# ALGORITMA RUTE TERPENDEK BERBASIS TEORI *GRAPH*

PRAPTO TRI SUPRIYO

Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor
Jl Meranti, Kampus IPB Darmaga, Bogor 16680 Indonesia

**ABSTRAK.** Masalah penentuan rute optimal armada kendaraan pada suatu jaringan jalan (*network*) seringkali dijumpai dalam perencanaan dan pengoperasian sistem pelayanan yang dilakukan oleh suatu instansi atau badan usaha tertentu. Hal ini memiliki beberapa masalah khusus. Satu diantara masalah tersebut adalah penentuan rute terpendek kendaraan tunggal yang meliputi semua jalan (*edge*) pada suatu *network*. Masalah penentuan rute terpendek kendaraan tunggal yang meliputi semua jalan pada suatu *network* tanpa memandang adanya urutan jalan yang harus dilalui, diperlihatkan analog dengan masalah menentukan himpunan *edge–edge* dengan panjang minimum yang ditambahkan pada *network* asli untuk memperoleh *network* diperbesar guna menyeimbangkan derajat masuk dan derajat keluar pada setiap verteks. Rute terpendek yang diperoleh akan berupa suatu *circuit* Euler pada *network* diperbesar.

**Katakunci:** *graph*, *network*, *circuit* Euler.

## 1. PENDAHULUAN

Banyak badan usaha, baik milik negara maupun swasta dihadapkan pada suatu masalah menentukan rute kendaraannya guna menyusuri hampir semua jalan dalam suatu kota, untuk suatu keperluan atau pelayanan tertentu, yang biasanya dilakukan secara periodik. Sebagai contoh pembersihan jalan dan pengambilan/pengangkutan sampah oleh Dinas Kebersihan Kota, pemeriksaan kabel listrik atau telpon dari suatu gangguan, dan patroli dalam kota yang dilakukan oleh polisi. Dalam masalah ini kita ingin menentukan suatu rute yang efisien guna melakukan tugas tersebut sedemikian sehingga setiap jalan yang harus dilewati, masing-masing dilewati sebanyak seminimal mungkin. Hal ini berkaitan dengan usaha meminimalkan biaya operasional yang harus dikeluarkan untuk melakukan tugas tersebut. Suatu rute dapat dipandang sebagai barisan dari lokasi-lokasi dimana suatu kendaraan harus mendatangi tempat tersebut untuk memberikan pelayanan [3].

Tulisan ini memberikan suatu model matematik yang berkaitan dengan penyelesaian masalah di atas, yakni penentuan rute kendaraan tunggal, yang meliputi semua jalan (*edge*) pada suatu jaringan jalan (*network*) yang sekaligus meminimumkan jarak total yang dilalui, tanpa kendala urutan *edge* yang harus dilalui.

## 2. ANALISA MASALAH

Jaringan jalan dalam suatu kota dapat dinyatakan dengan suatu *graph* berarah (*directed graph*) atau seringkali juga disebut sebagai *network.* Suatu *graph* berarah $G = (V,E)$ terdiri dari suatu himpunan $V$ yang anggotanya disebut verteks (*vertex)* dan himpunan $E$ yang anggotanya disebut *edge*, yang masing-masing berarah dari suatu verteks ke verteks yang lain. Referensi [5] membicarakan teori *graph* dan penerapannya. Suatu *edge* menyatakan suatu ruas jalan, sedangkan verteks menyatakan titik pertemuan beberapa ruas jalan seperti misalnya pertigaan atau perempatan. Jika suatu jalan diberlakukan dua arah, maka dinyatakan dalam dua *edge* dengan arah yang berlawanan. Selanjutnya didefinisikan suatu bobot pada masing-masing *edge* yang memberikan panjang jalan yang dinyatakan oleh *edge* tersebut. Selanjutnya kita ingin melakukan perjalanan, sedemikian sehingga setiap jalan dilalui sedikitnya sekali yang sekaligus meminimumkan jarak total yang dilalui. Dalam hal ini suatu jalan yang diberlakukan dua arah cukup dilalui sekali.

Sekarang, masalahnya dapat diformulasikan secara matematik sebagai berikut. Untuk suatu *network*, akan ditemukan rute terpendek yang meliputi semua *edge* dalam *network* tersebut, dimana jika ada suatu *edge* yang bertanda dua arah (yaitu yang menyatakan jalan dengan dua arah), maka cukup diambil salah satu arah saja. Jika kedua arah tersebut harus dilewati juga, E. Beltrami dan L. Bodin di tahun 1974 dalam referensi [9] telah memberikan prosedur untuk menyelesaikan masalah semacam ini. Karenanya, referensi [9] dipakai sebagai dasar utama dalam tulisan ini dengan menambah satu kendala bahwa suatu jalan dengan dua arah cukup dilalui sekali saja. Ternyata masalah yang dipandang di sini menjadi lebih kompleks dibanding dengan yang telah disampaikan oleh Beltrami dan Bodin dalam [9].

## 3. ANALISA MATEMATIK

Kita asumsikan bahwa rute terpendek yang akan ditentukan haruslah berupa suatu *walk* tertutup. Secara formal suatu *walk* tertutup adalah suatu barisan dari *edge-edge* $(e_1, e_2, \ldots, e_n)$ sedemikian sehingga akhir verteks dari $e_i$ merupakan awal dari $e_{i+1}$, dan $e_n$ berakhir pada awal dari $e_1$. Asumsi ini mengisyaratkan bahwa ada banyak cara mengawali *walk* tertutup seperti ini.

Secara umum setiap *walk* tertutup yang meliputi semua *edge*, mungkin akan melewati beberapa *edge* lebih dari sekali. Dalam beberapa *graph* khusus, suatu *walk* tertutup yang melewati masing-masing *edge* tepat sekali dapat diperoleh. Suatu *walk* tertutup yang meliputi/melewati semua *edge* tepat sekali disebut suatu *circuit* Euler. Jelas bahwa setiap *circuit* Euler merupakan *walk* terpendek. Syarat-syarat bagi adanya suatu *circuit* Euler, akan memberikan pada kita bagaimana cara mengkonstruksi suatu *walk* terpendek, bahkan seandainya *circuit* Euler tersebut tidak ada.

Kita definisikan derajat masuk dari suatu verteks adalah banyaknya *edge* yang berarah masuk atau menuju verteks tersebut. Sedangkan derajat keluar dari suatu verteks adalah banyaknya *edge* yang berarah keluar dari verteks tersebut Suatu *graph* disebut terhubung (*connected*) jika setiap dua verteks dihubungkan dengan suatu *path* (barisan dari *edge-edge* yang berbeda, dengan verteks awal tidak sama dengan verteks akhir). Dengan demikian, jelas bahwa *graph* yang mempunya *circuit* Euler adalah terhubung.

**Teorema 1.**
Suatu *graph* mempunyai suatu *circuit* Euler jika dan hanya jika *graph* tersebut terhubung dan untuk setiap verteks berlaku bahwa derajat masuk sama dengan derajat keluar.

**Bukti Teorema 1.**
Jelas bahwa *graph* berarah yang memuat *circuit* Euler pasti merupakan *graph* terhubung. Andaikan *graph* tersebut mempunyai verteks $v$ dimana derajat masuk tidak sama dengan derajat keluar, maka *circuit* yang dibuat pada *graph* tersebut pada suatu saat tidak dapat melewati suatu *edge* yang berkaitan dengan $v$ atau *circuit* akan terhenti di $v$ sebelum melewati semua *edge* dari *graph* tersebut. Terjadi kontradiksi.

Sebaliknya, jika suatu *graph* $G$ terhubung dan untuk setiap verteksnya berlaku bahwa derajat masuk sama dengan derajat keluar, maka untuk setiap verteks, kita dapat memilih sepasang *edge* masuk dan keluar. Dengan demikian kita dapat memilih suatu barisan dari *edge-edge* sebesar mungkin sedemikian sehingga barisan tersebut membentuk *cycle* (*walk* tertutup dengan semua verteks berbeda) $C_1$. Kita hapus semua *edge* yang terkait dengan $C_1$ dari $G$, sehingga diperoleh $G - C_1$, yang setiap verteksnya juga mempunyai derajat masuk yang sama dengan derajat keluarnya. Sehingga dengan cara yang sama dapat diperoleh *cycle* $C_2$. Demikian seterusnya dapat kita peroleh *cycle- cycle* $C_3$, $C_4$, … $C_n$ yang masing-masing mempunyai *edge* yang *disjoint*. Selanjutnya kita dapat mengombinasikan setiap dua *cycle* dengan suatu verteks yang berserikat pada kedua *cycle* tersebut. Demikian seterusnya, karena $G$ terhubung, maka diperoleh suatu *circuit* tunggal, yaitu *circuit* Euler. □

Perhatikan bahwa konstruksi pada bukti Teorema 1 di atas akan memberikan kepada kita sepasang *edge* masuk dan keluar di setiap verteks dalam perancangan untuk meminimumkan pembalikan arah yang tidak diinginkan di setiap verteks.

Sekarang kita pandang masalah menentukan suatu rute terpendek yang meliputi semua *edge* dalam suatu *network* sembarang, dengan batasan untuk *edge* dua arah cukup dilewati sekali. Kesulitan akan muncul di verteks-verteks yang mempunyai derajat masuk tidak sama dengan derajat keluar. Pada saatnya nanti, di verteks ini kendaraan akan menuju verteks berikutnya melewati *edge* yang pernah dilaluinya, tanpa memberikan pelayanan pada *edge* tersebut. Biaya yang dikeluarkan untuk melewati *edge* yang pernah dilalui seperti ini disebut biaya ekstra. Biaya ekstra inilah yang harus diminimumkan.

Kita definisikan derajat $d(x)$ dari suatu verteks $x$, adalah derajat keluar dari $x$ dikurangi derajat masuknya. Jika suatu verkets $x$ mempunyai $d(x) < 0$, artinya derajat masuk lebih besar, maka rute perjalanan kita akhirnya suatu ketika setelah mencapai $x$ harus melalui suatu *edge* (dengan biaya ekstra) yang berawal dari $x$, guna melanjutkan perjalanan untuk mencapai *edge-edge* yang lain. Kesulitan yang sama akan muncul jika $d(x) > 0$.

Kita dapat menambahkan *edge-edge* ekstra ke dalam graph yang ingin kita tentukan penyelesaiannya. Perhatikan bahwa tambahan *edge* ekstra akan memperluas *graph* semula ke dalam suatu *graph* yang mempunyai *circuit* Euler, yaitu ke dalam graph berarah dan terhubung dimana masing-masing verteks $x$ mempunyai $d(x) = 0$. Sehingga masalah menentukan rute terpendek menjadi masalah menentukan himpunan dari *edge-edge* dengan panjang minimum yang akan ditambahkan pada *graph* semula agar supaya $d(x) = 0$ untuk setiap verteks $x$.

**Teorema 2.**
Misalkan $G$ adalah graph berarah dengan suatu bobot yang diberikan pada setiap *edge*, dan $N$ adalah graph yang lebih besar dan memuat $G$. Misalkan $A$ adalah koleksi *edge-edge* (suatu *edge* tunggal dapat dihitung beberapa kali dalam $A$) dengan bobot minimal yang diambil dari graph $N$ sedemikian sehingga penambahan *edge* dari $A$ ke $G$ membuat $d$(x) = 0 untuk setiap verteks $x$ dalam *graph* yang baru. Kita asumsikan suatu himpunan $A$ semacam ini wujud. Maka $A$ dapat dipartisi ke dalam *path-path* dari verteks berderajat negatif ke verteks berderajat positif. Jika $d(x) = -k$ (atau $+k$) dalam $G$, maka sebanyak $k$ *path* berawal (berakhir) di $x$.

**Buti Teorema 2.**
Misalkan $G^*$ adalah *graph* yang hanya dibangun oleh *edge-edge* di dalam $A$ (jika suatu *edge* dijumpai sebanyak $k$ kali, maka *edge* tersebut dinyatakan sebanyak $k$). Hanya verteks-verteks dari $G^*$ yang berderajat tak-nol akan merupakan verteks-verteks dari $G$ yang berderajat tak-nol, karena jumlah derajat verteks dalam $G$ dan derajat dalam $G^*$ adalah nol. Marilah kita ambil pasangan-pasangan seperti dalam bukti Teorema 1, pasangan *edge* masuk dan keluar pada masing-masing verteks dari $G^*$ sebanyak mungkin. Kita kembali memperoleh suatu himpunan yang beranggotakan barisan-barisan dari *edge*. Awal dari setiap barisan tersebut berupa verteks yang berderajat positif (berderajat negatif dalam $G$), dan akhir dari barisan tersebut berupa verteks

berderajat negatif (berderajat posisitf dalam *G*). Jika suatu barisan membentuk *cycle*, kita dapat menghapus *edge-edge cycle* tersebut dari *A* tanpa merubah derajat dari setiap verteks. Sehingga dengan minimalitas dari *A*, semua barisan berupa *path* dari verteks berderajat negatif dalam *G* ke verteks berderajat positif dalam *G*. □

Teorema 2 memberikan kepada kita bagaimana meminimalkan biaya ekstra. Kita harus melihat pada semua jalan dari pasangan-pasangan verteks negatif dengan verteks positif dengan *path* ekstra, dan kemudian diambil himpunan dari pasangan-pasangan tersebut yang meminimumkan jumlah panjang dari *path-path* terpendek antara pasangan-pasangan verteks-verteks positif dan negatif. Jika himpunan minimal dari pasangan-pasangan ekstra diperoleh, kita tambahkan *edge-edge* ekstra tersebut ke dalam *graph* semula. Sekarang *graph* yang diperoleh mempunyai verteks-verteks berderajat nol, dan kita dapat menggunakan metode dalam bukti Teorema 1 untuk memperoleh *circuit* Euler. Jika diperoleh himpunan minimal dari *edge-edge* ekstra yang kita tambahkan pada *graph* semula, kita dapat memperoleh *circuit* Euler dalam beberapa cara yang berbeda.

Untuk menentukan pasangan minimal ini, kita memerlukan suatu matriks $W_{ij}$ yang memberikan panjang dari *path-path* terpendek antara verteks negatif ke-*i* $x_i$ dan verteks positif ke-*j* $y_j$, untuk semua *i, j*, dimana matriks $W_{ij}$ mempunyai suatu baris untuk setiap verteks negatif dan suatu kolom untuk setiap verteks positif, sedangkan unsur $a_{ij}$ adalah panjang atau bobot (bisa diartikan juga sebagai biaya perjalanan) dari *path* terpendek yang berawal dari verteks negatif ke-*i* $x_i$ ke verteks positif ke-*j* $y_j$. Untuk menentukan *path* terpendek antara dua verteks ini digunakan algoritma *path* terpendek (*shortest path algoritma*) yang dapat dijumpai dalam referensi [2], [7], dan [10]. Selanjutnya dari matriks $W_{ij}$, ditentukan himpunan minimal dari semua pasangan-pasangan verteks negatif dan positif sedemikian sehingga meminimumkan panjang keseluruhan dari *path-path* terpendek yang menghubungkan semua pasangan verteks tersebut. Hal ini tidak lain meminimumkan total jarak perjalanan dari semua $x_i$ ke suatu $y_j$. Masalah ini sering muncul dalam riset operasi dan disebut sebagai masalah transportasi klasik. Bagi pembaca yang ingin mempelajari masalah transportasi klasik, dipersilakan melihat di dalam referensi [7] dan [10].

Penyelesaian dari masalah transportasi ini memberikan suatu himpunan minimal dari pasangan-pasangan yang memberikan kita *path-path* dari *edge-edge* ekstra yang akan ditambahkan. Misalkan *G'* adalah *G* ditambah dengan *edge-edge* ekstra ini. Sekarang kita siap membentuk suatu *circuit* Euler untuk *G'*. Untuk membentuk *circuit* Euler kita perlu memasangkan *edge* masuk dan keluar di setiap verteks yang diperlukan untuk mengkonstruksi *cycle-cycle* seperti dalam Teorema 1. Pemasangan ini dilakukan untuk meminimumkan perubahan arah yang tidak diinginkan. Untuk keperluan ini kita dapat memberikan bobot untuk masing-masing tipe yang mungkin dari pasangan-pasangan tersebut. Untuk setiap verteks, kita tentukan sepasang *edge* masuk dan keluar dalam G*'* yang meminimumkan jumlah total bobot perubahan arah.

Untuk menyelesaikan masalah ini, kita buat matriks $W^{(k)}$ *untuk verteks* ke-$k$ $v_k$ dengan suatu baris untuk setiap *edge* masuk ke $v_k$ dan suatu kolom untuk setiap *edge* keluar dari $v_k$. Unsur $w_{ij}$ dalam $W^{(k)}$ adalah bobot untuk pasangan *edge* masuk ke-$i$ dengan *edge* keluar ke-$j$. Masalah pasangan minimal dari elemen-elemen baris dengan elemen-elemen kolom seperti ini disebut masalah penugasan (*assignment problem*). Bagi pembaca yang ingin mempelajari tentang masalah penugasan dipersilakan membaca di dalam referensi [7] dan [10].

Setelah masalah penugasan diselesaikan untuk setiap verteks, kita membentuk *cycle-cycle* seperti dalam bukti Teorema 1. Selanjutnya kita kombinasikan *cycle-cycle* ini secara bersamaan seperti dalam bukti tersebut untuk memperoleh *circuit* Euler. Memasangkan kembali yang diperlukan untuk mengombinasikan pasangan-pasangan dari *cycle-cycle* dapat membentuk perubahan arah yang tidak diinginkan. Biasanya dalam praktek hanya ada beberapa *cycle*, oleh karenanya penentuan suatu cara optimal dalam menggabungkan *cycle-cycle* ini secara bersamaan untuk memperoleh *circuit* Euler menjadi tidak begitu penting.

## 4. RINGKASAN PROSEDUR

Analisa yang dikembangkan pada bagian 3 dapat dibagi ke dalam tiga langkah. Pertama, kita tambahkan suatu himpunan dengan panjang minimal dari *edge-edge* sedemikian sehingga graph *G'* yang dihasilkan mempunyai suatu *circuit* Euler. Kedua, kita pasangkan *edge* masuk dan keluar di setiap verteks untuk meminimumkan pembalikan arah yang tidak diinginkan, dan gunakan pasangan-pasangan ini untuk mengkonstruksi suatu *cycle* untuk setiap komponen dari *G'*. Ketiga, kita gabungkan *cycle-cycle* untuk semua komponen dari *G'* menjadi suatu *circuit* Euler dalam *G'* yang merupakan rute terpendek yang ingin kita tentukan.

Kita ansumsikan graph $G$ yang diberikan termuat dalam graph $H$ yang lebih besar dari graph $G$. Secara rinci ketiga langkah tersebut dapat kita tuliskan sebagai berikut:

1. Tambahan *edge-edge* ke $G$ untuk memperoleh suatu graph *G'* dengan *cycle-cycle* dalam setiap komponen-komponennya.
   (a) Tentukan matriks dari jarak-jarak terpendek dalam $H$ antara verteks-verteks negatif $x_i$ dan verteks-verteks positif $y_j$ dari $G$. Gunakan algoritma *path* terpendek untuk menentukan jarak terpendek. Seandainya penentuan verteks-verteks negatif dan verteks-verteks positif mempunyai banyak kemungkinan, lanjutkan ke langkah 1(c).
   (b) Selesaikan masalah transportasi dengan matriks dari langkah 1(a) dengan baris asal $-d(x_i)$ dan kolom tujuan $d(y_j)$. Kemudian ke langkah 1(g).
   (c) Tentukan $d(x)$ untuk semua verteks sedemikian sehingga $|d(x)|$ sekecil mungkin. Selanjutnya kita bentuk himpunan $D$ yang anggota-anggotanya adalah semua verteks $x$ dengan $d(x) \neq 0$. Kemudian kita

tentukan jarak terpendek (*path* terpendek) $p(x_i,x_j)$ untuk semua $x_i$ dan $x_j \in D$.

(d) Bentuk *bigraph* dari $D$ yang terdiri dari dua kelompok (satu kelompok verteks-verteks negatif $x_i$ dan satu kelompok verteks-verteks positif $y_j$) untuk semua kombinasi yang mungkin dari $D$. Kita peroleh $n$ bigraph yang berbeda dari $D$.

(e) Untuk setiap *bigraph* hasil dari 1(d), tentukan matriks transportasi dari jarak-jarak terpendek antara verteks-verteks negatif dan verteks-verteks positif. Kemudian selesaikan masalah transportasi ini dengan baris asal $-d(x_i)$ dan kolom tujuan $d(y_j)$.

(f) Dari hasil 1(e), pilih suatu bigraph dengan biaya transportasi paling rendah (minimum).

(g) Tambahkan *edge-edge* ekstra (misalnya dengan *edge* berupa garis putus-putus) pada $G$ sedemikian sehingga ada $k$ *edge* tambahan (dengan garis putus-putus) yang diperoleh dari langkah 1(b) atau 1(f). Sebut *graph* baru tersebut sebagai $G'$.

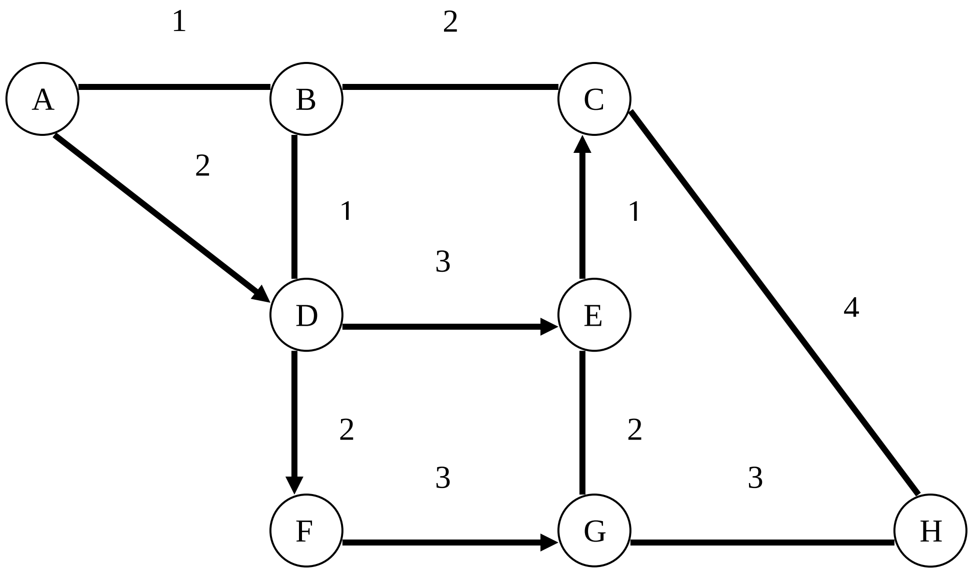
2. Tentukan *cycle-cycle* pada setiap komponen dari $G'$.
   (a) Pasangkan *edge* masuk dan keluar di setiap verteks pada $G'$ dengan pendekatan masalah penugasan.
   (b) Bentuk *cycle-cycle* yang dibangun dari pasangan-pasangan dalam langkah-langkah 2(a) untuk memperoleh suatu *cycle* dalam setiap komponen dari $G'$.

3. Tentukan *circuit* Euler pada $G'$.
   (a) Gabungkan *cycle-cycle* dari semua komponen dalam $G'$ untuk memperoleh *circuit* Euler dalam $G'$.

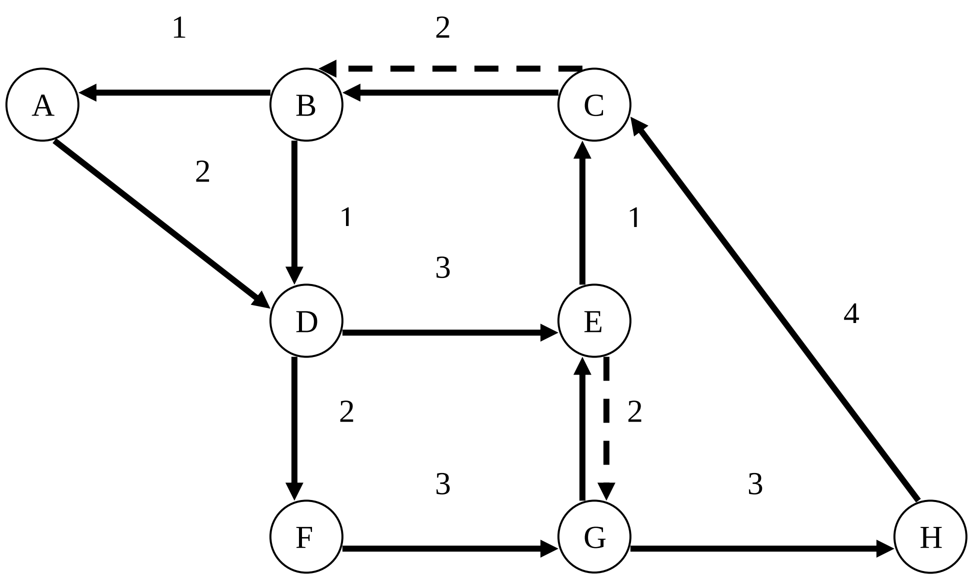
## 5. CONTOH PENERAPAN

Pandang Gambar 1 yang memberikan situasi jalan pada suatu daerah yang diberikan dalam bentuk *graph* berarah. *Edge* tanpa tanda panah pada Gambar 1 menunjukkan bahwa jalan yang dinyatakan oleh *edge* tersebut berlaku dua arah, sedangkan bilangan pada setiap *edge* menyatakan bobot setiap jalan yang dinyatakan oleh *edge* tersebut. Akan digunakan prosedur yang diberikan pada bagian 4 untuk memperoleh suatu rute terpendek yang meliputi semua *edge* dalam *network* yang diberikan pada Gambar 1.

Pada langkah 1 akan diperoleh dua *edge* ekstra (diberikan dengan garis putus-putus), yaitu *edge* dari verteks C ke verteks B, dan *edge* dari verteks E ke verteks G (lihat Gambar 2).




**Gambar1**. *Network* pada suatu daerah




**Gambar 2.** *Graph* berarah dengan tambahan dua *edge* ekstra yang merupakan hasil langkah 1 untuk situasi jalan pada Gambar 1.

Pada langkah 2 diperoleh dua *cycle* yang masing-masing mempunyai *edge* yang terpisah (*disjoint*), yaitu *cycle* $C_1$ : D – F – G – E – C – B – D dan *cycle* $C_2$ : D – E – G – H – C – B – A – D. *Cycle*-*cycle* ini dapat diawali dari verteks sembarang dan pada prakteknya dapat diperoleh himpunan dari *cycle*-*cycle* yang berbeda.

Akhirnya pada langkah 3 diperoleh suatu *circuit* Euler yang merupakan rute terpendek yang ingin kita tentukan, yaitu D – F – G – E – C – B – D – E – G – H – C – B – A – D.

## 6. KESIMPULAN

1. Tulisan ini mengemukakan prosedur matematik untuk menentukan rute terpendek kendaraan tunggal yang meliputi semua *edge* pada suatu *network* tanpa adanya kendala urutan *edge* yang harus dilalui.
2. Masalah menentukan rute terpendek kendaraan tunggal yang meliputi semua *edge* pada suatu *network* tanpa adanya kendala urutan *edge* yang harus dilalui, analog dengan masalah menentukan suatu himpunan dari *edge-edge* ekstra dengan bobot minimum yang ditambahkan pada *network* asli guna menyeimbangkan derajat masuk dan derajat keluar pada setiap verteks.
3. Dengan menambah *edge-edge* ekstra dengan bobot minimum (jika diperlukan) pada network asal, rute terpendek kendaraan tunggal yang meliputi semua *edge* tanpa adanya kendala urutan *edge* yang harus dilalui pada *network* yang baru (*network* asal + *edge-edge* ekstra dengan bobot minimum) akan berupa suatu *circuit* Euler.

## DAFTAR PUSTAKA


1. Andrews, J.G. & McLone, R.R., *Mathematical Modelling,* Butterworth & Co. Ltd., London, 1976.
2. Bertsekas, D.P., *Linear Network Optimization: Algorithms and Codes.* The MIT Press, Massachusetts, 1981.
3. Bodin, L., et al., *Routing and Scheduling of Vehicles and Crews The State of The Art,* Pergamon Press, Oxford, 1983.
4. Chartrand, Garry, *Introductory Graph Theory,* Dover Pub.Inc., New York, 1977.
5. Deo, Narsingh, *Graph Theory with Applications to Engineering and Computer Science,* Prentice-Hall Inc., New Delhi, 1986.
6. Lucas, W.F., et al., *Discrete and System Models,* Springer-Verlag. New York, 1983.
7. Mital, K.V., *Optimization Methods in Operation Research & System Analysis,* Wiley Eastern Ltd ., New Delhi, 1979.
8. Murthy, D.N.P., et al., *Mathematical Modelling: A Tool for Problem Solving in Engineering, Physical, Biological, and Social Sciences,* Pergamon Press, Oxford, 1990.
9. Teodorovic, D., *Transportation Networks,* Gordon & Breach Science Publishers, New York, 1986.
10. Winston, W.L., *Operation Research Applications and Algorithms,* Brooks/Cole, Canada, 2004.