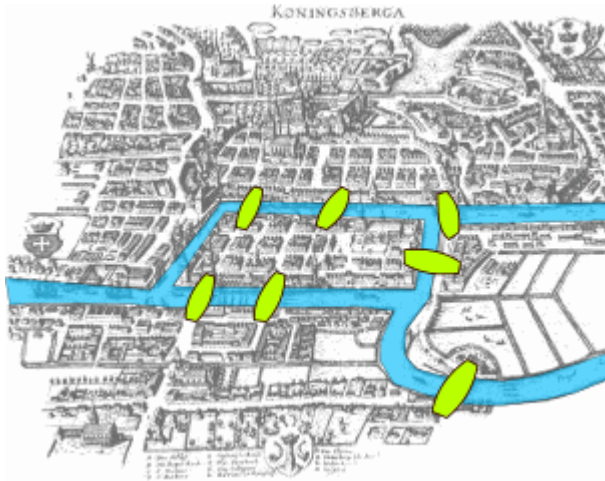
Jika derajat dari tiap – tiap simpul lebih besar atau sama dengan  $\sqrt{2}$ , maka masalah dapat diselesaikan berbanding lurus dengan fungsi kuadrat.

### **1.2 Algoritma acak**

Algoritma acak untuk menentukan lintasan hamilton pada kebanyakan graf adalah sebagai berikut : Mulai dari sembarang simpul, kemudian lanjutkan jika ada simpul tetangga yang tidak dikunjungi. Namun, jika tidak ada

lagi simpul tetangga yang berkunjung, dan lintasan yang terbentuk bukan hamilton, ambil sebuah simpul tetangga secara acak, dan putar dengan simpul tetangga itu sebagai porosnya. Lalu, lanjutkan algoritma pada lintasan terbaru.

## 2. Tujuh Jembatan di Königsberg

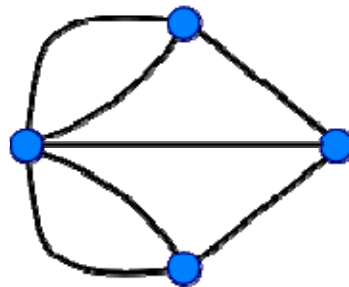
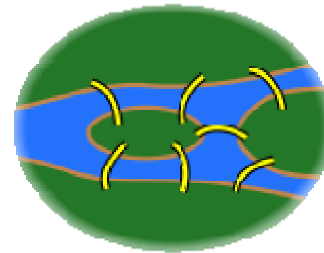
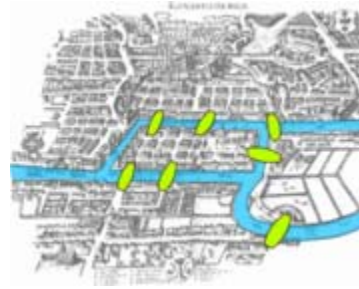


*Peta Königsberg pada masa Euler menunjukkan letak jembatan dan sungai Pregolya*

Tujuh jembatan di Königsberg adalah permasalahan matematika yang cukup terkenal yang diinspirasi oleh keadaan nyata pada masa itu. Kota Königsberg, Prussia (sekarang Kaliningrad, Russia) dibagi oleh sungai Pregel menjadi 2 daratan utama yang dihubungkan oleh 7 buah jembatan satu sama lain. Pertanyaannya adalah apakah suatu hal yang mungkin untuk berjalan melintasi setiap jembatan sebanyak satu kali dan kembali pada tempat semula. Pada 1736, Leonhard Euler menyatakan bahwa hal itu tidaklah mungkin.

### 2.1 Solusi Euler

Untuk membuktikan hasilnya, Euler memformulasikan masalah itu dalam teori graf, dengan menganalogikan pada kasus Königsberg – pertama, dengan mengeliminasi semua fitur kecuali daratan dan jembatan yang menghubungkannya; kedua, dengan mengganti daratan dengan titik, disebut simpul (vertex) dan tiap jembatan dengan garis. Disebut sisi (edge). Hasil struktur matematikanya disebut graf.



Bentuk dari graf dapat diubah-ubah dalam berbagai cara tanpa mengubah graf itu sendiri, selama sisi dari simpul tidak diubah. Tidak bermasalah antara sisi yang berbentuk lurus atau lengkung, atau simpul yang terletak di sisi kiri atau kanan sisi lainnya.

Euler menyadari bahwa masalah itu dapat diselesaikan berdasarkan derajat dari simpul – simpulnya. Derajat dari tiap simpul adalah jumlah sisi yang bersinggungan dengannya; pada graf jembatan Königsberg, tiga simpul mempunyai derajat 3 dan salah satu berderajat 5. Euler menunjukkan sirkuit yang kita inginkan hanya mungkin jika tidak ada simpul yang berderajat negatif. Cara perjalanan itu disebut sirkuit Euler. Karena graf Königsberg ini mempunyai 4 simpul yang berderajat ganjil maka graf ini tidak mempunyai sirkuit Euler.

Masalah ini dapat dikembangkan seperti berikut ini, ketika kita diminta untuk menyusun rute yang melintasi semua jembatan tapi tidak perlu punya titik mula dan akhir yang sama. Lintasan ini disebut lintasan euler. Lintasan ini dimungkinkan ada jika dan hanya jika

- 1) graf ini tidak punya simpul yang berderajat ganjil;
- 2) tepat ada 2 simpul yang berderajat ganjil, dan kedua simpul itu akan menjadi titik awal dan akhir. (ini juga hal yang mustahil untuk graf 7 jembatan Königsberg).

## 2.2 Signifikasi dalam sejarah matematika

Pada sejarah perkembangan matematika, solusi euler tentang jembatan Königsberg dianggap sebagai teorema pertama dari teori graf, yang sekarang digeneralisasi sebagai cabang ilmu kombinatorial (meskipun masalah kombinatorial telah ditemukan sebelumnya).

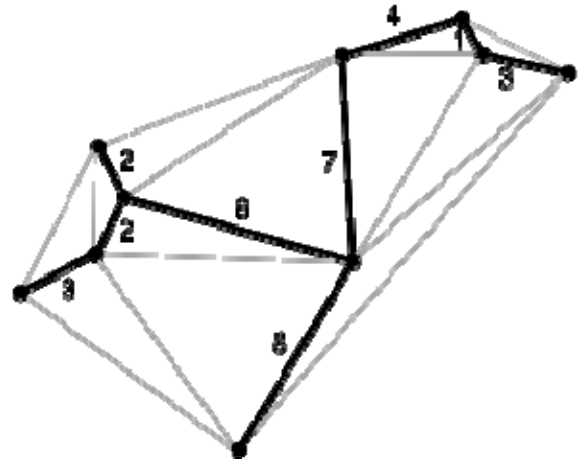
Sebagai tambahan, sesuai pengakuan Euler bahwa kuncinya terletak pada jumlah jembatan dan daftar dari titik akhirnya (lebih dari posisi aslinya) ditandai dengan perkembangan tipologi. Perbedaan antara layout nyata dan skema graf adalah contoh yang bagus dari ide bahwa topologi tidak berpatokan pada bentuk asli benda.

## 2.3 Keadaan jembatan itu sekarang

Dua dari tujuh jembatan asli hancur saat pengeboman sekutu ke Königsberg pada Perang Dunia II. Dua lainnya dirusak oleh Rusia dan diganti dengan jembatan modern. Tinggal 3 lainnya yang tersisa, meskipun hanya ada 2 yang berasal dari masa Euler (salah satu dibangun ulang oleh Jerman pada 1935).

Menurut teori graf, dua dari simpul sekarang mempunyai derajat 2, dan 2 lainnya punya derajat 3. Karena itu, sekarang lintasan euler mungkin dapat dibentuk, tapi karena harus dimulai dari sebuah pulau dan berakhir pada pulau yang berbeda, tentu bukan hal yang menyenangkan bagi turis.

## 3. Pohon Merentang Minimum



*Pohon merentang minimum dari graf planar. Tiap sisi diberi bobot, yang sebanding dengan panjangnya.*

Diberikan graf yang terhubung dan tidak berarah, pohon merentang dari graf tersebut adalah upagraf yang menghubungkan simpul – simpul itu sekaligus. Sebuah graf mempunyai berbagai macam pohon merentang yang berbeda – beda. Kita juga dapat menentukan bobot dari tiap sisi, sebagai angka yang menggambarkan seberapa mendukungnya, dan gunanya menentukan bobot untuk pohon merentang adalah untuk menghitung jumlah bobot pada pohon merentang tersebut. Pohon merentang yang minimum adalah pohon merentang yang mempunyai bobot lebih sedikit atau sama dengan pohon merentang lainnya. Secara umum, setiap graf tidak berarah punya pohon merentang minimum.

Salah satu contoh kita ambil dari perusahaan TV kabel yang menghubungkan kabel ke pengguna baru. Jika ada aturan untuk mengubur kabel sepanjang jalur tertentu, maka akan ada graf yang menggambarkan titik – titik yang terhubung oleh jalur tersebut. Beberapa lintasan ini kadang sangat mahal, karena lebih panjang, atau membutuhkan kabel untuk dikubur lebih dalam; jalur ini dapat dilambangkan dengan sisi dengan bobot yang lebih berat. Pohon merentang untuk graf ini adalah upaset dari jalur yang tidak mempunyai sirkuit tapi tetap menghubungkan setiap rumah. Akan ada beberapa pohon merentang yang mungkin terbentuk. Pohon merentang minimum merupakan jalur yang biayanya paling murah.

Pada kasus jaringan tersebut, mungkin ada beberapa kemungkinan pohon merentang minimum; Pada kasus khusus, jika semua bobotnya sama, semua pohon merentang adalah minimum. Meskipun begitu, sebuah teorema mengingatkan bahwa jika setiap sisi punya bobot yang berbeda, pohon merentang minimum yang dihasilkan adalah unik. Hal ini benar pada beberapa hal nyata. Seperti tertulis di atas, jika ditemui 2 lintasan dengan biaya yang sama. Hal ini menunjukkan keumuman hutan merentang.

Jika bobotnya tidak negatif, maka pohon merentang minimum pada kenyataannya adalah biaya minimum yang dibutuhkan yang untuk menghubungkan semua simpul, meski upagraf yang mengandung sirkuit selalu punya bobot total.

### 3.1 Algoritma

Algoritma pertama yang menemukan cara mencari pohon merentang minimum dikembangkan oleh ilmuwan Ceko-Slovakia Otakar Borůvka pada 1926 (algoritma Borůvka). Kegunaannya adalah menjamin keefisienan listrik di Bohemia. Setelah itu muncul 2 algoritma baru yang sering digunakan, algoritma prim dan algoritma kruskal. Keduanya adalah algoritma yang boros dan berjalan sesuai fungsi polinomial, jadi permasalahannya adalah mencari beberapa pohon dari  $P$ .

Algoritma yang paling cepat dalam mencari pohon merentang minimum dikembangkan oleh Bernard Chazelle, dan berbasis pada algoritma Borůvka. Waktu kerjanya  $O(e \alpha(e, v))$ , dimana  $e$  adalah jumlah dari sisi,  $v$  adalah jumlah simpul dan  $\alpha$  adalah fungsi invers klasik dari fungsi Ackermann. Fungsi  $\alpha$  tumbuh sangat lambat, sehingga untuk semua kebutuhan praktis dapat dianggap konstan dan kurang dari 4; sehingga algoritma Chazelle mendekati  $O(e)$ .

Algoritma apa yang mungkin paling cepat untuk masalah ini? Ini adalah salah satu pertanyaan umum yang sudah sangat tua dalam bidang ilmu komputer. Sangat jelas bahwa kita harus memeriksa semua bobot. Jika bobot semua sisi adalah bilangan bulat dengan jumlah bit yang dibatasi, maka waktu algoritma pemeriksaannya sebanding dengan  $O(e)$ . Untuk bobot secara umum, algoritma random dikenal berjalan linier dengan waktu yang diharapkan.

Apakah ada algoritma yang dapat bekerja dengan waktu kerja yang linier untuk graf berbobot secara umum, masih menjadi bahan diskusi. Meskipun, Seth Pettie dan Vijaya Ramachandran telah menemukan algoritma penemuan pohon merentang minimum yang cukup optimal, tingkat kompleksitas komputasinya masih belum diketahui.

Baru – baru ini, ilmuwan memfokuskan pada cara mendapatkan pohon merentang minimum pada kasus yang sering terjadi. Sebagai contoh, Hasil praktek yang ditulis pada suatu paper “Fast Shared-Memory Algorithms for Computing the Minimum Spanning Forest of Sparse Graphs” oleh David A. Bader and Guojing Cong pada 2003 menunjukkan algoritma yang dapat melakukan komputasi MSTs 5 kali lebih cepat pada 8 prosesor, dibandingkan algoritma sekuensial selama ini. Mirip dengan sebelumnya, algoritma paralel berdasarkan pada algoritma Borůvka – prim dan terutama algoritma kruskal tidak sebanding dengan prosesor yang ditambah.

Algoritma khusus lain telah didesain untuk menghitung pohon merentang minimum dari graf terlalu besar yang sebagian besar darinya harus disimpan pada disket dalam waktu lama. Algoritma penyimpanan eksternal ini, sebagai contoh dijelaskan dalam "Engineering an External Memory Minimum Spanning Tree Algorithm" oleh Roman Dementiev dan kawan – kawan, dapat beroperasi sedikitnya 2 – 5 kali lebih lambat dibandingkan tradisi dalam algoritma memori; Mereka mengklaim bahwa "massive minimum spanning tree problems filling several hard disks can be solved overnight on a PC." Mereka percaya algoritma sorting penyimpanan eksternal yang efisien dan pada pemendekan teknik graf untuk mengurangi keefisienan ukuran graf.

### 3.2 Pohon merentang minimum pada graf lengkap

Hal ini telah ditunjukkan oleh J. Michael Steele berdasar hasil kerja A.M. Frieze yang diberikan graf lengkap dengan  $n$  simpul, dengan bobot simpul dipilih dari distribusi acak kontinu  $f$  sehingga  $f'(0) > 0$ , sebagai pendekatan yang tidak terbatas ukuran dari pendekatan MST  $\zeta(3) / f'(0)$ , dimana  $\zeta$  adalah fungsi Riemann zeta.

Untuk keseragaman bobot acak dalam  $[0,1]$ , Ukuran sebenarnya dari pohon merentang

minimum yang diharapkan telah dihitung dalam graf lengkap ukuran kecil.

Simpul	Ukuran yang diharapkan
2	1 / 2
3	3 / 4
4	31 / 35
5	893 / 924
6	278 / 273
7	30739 / 29172
8	199462271 / 184848378
9	126510063932 / 115228853025

### 3.3 Masalah yang terkait

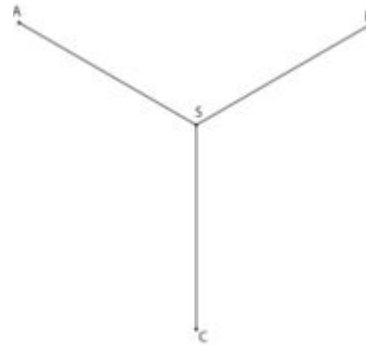
Graf yang terkait dengan k-pohon merentang minimum adalah pohon yang terbagi menjadi beberapa upaset dari k simpul dengan graf berbobot minimum.

Sebuah set dari k- pohon merentang paling kecil adalah upaset dari k-pohon merentang (tidak mungkin semua pohon merentang) seperti tidak ada pohon merentang diluar upaset yang punya bobot lebih kecil. (catatan : masalah ini tidak berkaitan dengan k- pohon merentang minimum).

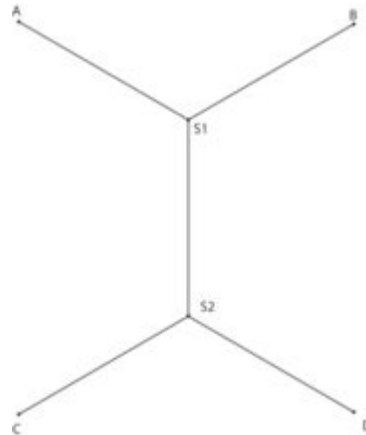
Masalah geometri yang terkait adalah Pohon merentang minimum euclidean.

Pada model sistem terdistribusi, dimana setiap simpul dianggap sebuah komputer dan tidak ada simpul dikenal semua pengecualian pada tiap link penghubungnya, salah satu dianggap sebagai pohon merentang minimum terdistribusi. Definisi masalah ini dalam matematika adalah sama tapi punya pendekatan yang berbeda dalam solusi.

### 4. Pohon Steiner



*Solusi untuk 3 titik; Titik steiner ada di tengah – ingat bahwa tidak ada koneksi langsung antara A,B, dan C*



*Solusi untuk 4 titik – ingat bahwa ada dua titik steiner, S<sub>1</sub> dan S<sub>2</sub>*

Masalah pohon steiner adalah masalah dalam optimalisasi kombinatorial.

Masalah pohon steiner sama dangkalnya dengan masalah pohon merentang minimum : diberikan set dari V titik (simpul), saling dihubungkan dengan jaringan (graf) dengan panjang terpendek. Perbedaan antara masalah pohon steiner dengan pohon merentang minimum adalah pada pohon steiner tambahan penghubung simpul dan sisi dapat ditambahkan pada graf untuk mengurangi panjang dari pohon merentang. Simpul – simpul baru ini dimunculkan untuk mengurangi jumlah panjang dari koneksi yang dikenal sebagai titik steiner atau simpul steiner. Sebagai tambahan, simpul di V tidak boleh mempunyai koneksi langsung antar sesama. Ini membuktikan bahwa hasil koneksi adalah pohon, dikenal sebagai pohon

steiner. Mungkin ada beberapa pohon steiner dalam setiap simpul – simpul awal yang diberikan.

Masalah sebenarnya dinyatakan dalam form yang kemudian menjadi dikenal sebagai masalah pohon steiner euclidean: Diberi  $N$  titik dalam bidang datar, diminta untuk menghubungkan mereka dengan garis yang punya jumlah panjang minimal seperti jalan anatar dua titik yang dapat saling terhubung oleh segmen garis berarah keduanya atau lewat titik lain dan segmen garis.

Untuk masalah steiner euclidean, titik ditambahkan pada graf (titik steiner) harus punya derajat 3, dan 3 sisi berkaitan seperti sebuah titik harus membentuk 3 buah sudut  $120^\circ$ . Hal ini diikuti bahwa jumlah maksimum dari titik steiner bahwa pohon steiner punya  $N-2$ , dimana  $N$  adalah jumlah awal titik yang diberikan.

Lebih lanjut dapat digeneralisasi ke ke metrik masalah pohon steiner. Diberikan graf berbobot  $G(S, E, w)$  yang simpulnya berkorespondensi ke titik di ruang metrik, dengan sisi berbobot yang dianggap sebagai jarak dalam ruang, dibutuhkan untuk mencari pohon dengan jumlah panjang minimum yang simpul – simpulnya adalah supersset dari set  $S$  dari simpul – simpul pada  $G$ .

Versi paling umum adalah pohon steiner dalam graf. Diberikan graf berbobot  $G(V, E, w)$  dan sebuah subset simpul – simpul  $S \subseteq V$  mencari pohon dengan bobot minimum yang menyertakan semua simpul – simpul dari  $S$ .

Masalah metrik pohon steiner berkaitan dengan pohon steiner dalam masalah graf dimana graf mempunyai jumlah simpul yang tidak terbatas, dimana semua titik adalah ruang metrik.

Masalah pohon steiner punya aplikasi dalam layout sirkuit dan desain jaringan. Kebanyakan versi dari pohon steiner NP-lengkap. Contohnya perhitungan yang rumit. Pada kenyataannya, salah satu darinya termasuk dalam masalah *Karp's original 21 NP-complete*. Beberapa kasus tertentu dapat diselesaikan dalam waktu polinomial. Saat latihan, kita perlu menyelidiki sendiri.

Salah satu penaksiran yang biasa digunakan untuk masalah pohon steiner euclidean adalah

untuk menghitung pohon merentang minimum euclidean.

## 5. Masalah Lintasan Terpendek

Pada teori graf masalah lintasan terpendek dengan satu sumber adalah masalah dalam menemukan lintasan antara simpul – simpul yang jumlah bobotnya dengan sisi yang dipilihnya minimal. Lebih formal lagi, diberikan graf berbobot (yaitu, set dari simpul – simpul  $V$ , set dari sisi – sisi, dan nilai bobot dalam real  $f : E \rightarrow \mathbb{R}$ ), dan selanjutnya diberi sebuah elemen  $v$  dari  $V$ , cari lintasan  $P$  dari  $v$  ke  $v'$  dari  $C$  sehingga,

$$\sum_{p \in P} f(p)$$

adalah minimal diantara semua lintasan yang menghubungkan  $v$  ke  $v'$ . Semua pasangan masalah lintasan terpendek, dimana kita harus mencari lintasan antara  $v$  ke  $v'$ .

Solusinya dikenal dengan masalah lintasan terpendek dan kadang dikenal dengan algoritma *pathing*. Algoritma penyelesaian paling penting adalah :

- Algoritma Dijkstra's — menyelesaikan sebuah permasalahan jika bobot semua sisi adalah lebih besar atau sama dengan 0. Tanpa menjelekkan waktu eksekusinya, pada kenyataannya algoritma ini melakukan komputasi lintasan terpendek dari titik awal ke semua simpul lainnya.
- Algoritma Bellman-Ford — menyelesaikan sebuah permasalahan jika bobot sisinya boleh negatif.
- Algoritma A\* search menyelesaikan sebuah permasalahan lintasan terpendek menggunakan penyelidikan sendiri untuk mempercepat pencarian.
- Algoritma Floyd-Warshall — menyelesaikan semua pasangan lintasan terpendek.
- Algoritma Johnson — Menyelesaikan semua pasangan lintasan terpendek, dapat delibeh cepat daripada Floyd-Warshall pada grah yang sembarang.
- Teori Perturbation ; mencari (paling jelek) lintasan terpendek lokal.

Masalah yang terkait adalah masalah pedagang keliling, dimana masalahnya adalah untuk mencari lintasan terpendek yang melewati setiap simpul tepat sekali, dan kembali pada titik awal. Masalah itu adalah *NP-complete*, sehingga solusi yang efisien tampaknya belum ada.

Pada bidang *networking* dan telekomunikasi, masalah lintasan terpendek ini sering disebut *the min-delay path problem* dan biasanya tergantung pada luas masalah lintasan. Seperti : *Shortest (min-delay) widest path or Widest shortest (min-delay) path*.

Aplikasi lain adalah permainan "six degrees of separation" yang mencoba untuk mencari lintasan terpendek dalam graf seperti seorang artis yang muncul dalam film yang sama.

### 6. Masalah penyelidikan rute

Pada teori graf, cabang dari matematika, persoalan tukang pos Cina (the Chinese postman problem) atau penyelidikan rute yang mencari lintasan (sirkuit) terpendek yang melewati semua sisi (terhubung) pada graf tidak berarah. Saat graf merupakan sirkuit euler maka, sirkuit adalah solusi paling optimal.

Alan Goldman dari NIST pertama menemukan nama 'Chinese Postman Problem' untuk masalah ini, seperti penelitian awal yang dilakukan matematikawan Cina Mei Ko Kuan pada 1962.

#### 6.1 Lintasan dan sirkuit euler

Dengan maksud agar graf mempunyai kemiripan dengan lintasan euler, sudah seharusnya semua terhubung.

Sehingga andai kita punya graf terhubung  $G = (V,E)$ , pernyataan berikut ini adalah ekuivalen :

1. Semua simpul – simpul di  $G$  punya derajat genap.
  2.  $G$  terdiri dari sisi yang tidak menyatu pada satu sirkuit, dan simpul –simpul bukan sirkuit.
  3.  $G$  punya lintasan euler.
1. → 2. dapat ditunjukkan dengan induksi ke jumlah dari sirkuit.
2. → 3. dapat ditunjukkan dengan induksi pada jumlah sirkuitnya, dan

3. → 1. dapat secara langsung.

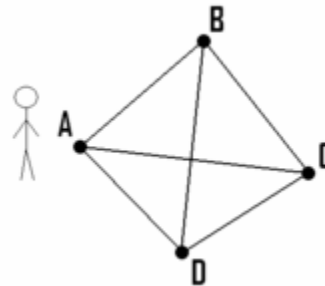
Sebuah lintasan semi-euler (lintasan yang tidak tertutup tapi melewati semua sisi pada  $G$  sekali) ada jika dan hanya jika  $G$  terhubung dan tepat punya 2 simpul – simpul yang tidak berderajat genap.

### 6.2 Solusi

Jika graf adalah graf euler, dan lintasan eulernya melewati semua sisi, sehingga solusinya adalah untuk memilih lintasan euler.

Jika grafnya bukan euler, maka graf itu pasti mengandung simpul – simpul dengan derajat ganjil. Berdasar hasil pada teori graf, akan ada jumlah genap dari tipe simpul – simpul. Ingat bahwa kita harus memeriksa semua sisi – sisi yang berasal dari simpul – simpul ini sebagai solusi. Kita dapat membuat graf euler dengan menggandakan lintasan yang menghubungkan simpul – simpul itu dalam pasangan. Kita memilih pasangan – pasangan seperti total semua jarak lintasan yang menghubungkan simpul – simpul sekecil mungkin. Sekarang solusinya adalah lintasan euler pada graf baru.

### 7. Persoalan Pedagang Keliling (Travelling salesman problem/ TSP)



Jika seorang salesman memulai pada titik A, dan jika jarak antar setiap pasang titik diketahui, apa jarak terpendek rute yang mengunjungi semua titik dan kembali ke titik A?

Permasalahan pedagang keliling (The traveling salesman problem / TSP) adalah sebuah masalah yg mempunyai ciri – ciri tersendiri atau gabungan keoptimisan.

## 7.1 Pengungkapan masalah

Diberikan sejumlah kota dan biaya perjalanan dari suatu kota ke kota lain, apa rute perjalanan termurah yg mengunjungi masing – masing kota satu persatu dan kemudian kembali ke kota awal?

Sebuah formula ekuivalen pada istilah teori graf adalah : dengan sebuah graf berat yang komplit (dimana puncak akan menggambarkan kota, tebing akan menggambarkan jalan,dan bobot akan menjadi biaya/jarak dari jalan tsb),menemukan sebuah putaran Hamiltonian dengan bobot paling sedikit.

Ini dapat ditunjukkan dengan syarat kembali ke kota awal tidak mengubah kerumitan perhitungan dari masalah itu..

Sebuah masalah yang berhubungan adalah kemacetan TSP( *bottleneck* TSP ) : menemukan sebuah putaran Hamiltoni pd graf berbobot dengan panjang minimum dari tebing terpanjang.

Masalahnya adalah praktis sangat penting, terpisah dari area transportasi dan logistik yang jelas. Sebuah contoh klasik ada pada *printed circuit manufacturing* : mengatur sebuah rute pd mesin bor untuk mengebor lubang pada suatu PCB. Pada permesinan robot/aplikasi bor, "kota" adalah bagian dari mesin/lubang (pd ukuran yg berbeda) untuk mengebor, dan "biaya perjalanan" termasuk waktu untuk melengkapi kembali robot tsb (masalah rangkaian tunggal kerja mesin).

## 7.2 Sejarah dari masalah

Masalah matematika yang berhubungan dengan masalah TSP telah dibicarakan pada sekitar tahun 1800an oleh pakar matematika Irlandia Sir William Rowan Hamilton dan oleh pakar matematika Inggris Thomas Penyngton Kirkman. Sebuah diskusi menyenangkan tentang kerja awal dari hamilton dan kirkman dapat ditemukan di buku Graph Theory 1736-1936 oleh N.L. Biggs, E.K. Lloyd, dan R.J. Wilson, Clarendon Press, Oxford 1976.

Bentuk umum dari kemunculan TSP pertama kali telah dipelajari oleh para pakar matematika dimulai pada tahun 1930an oleh Karl Menger di Vienna dan Harvard. Masalah itu terakhir

diperkenalkan oleh Hassler Whitney dan Merrill Flood di Princeton. Sebuah perlakuan lengkap pada hubungan diantara Menger dan Whitney, dan pertumbuhan TSP sebagai suatu topik belajar dapat ditemukan di dokumen Alexander Schrijver's "On the History of Combinatorial Optimization (smp 1960)".

## 7.3 Kompleksitas komputasi

Solusi paling baik adalah mencoba seluruh permutasi (gabungan permintaan) dan melihat mana yang paling murah (menggunakan pencarian paksa yang kasar), tapi angka permutasinya adalah  $n!$  (faktorial dari jumlah kota,  $n$ ), solusi ini dengan cepat menjadi tidak praktis.

Menggunakan teknik program dinamis dapat memecahkan masalah tepatnya dalam waktu  $O(2^n)$ . Walaupun ini adalah eksponen aljabar, ini jauh lebih baik dari  $O(n!)$ . (Lihat notasi Big O).

## 7.4 NP-hardness

Masalah ini telah ditunjukkan menjadi *NP-hard* dan versi keputusan masalah ("dengan biaya dan sebuah angka  $x$ , memutuskan bahwa ada sebuah rute perputaran lebih murah daripada  $x$ ") adalah *NP-complete*.

Kemacetan TSP termasuk juga NP-hard.

Masalah tetap NP-hard bahkan untuk kasus ketika kota ada di dalam suatu pesawat dengan jarak Euclidean, dan juga pada sejumlah kasus yang membatasi lainnya.

Memindahkan kondisi dari kunjungan masing-masing kota "hanya sekali" tidak memindah *NP-hardness*, sejak itu dengan mudah dapat dilihat bahwa pada kasus planar sebuah perjalanan kunjungan kota yg optimal hanya sekali (sebaliknya, dengan segitiga ketidaksamaan, sebuah jalan pintas yang melewati sebuah kunjungan pengulangan akan mengurangi panjang perjalanan).

## 7.5 Algoritma

Garis tradisional pemecahan untuk masalah *NP-hard* adalah sebagai berikut:



- memikirkan algoritma untuk menemukan solusi tepat (mereka akan bekerja dengan alasan cepat hanya untuk ukuran masalah yang secara relatif kecil).
- memikirkan “suboptimal” atau algoritma heuristik, yaitu algoritma yang mungkin merupakan solusi yang baik, tapi yang tidak mungkin dibuktikan secara optimal.
- menemukan kasus spesial untuk masalah (“subproblems”) yang mungkin lebih baik atau tepatnya mungkin dilakukan.

Untuk menentukan TSP algoritma, TSPLIB suatu perpustakaan contoh instan TSP dan masalah yang berhubungan adalah dipertahankan, lihat TSPLIB referensi eksternal. Banyak dari mereka adalah daftar dari kota sebenarnya dan rancangan cetakan perjalanan keliling sebenarnya.

#### 7.5.1 Algoritma yang tepat

- Berbagai cabang dan loncatan algoritma, yang dapat digunakan untuk memproses TSP, berisi 40-60 kota.
- Kemajuan progresif algoritma yang menggunakan teknik yang mengingatkan kepada program linear. Bekerja dengan baik untuk sampai dengan 120-200 kota.
- Implementasi baru-baru ini dari cabang-cabang loncatan dan potongan berdasarkan pada program linear bekerja sangat baik sampai dengan 5000 kota, dan pendekatan ini telah digunakan untuk memecahkan dalam hal ini sampai dengan 33810 kota.

Sebuah solusi tepat untuk 15112 kota-kota Jerman dari TSPLIB ditemukan tahun 2001 menggunakan metode cutting-plane diajukan oleh George Dantzig, Ray Fulkerson, dan Selmer Johnson tahun 1954, berdasarkan pada program linear. Perhitungan ditunjukkan pada sebuah jaringan kerja dari 110 processor yang terletak di Universitas Rice dan Universitas Princeton. Perhitungan total waktu sama dengan 22,6 tahun pada 500 MHz processor Alpha tunggal. Pada Mei 2004, masalah perjuangan salesman yang mengunjungi semua 24978 kota di Swedia telah dipecahkan: sebuah perjalanan dengan perkiraan

panjang 72500 kilometer ditemukan dan itu telah dibuktikan tidak ada yang lebih pendek.

Pada Maret 2005, kunjungan TSP ke 33,810 titik pada suatu circuit board telah dipecahkan menggunakan *CONCORDE* : sebuah perjalanan sepanjang 66,048,945 unit telah ditemukan dan ini telah dibuktikan bahwa tidak ada perjalanan terpendek, perhitungan diperkirakan mengambil perkiraan 15.7 CPU tahun.

#### **7.5.2 Heuristik**

Banyak perkiraan algoritma, yang dengan cepat menghasilkan solusi yang baik dengan kemungkinan tinggi, telah dipikirkan. Metode modern dapat menemukan solusi luar biasa untuk masalah besar (jutaan kota) dengan waktu yang masuk akal dengan suatu kemungkinan tinggi hanya 2-3 % dari solusi optimal.

Beberapa kategori heuristics yang dikenali :

##### 7.5.2.1 Heuristik membangun

- Tetangga terdekat algoritma, yang secara normal hampir mendekati rute optimal, dan tidak memakan waktu lama untuk melaksanakan. Rosenkrantz dkk [1977] menunjukkan bahwa tetangga terdekat algoritma memiliki faktor perkiraan  $\Theta(\log |V|)$ . Sayangnya, ini dapat dipercaya hanya untuk kasus khusus TSP, yaitu dimana segitiga sembarang dipuaskan. Pada kasus umum, ada sebuah contoh dimana algoritma tetangga terdekat memberikan kemungkinan rute paling buruk. Pada praktisnya heuristik menyediakan solusi dengan rata-rata 10 sampai 15 persen diatas optimal. Sebuah contoh dari tetangga terdekat algoritma menemukan kasus path paling buruk yang akan melewati sebuah garis, dimulai dari 0, ke alternatif negatif dan kekuatan positif dari 2 minus 1, dan juga, dari 0,1,-3,7,-15,31,-63...  $(2^n - 1)(-1)^n$ .

##### 7.5.2.2 Perkembangan iteratif

- *Pairwise exchange*, or Kernighan-Lin heuristik.

- k-opt heuristik : Ambil sebuah perjalanan yang diberikan dan hapus satu sama lain tebing yang saling terpotong-potong. Mengumpulkan sisa fragment pada sebuah perjalanan, meninggalkan subrute yang tidak terpotong-potong (jangan menghubungkan sebuah titik akhir fragmen bersama). Efek ini mempermudah TSP untuk dipertimbangkan ke masalah yang lebih simpel. Masing-masing titik akhir fragmen dapat dihubungkan ke  $2k - 2$  kemungkinan lain: dari  $2k$  total titik akhir fragmen yang tersedia, dua titik akhir fragmen dibawah pertimbangannya adalah tidak diijinkan. Seperti ketidakleluasaan  $2k$  kota TSP dapat kemudian dipecahkan dengan metode paksa kasar untuk menemukan gabungan biaya terendah dari fragmen orisinal.

### 7.5.2.3 Pengembangan acak

- Optimalisasi rangkaian algoritma Markov yang menggunakan pencarian heuristik lokal upa-algoritma dapat menemukan sebuah rute yang benar-benar tertutup sampai rute optimal untuk 700-800 kota.
- Lintasan acak mengubah algoritma *state-of-the-art search* dan bekerja sampai dengan 100.000 kota. Konsepnya sangat simpel : Memilih sebuah lintasan acak, memilih empat titik terdekat, tukar jalan mereka untuk menciptakan sebuah lintasan acak baru, dimana secara paralel bertambahnya loncatan paling atas dari suatu panjang lintasan. Jika pengulangan sampai suatu angka percobaan dari lintasan acak mengubah kegagalan menjadi loncatan paling atas, satu telah menemukan suatu minimum lokal dengan kemungkinan tertinggi, dan bahkan walau sebuah global minimum dengan kemungkinan tertinggi (dimana tinggi berarti bahwa sisa kemungkinan berkurang secara eksponen pada masalah ukuran  $n$  itu untuk 10.000 atau jenis simpul, kesempatan untuk gagal dapat diabaikan).

TSP adalah sebuah batu sentuh untuk banyak heuristik umum memikirkan untuk gabungan optimalisasi: *genetic algorithms*, *simulated annealing*, *Tabu search*, *ant system*.

### **7.5.3 Kasus khusus**

#### 7.5.3.1 Segitiga sembarang dan algoritma christofides

Sebuah larangan pembatasan TSP natural adalah segitiga sembarang. Untuk 3 kota A, B dan C, jarak diantara A dan C harus paling tidak jarak dari A ke B ditambah jarak dari B ke C. Kebanyakan instan alami memuaskan keterbatasan TSP .

Pada kasus ini, ada sebuah *constant-factor approximation algorithm* (berdasar Christofides, 1975) yang selalu menemukan sebuah perjalanan panjang paling tidak 1.5 waktu perjalanan paling pendek. Pada paragraf selanjutnya, kami menjelaskan suatu yg lemah (tapi lbh mudah) algoritma dimana menemukan sebuah perjalanan panjang paling tidak 2 kali dari yg terpendek.

Panjang dari pohon merentang minimum dari suatu jaringan adalah sebuah loncatan alami terendah dari panjang rute optimal. Pada TSP dengan kasus *triangle inequality* ini dimungkinkan untuk membuktikan loncatan tertinggi pada tema pohon merentang minimum dan mendesain sebuah algoritma yang telah terbukti memiliki loncatan tertinggi dari panjang rute. Contoh yang pertama kali dipublikasikan (dan yang termudah) mengikuti.

- Langkah 1: Mendirikan pohon merentang minimum.
- Langkah 2: Menggandakan seluruh tebingnya. Dimana ada suatu tebing dari  $u$  ke  $v$ , tambahkan tebing kedua dari  $u$  ke  $v$ . Yang memberi graf euler.
- Langkah 3: Temukan sebuah sirkuit euler disitu. Panjangnya adalah 2 kali dari panjang pohon.
- Langkah 4: Merubah sirkuit euler menjadi hamilton dengan cara sbb: berjalan sepanjang sirkuit Euler, dan setiap waktu kamu akan mendatangi suatu puncak yg siap dikunjungi, loncati itu dan coba untuk pergi ke yang satu lagi (sepanjang sirkuit euler).

Hal ini sangat mudah dibuktikan bahwa langkah terakhir bekerja. Lebih banyak lagi, terima kasih kepada triangle inequality, setiap loncatan pada langkah 4 adalah fakta suatu jalan pintas, yaitu panjang dari cycle tidak bertambah. Sebab itu memberi kita perjalanan TSP tidak lebih dari 2 kali selama optimal lainnya.

Algoritma christofides mengikuti suatu outline yang mirip tetapi menggabungkan pohon merentang minimum dengan suatu solusi masalah lainnya, berat minimum dengan pasangan ideal. Memberikan suatu perjalanan TSP yang 1,5 kali lebih optimal. Ini sudah berjalan lama (sejak 1975) membuka masalah untuk meningkatkan 1.5 ke konstan yg lebih kecil. Ini diketahui, bagaimanapun, bahwa tidak ada waktu eksekusi polinomial yg menemukan suatu perjalanan dari panjang plg tidak  $1/219$  lebih dari optimal, paling tidak  $P = NP$  (Papadimitriou and Vempala, 2000). Pada kasus *bounded metrics* diketahui bahwa tidak ada waktu eksekusi polinomial yang membangun suatu perjalanan paling tidak  $1/388$  lebih dari optimal, paling tidak  $P = NP$  (Engebretsen and Karpinski, 2001). Perkiraan waktu algoritma polinomial paling baik untuk masalah TSP dengan jarak satu dan dua menemukan suatu panjang perjalanan plg tidak  $1/7$  lebih dari optimal (Berman and Karpinski, 2006).

Algoritma christofides adalah salah satu algoritma perkiraan, dan sebuah bagian tanggung jawab untuk menggambar perhatian ke perkiraan algoritma sebagai suatu latihan untuk kekuatan masalah. Sebagai satu kenyataan masa algoritma tidak secara umum diperpanjang ke algoritma perkiraan sampai sesudahnya. Pada waktu publikasi, algoritma christofides direferensikan sebagai heuristik christofides.

### 7.5.3.2 TSP Euclidean

TSP Euclidean atau TSP planar, adalah TSP dengan jarak yang umumnya adalah jarak euclidean. Meskipun semua masalah merupakan *NP-hard*, dapat dikenali sebagai adanya upa eksponensial waktu algoritma untuk menjalankannya. Lebih dari itu, mencari sendiri akan lebih baik.

TSP euclidean adalah masalah khusus dari TSP dengan segitiga sembarang, karena jarak pada bidang datar mematuhi segitiga sembarang. Bagaimanapun, terlihat lebih mudah dalam

dibanding TSP pada umumnya dengan segitiga sembarang. Sebagai contoh, pohon merentang minimum dari graf yang berasosiasi dengan contoh dari TSP euclidean adalah pohon merentang minimum euclidean, dan sehingga dapat melakukan komputasi yang diharapkan  $O(n \log n)$  untuk  $n$  titik (anggap lebih kecil dari jumlah sisi). Hal ini membuat 2 perkiraan simpel algoritma untuk TSP dengan segitiga sembarang di atas dapat dioperasikan dengan lebih cepat.

Secara umum, untuk setiap  $c > 0$ , ada waktu algoritma polinomial untuk mencari panjang rute yang hampir  $(1 + 1/c)$  kali dari contoh geometrik untuk TSP (aurora); yang disebut *polynomial-time approximation scheme*. Hasil ini penting dalam algoritma teoritis tapi tidak sama dengan prakteknya. Sebagai pengganti, bekerja sendiri dengan jaminan yang lemah kadang sering digunakan, meski mereka juga menunjukkan hasil lebih baik sebagai contoh TSP euclidean daripada contoh umum lainnya

### 7.5.3.3 TSP Asimetris

Pada kebanyakan kasus, jarak antara 2 simpul dalam jaringan TSP adalah sama dalam 2 arah. Kasus dimana jarak antara A ke B tidak sama dengan jarak antara B ke A dinamakan TSP asimetris. Aplikasi praktis dari TSP asimetris adalah pengoptimalan rute menggunakan level perutean jalan (asimetris terjadi karena jalan satu arah, jalan licin dan jalan kendaraan bermotor).

### **Penyelesaian dengan mengonversi ke TSP simetris**

Menyelesaikan TSP tidak simetris dapat sangat kompleks. Berikut ini adalah matriks  $3 \times 3$  mengandung semua lintasan berbobot yang mungkin antara simpul A, B, dan C. Salah satu opsi adalah dengan mengubah matriks simetris berukuran  $N$  menjadi matriks simetris berukuran  $2N$ , mengadakan kompleksitasnya.

Bobot lintasan asimetris

	A	B	C
A		1	2
B	6		3
C	5	4	

Untuk menggandakan ukurannya, setiap simpul dari graf digandakan, membuat simpul semu

kedua. Menggunakan titik – titik semu dengan bobot yang sangat kecil, seperti  $-\infty$ , menyediakan rute “linking” kembali ke simpul sebenarnya yang murah dan membuat evaluasi simetrik dapat dilanjutkan. Matriks asli  $3 \times 3$  ditunjukkan seperti di atas adalah dapat dilihat di kiri bawah dan invers dari originalnya pada kanan atas. Kedua kopian dari matriks punya diagonalnya menggantikan lintasan dengan biaya kecil, ditunjukkan dengan  $-\infty$ .

Bobot lintasan simetris

	A	B	C	A'	B'	C'
A				$-\infty$	6	5
B				1	$-\infty$	4
C				2	3	$-\infty$
A'	$-\infty$				1	2
B'	6	$-\infty$				3
C'	5	4	$-\infty$			

Matriks  $3 \times 3$  asli akan memproduksi 2 sirkuit hamilton (lintasan yang melewati setiap simpul sekali), dinamakan A-B-C-A (nilai = 12) dan A-C-B-A (nilai = 9). Mengevaluasi versi simetris  $6 \times 6$  dari masalah yang sama memunculkan beberapa lintasan seperti A-A'-B-B'-C-C'-A, A-B'-C-A'-A, A-A'-B-C'-A [semua bernilai  $9-\infty$ ].

Hal penting dari setiap langkah bahwa ada alternatif antara yang dibuang (A',B',C') dan yang tidak akan dibuang (A,B,C) dan bahwa linkup untuk “jump” antara pasangan yang berelasi (A-A') adalah kebebasan yang efektif. Versi algoritma ini dapat digunakan untuk semua bobot pada lintasan A-A', selama bobotnya adalah lebih kecil dibanding lintasan lainnya. Sebagai lintasan berbobot untuk “jump” harus lebih efektif untuk “free”, nilai nol (0) dapat digunakan untuk merepresentasi biayanya – jika nol tidak digunakan untuk kebutuhan yang lain (seperti untuk menyatakan lintasan yang tidak valid). Pada 2 contoh di atas, lintasan yang tidak eksis antara simpul –simpul ditunjukkan dengan kotak yang kosong.

## 7.6 Kegiatan Manusia pada TSP

TSP, variasi khusus dari euclidean, telah menarik perhatian dari peneliti pada psikologis kognitif. Hal ini diamati dari manusia yang dapat memproduksi solusi yang berkualitas dengan sedikit usaha kognitif dan menggunakan waktu yang relatif sedikit. Pengamatan ini kontras dengan kompleksitas komputasi yang jelas pada TSP.

Dua hipotesis umum yang dirumuskan untuk menjelaskan kesuksesan manusia dalam menyelesaikan TSP :

- Hipotesis 1. Tugas untuk mencari rute terpendek yang menghubungkan set dari titik pada peta datar euclidean ke arti natural dari sistem penglihatan manusia untuk mengorganisasir informasi yang dapat ditangkap menurut prinsip Gestalt tentang kesederhanaan, kiasan yang baik atau Prägnanz (MacGregor & Ormerod, 1996) (lihat juga *perception, Gestalt psychology*). Hipotesis ini didukung oleh kemudahan dan kecepatan dengan siapa manusia membuat rute dalam tugas TSP.
- Hipotesis 2. Tindakan manusia dalam TSP menunjukkan pokok intelegensia dan keadaptifan dari kehidupan manusia. Hipotesis ini didukung oleh penemuan bahwa tindakan TSP manusia berkaitan dengan ukuran dari kecerdasan global/ IQ (Vickers dkk.).

Penyebaran pertama dari *Journal of Problem Solving* setia kepada topik tindakan manusia dalam TSP.

## **8. Kesimpulan**

Graf adalah salah satu cabang ilmu dalam matematika yang mempunyai peruntukan yang cukup luas dalam kehidupan nyata. Diantaranya adalah penggunaan graf dalam melakukan perutean. Perutean yaitu kegiatan membuat rute dengan tujuan tertentu. Dengan penggunaan graf kita akan mendapatkan lintasan dengan keunggulan – keunggulan tertentu misalnya :

1. lintasan dengan biaya paling murah.
2. lintasan dengan waktu tempuh paling cepat.
3. lintasan dengan jarak paling pendek.
4. lintasan dengan tingkat efisiensi paling tinggi.

Pada bidang yang lebih luas, jika kita menganalogikan lintasan yang kita buat dengan alur kerja yang harus kita lakukan , maka akan kita dapatkan efisiensi kerja dan hasil yang optimal.

## DAFTAR PUSTAKA

- [1] <http://wikipedia.org>. Tanggal akses : 30 Desember 2006 pukul 19:51.
  
- [2] Munir, Rinaldi. (2006). Bahan Kuliah IF2153 Matematika Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.
  
- [3] *NIST*. (2004). National Institute of Standards and Technology. <http://www.nist.gov>. Tanggal akses: 30 Desember 2006 pukul 23:15.