

MEMBANDINGKAN KEMANGKUSAN ALGORITMA DINIC DAN ALGORITMA PELABELAN FORD-FULKERSON UNTUK MASALAH ARUS MAKSIMUM

Nerli Khairani

Jenny Sirait

Abstrak

Teori graf merupakan salah satu cabang matematika yang paling banyak aplikasinya dalam kehidupan sehari-hari. Salah satu bentuk dari graf adalah jaringan (*network*), yaitu graf berarah berbobot sederhana yang memiliki simpul sumber (*source*), simpul tujuan (*sink*), dan tiap sisinya mempunyai kapasitas tertentu. Algoritma Dinic dan algoritma Pelabelan Ford-Fulkerson merupakan algoritma yang digunakan dalam pemecahan masalah arus maksimum. Algoritma Dinic memanfaatkan jaringan sisa. Pada jaringan sisa ini diidentifikasi *f-augmenting path* terpendek melalui *layered network*, kemudian dikonstruksi suatu *blocking flow* yang dapat digunakan untuk menentukan arus maksimum. Algoritma pelabelan Ford-Fulkerson berisi 2 fase. Fase pertama melakukan pelabelan untuk memeriksa apakah terdapat *f-augmenting path*. Jika terdapat *f-augmenting path*, maka menentukan dan menambahkan *f-augmenting path* pada arus *f*. Algoritma Dinic dan algoritma pelabelan Ford-Fulkerson memberikan solusi arus maksimum yang sama yaitu sebanyak 7. Algoritma pelabelan Ford-Fulkerson lebih mangkus dibandingkan dengan algoritma Dinic karena algoritma pelabelan Ford-Fulkerson membutuhkan waktu dan ruang memori yang lebih sedikit dalam mencari solusi dari masalah arus maksimum.

Kata kunci: *algoritma pelabelan ford-fulkerson, algoritma dinic, masalah arus maksimum, jaringan*

PENDAHULUAN

Teori graf merupakan salah satu cabang matematika yang paling banyak aplikasinya dalam kehidupan sehari-hari. Salah satu bentuk dari graf adalah *flow network*, yaitu graf berarah yang tiap sisinya mempunyai kapasitas tertentu. *Flow network* sering digunakan untuk memodelkan jaringan yang sering menjadi masalah dalam kehidupan seperti jaringan lalu lintas, masalah arus listrik, jaringan komunikasi, masalah produksi, distribusi, perencanaan proyek, penentuan lokasi,

jaringan pipa air, dan lain-lain. Transportasi barang dari lokasi sumber ke lokasi tujuan yang melewati beberapa lokasi-antara merupakan salah satu contoh masalah optimasi yang dapat didefinisikan ke dalam bentuk graf atau jaringan.

Optimasi adalah salah satu disiplin ilmu dalam matematika yang fokus untuk mendapatkan nilai minimum atau maksimum secara sistematis dalam berbagai kasus. Terdapat 5 algoritma yang bisa digunakan dalam menyelesaikan

Nerli Khairani adalah Dosen Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan; Jenny Sirait adalah Alumni Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan

masalah *flow network*. Kelima algoritma tersebut antara lain algoritma pelabelan Ford-Fulkerson, algoritma pelabelan Edmonds dan Karp, algoritma Dinic, algoritma MPM (Malhotra, Pramodh Kumar & Maheswari), dan algoritma Goldberg-Tarjan (Thulasiraman & Swamy, 1992).

Menurut Moligane, Algoritma Dinic lebih baik dibandingkan algoritma Pelabelan Ford-Fulkerson, maupun algoritma Pelabelan yang telah dimodifikasi oleh Edmonds-Karp (Yudhianto, 2003). Algoritma Dinic diperkenalkan oleh Dinic pada tahun 1970. Sedangkan algoritma Ford-Fulkerson merupakan algoritma pertama dalam menangani masalah *flow network*, ditemukan oleh Ford dan Fulkerson pada tahun 1956. Dalam metode ini mereka ingin memaparkan suatu cara untuk mencari kapasitas maksimum suatu aliran di dalam jaringan. Selain menghemat waktu dengan mengolah algoritma ini menjadi suatu program, metode ini juga akan efektif untuk para penggunanya dalam melakukan suatu proses, tindakan, atau pengambilan keputusan untuk tujuan tertentu dengan mengetahui arus

maksimum yang terdapat dalam suatu jaringan (Ahuja & Orlin, 1989).

Dalam penulisan ini akan dibahas algoritma Dinic dan Algoritma Pelabelan Ford-Fulkerson dengan langkah per langkah hingga akhirnya ditemukan arus maksimum. Setelah itu dibandingkan kemangkusan dari kedua algoritma diatas. Dua simpul penting di graf G adalah simpul sumber s dan simpul tujuan t . Setiap jalur di G telah terkait dengan angka positif yang disebut kapasitas. Sebuah arus dalam jaringan adalah kumpulan arus rangkaian yang memiliki sifat bahwa jumlah dari banyaknya semua arus rangkaian yang terkandung dalam jalur manapun adalah tidak lebih besar dari kapasitas busur itu (Ford & Fulkerson, 1956).

Berdasarkan hasil dari penelitian yang telah dilakukan oleh Yudhianto, algoritma Dinic dapat digunakan untuk menyelesaikan masalah arus maksimum yang dimodelkan dalam suatu jaringan transportasi (Yudhianto, 2003). Dalam Kamus Besar Bahasa Indonesia, kata dasar kemangkusan dan keefektifan, mangkus dan efektif, sama-sama memiliki arti berhasil guna.

TINJAUAN PUSTAKA

Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi G

$= (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *nodes*) dan E adalah

himpunan jalur (*edges* atau *arcs*) yang menghubungkan sepasang simpul (Munir, 2007).

Graf Berarah

Graf yang setiap jalurnya diberikan arah disebut sebagai graf berarah. Pada graf berarah, (u,v) dan (v,u) menyatakan dua buah jalur yang berbeda, dengan kata lain $(u,v) \neq (v,u)$.

(Siahaan, 2010).

Graf Berbobot

Sebuah graf dengan bilangan-bilangan pada jalur-jalurnya disebut graf berbobot. (Johnsonbaugh, 2002).

Path

Path dengan panjang n dari v ke w adalah *walk* dari v ke w yang semua jalurnya berbeda. *Path* v ke w dituliskan sebagai $v = v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n = w$ dengan e_i untuk $i = 1, \dots, n$. (Siang, 2006).

Jaringan

Sebuah jaringan adalah suatu graf berarah berbobot sederhana yang memenuhi :

- Simpul bertanda, yang merupakan sumber (*source*), tidak mempunyai jalur yang masuk.
- Simpul bertanda, yang merupakan tujuan (*sink*), tidak mempunyai jalur yang keluar.
- Bobot C_{ij} dari jalur berarah (i, j) disebut kapasitas dari (i, j) , merupakan bilangan tak negatif. (Johnsonbaugh, 2002)

Kapasitas Sisa

Misalkan f adalah arus pada $G = (V, E)$, besarnya arus tambahan yang dapat ditambahkan dari x ke y tanpa melebihi kapasitas $c(x,y)$ adalah kapasitas sisa (x,y) .
 $c_f(x,y) = c(x,y) - f(x,y)$ (Handoyo, 2011).

Nilai Arus

Nilai (*value*) dari suatu arus f di jaringan transportasi $G = (V, E)$ dengan s adalah simpul *source* di G , yang dinotasikan dengan $val(f)$, didefinisikan sebagai berikut :

$$val(f) = \sum_{j \in V} f(s, j)$$

Dengan menggunakan kendala konservasi, dimana t adalah simpul *sink* di G didapatkan bahwa : $val(f) = \sum_{j \in V} f(s, j) = \sum_{j \in V} f(j, t)$ (Thulasiraman & Swamy, 1992).

Cut

Misalkan $G = (V, E)$ adalah graf terhubung dengan $V = V_1 \cup V_2$ dan $V_1 \cap V_2 = \emptyset$. Maka himpunan $S = \{(u,v) \mid u \in V_1 \text{ dan } v \in V_2\}$ dinamakan cut dari graf G , dan dinyatakan dengan $\langle V_1, V_2 \rangle$. Kapasitas cut $K = \langle S, \bar{S} \rangle$ pada jaringan transportasi $G = (V, E)$ didefinisikan sebagai : $c(K) = c(S, \bar{S}) = \sum_{i \in S} \sum_{j \in \bar{S}} c(i, j)$ (Thulasiraman & Swamy, 1992).

Lemma 1. (Handoyo, 2011)

Misalkan f adalah arus pada arus jaringan G dengan sumber s dan tujuan t , dan

misalkan (S,T) adalah *cut* dari G . Maka arus yang melewati (S,T) adalah

$$f(S,T) = |f|$$

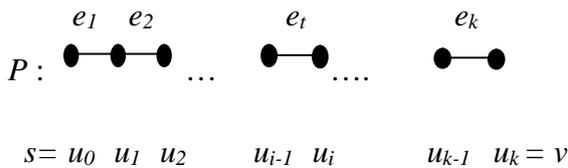
Bukti:

Kita perhatikan bahwa $f(S-s, V) = 0$,

karena arus tersebut tidak memiliki sumber s.

$$\begin{aligned} f(S,T) &= f(S, V) - f(S, S) = f(S, V) \\ &= f(s, V) + f(S-s, V) = f(s, V) = |f| \end{aligned}$$

Sisi Maju dan Sisi Mundur



P adalah suatu *path* di jaringan transportasi $N=(V,E)$ dengan arus f dari *source* s ke simpul v . Maka jalur e_i di P dinamakan jalur maju dari P jika jalur e_i berorientasi dari u_{i-1} ke u_i dan selainnya dinamakan jalur mundur dari P . Untuk setiap jalur di P , misalkan:

***f*-unsaturated Path**

Suatu *path* dikatakan *f-unsaturated* jika semua jalur maju dari *path* adalah jalur *f-unsaturated* dan semua jalur mundur dari *path* adalah jalur *f*-positif (Thulasiraman & Swamy, 1992).

***f*-augmenting Path**

Suatu *path* $s-t$ P dikatakan *f-augmenting* jika P adalah *f-unsaturated*. Berdasarkan definisi *f-augmenting path*, terlihat bahwa

1. $\epsilon_i(P) = c(e_i) - f(e_i)$, jika e_i adalah jalur maju,
2. $\epsilon_i(P) = f(e_i)$, jika e_i adalah jalur mundur.

Untuk suatu *path* P didefinisikan suatu pemadanan yang memadankan *path* P dengan bilangan real tak negatif $\epsilon(P)$ yang didefinisikan sebagai : $\epsilon(P) = \min_i \{\epsilon_i(P)\}$

Perhatikan bahwa nilai $\epsilon(P) \geq 0$, $f(i,j) > 0$ dan f -nol jika $f(i,j) = 0$ (Thulasiraman & Swamy, 1992).

Arus Maksimum

Suatu arus f' di jaringan transportasi G dikatakan maksimum jika tidak ada arus f di G sedemikian rupa sehingga $val(f) > val(f')$. Teorema penting dalam masalah arus maksimum adalah Teorema *maximum flow minimum cut* Ford-Fulkerson. Berikut ini diberikan terlebih dahulu definisi *cut minimum* (Thulasiraman & Swamy, 1992).

f-augmenting path merupakan *path* dari s ke t yang memuat jalur-jalur yang dapat ditingkatkan lagi arusnya karena belum sampai pada batas kapasitasnya. Dari definisi $\epsilon(P)$ terlihat bahwa untuk suatu *path* $s-t$ P , berlaku $\epsilon(P) > 0$ jika dan hanya jika P adalah *f-augmenting path* (Thulasiraman & Swamy, 1992).

Teorema Max-Flow Min-Cut

Pada suatu jaringan transportasi, nilai arus maksimum $val(f)$ sama dengan kapasitas cut minimum $c(K)$ (Thulasiraman & Swamy, 1992).

Bukti:

Karena $c(K) = c(S, \bar{S})$, kita anggap jalur berarah (v, w) dengan $v \in S$ dan $w \in \bar{S}$. Misalkan $s \in S$, terdapat suatu f -unsaturated s - v path. Jalur (v, w) harus f -saturated. Tidak mungkin terdapat f -unsaturated s - w path karena $w \in \bar{S}$. Sehingga $f(\bar{S}, S) = 0$, maka $val(f) = f(S, \bar{S}) - f(\bar{S}, S) = f(S, \bar{S}) = c(S, \bar{S})$

Algoritma Dinic

Algoritma Dinic digunakan untuk memecahkan masalah arus maksimum dengan menggunakan jaringan transportasi N , jaringan sisa, jaringan pelapis (*layered network*) dan arus penghambat (*blocking flow*) (Thulasiraman & Swamy, 1992).

Jaringan Sisa

Algoritma Dinic mengatasi masalah arus maksimum dengan memanfaatkan jaringan sisa, yaitu jaringan dengan jalur-jalurnya mempunyai kapasitas positif. Misalkan diberi suatu jaringan transportasi $N = (V, E)$ dengan arus f . Maka $N_f = (V, E_f)$ dikatakan suatu jaringan sisa relatif terhadap arus f , jika dan hanya jika:

1. Himpunan simpul V di N_f sama dengan himpunan simpul V di N .
2. Untuk setiap jalur $e = (u, v)$ di N , maka N_f mempunyai:
 - (i). Jalur $e' = (u, v)$, jika $f(e) < c(e)$, dengan kapasitas sisa $c_f(e') = c(e) - f(e)$.
 - (ii). Jalur $e' = (v, u)$, jika $f(e) < 0$, dengan kapasitas sisa $c_f(e') = f(e)$ (Thulasiraman & Swamy, 1992).

Layered Network

Akan dibuat konstruksi dari suatu layered Network relatif terhadap arus f dengan menggunakan jaringan transportasi $N = (V, E)$ dengan simpul *source* s dan simpul *sink* t , dan jaringan sisa N_f yang telah didapatkan.

Langkah-langkah *layered network* :

1. $V_0 = \{s\}$ dan $i = 0$.
2. $T = \{v | v \notin V_j \text{ untuk } j \leq i, \text{ dan } (u, v) \in E_f \text{ untuk suatu } u \in V_i\}$
3. Jika $T = \emptyset$, maka arus f adalah maksimum. Berhenti.
4. Jika simpul *sink* $t \in T$, maka $l = i + 1$, $V_l = \{t\}$ dan $E_l = \{(u, v) \in E_f | u \in V_i \text{ dan } v \in V_{i+1}\}$. Masukkan semua

jalur di E_i , dalam *layered network*, berhenti.

5. $V_{i+1} = T$, $E_i = \{(u, v) \in E_f | u \in V_i \text{ dan } v \in V_{i+1}\}$. Dan masukkan semua jalur di E_i dalam *layered network*.
6. $i = i + 1$ dan pergi ke langkah 2.

Himpunan-himpunan simpul V_i yang didefinisikan pada *Layered Network* dinamakan lapis (*layer*) dari N . Kapasitas dari suatu jalur e di *layered network* adalah sama dengan $c_f(e)$ kapasitas sisa N_f (Thulasiraman & Swamy, 1992).

Konstruksi Arus Baru

Kemudian dengan menggunakan blocking flow, seperti yang didefinisikan di atas, dapat dikonstruksi suatu arus maksimum f' , yang didefinisikan $f': E \rightarrow \mathbb{R}^+ \cup \{0\}$, dengan nilai $val(f') > val(f)$ dengan cara sebagai berikut:

1. Jika (u, v) , dengan $u \in V_{i-1}$ dan $u \in V_i$ adalah suatu jalur maju, maka $f'(u, v) = f(u, v) + g(u, v)$
2. Jika (u, v) , dengan $u \in V_{i-1}$ dan $u \in V_i$ adalah suatu jalur mundur, maka $f'(v, u) = f(v, u) - g(u, v)$
3. Untuk semua jalur (u, v) yang lain di N , $f'(u, v) = f(u, v)$

Algoritma Pelabelan Ford-Fulkerson

Algoritma Pelabelan Ford-Fulkerson adalah algoritma untuk menentukan arus

maksimum pada jaringan transportasi N . Algoritma ini berisi 2 (dua) fase, yaitu:

Fase pertama adalah melakukan pelabelan untuk memeriksa apakah terdapat *f-augmenting path*. Jika tidak ada, maka dari teorema 2, f adalah arus maksimum. Jika terdapat *f-augmenting path*, maka fase kedua yang harus dikerjakan, yaitu menentukan *f-augmenting path* dan menambahkannya pada arus f . Pada fase pertama, label (d_v, Δ_v) , adalah label untuk suatu simpul v . Notasi menyatakan simpul dimana v menerima label, notasi ini juga menyatakan arah pelabelan: maju atau mundur. Jika simpul v dilabeli, maka berarti terdapat suatu *path s-v P* yang tidak dipenuhi (*unsaturated*) dan untuk *path P* ini, $\epsilon(P) = \Delta_v$ (Thulasiraman & Swamy, 1992).

Kemangkusan Algoritma

Kemangkusan algoritma diukur dari berapa jumlah waktu dan ruang memori yang dibutuhkan untuk menjalankannya. Algoritma yang mangkus ialah algoritma yang meminimumkan kebutuhan waktu dan ruang. Kemangkusan algoritma dapat digunakan untuk menilai algoritma mana yang terbaik dari kedua algoritma Dinic dan algoritma Ford-Fulkerson dengan melihat nilai yang paling maksimum dari hasil penggunaan kedua algoritma tersebut (Nurhayati, 2013).

METODE PENELITIAN

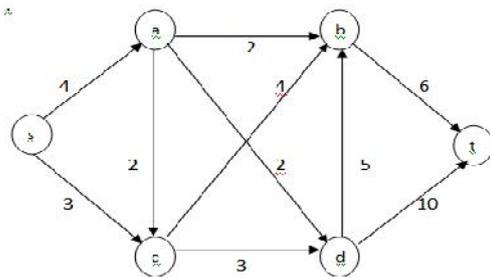
Penulis menggunakan teori-teori yang berkaitan dengan arus maksimum menggunakan algoritma pelabelan Ford-Fulkerson dan algoritma Dinic, dan prosedur penelitian yang dilakukan adalah dengan cara mengumpulkan definisi

ataupun teorema, mencari solusi dengan penggunaan algoritma Dinic dan algoritma Pelabelan Ford-Fulkerson untuk masalah arus maksimum, serta membandingkan hasil penggunaan kedua algoritma.

PEMBAHASAN

Algoritma Dinic

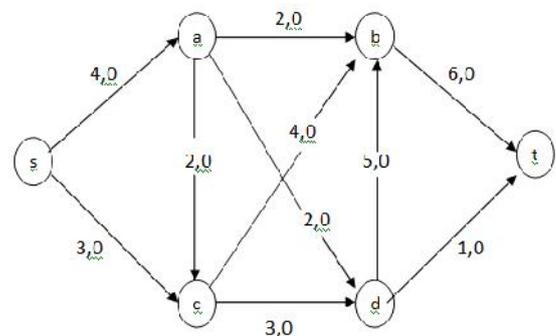
Misalkan diberikan jaringan transportasi $N = (V, E)$ beserta kapasitas jalur $c(e)$ seperti pada gambar dibawah ini.



Gambar 1. Jaringan Transportasi N dengan kapasitas jalur.

Langkah 1. $f(e) = 0$, untuk setiap jalur e di jaringan transportasi $N = (V, E)$.

Angka yang tertera pada setiap jalur berturut-turut menyatakan kapasitas jalur $c(e)$ dan arus $f(e)$ seperti gambar di bawah ini.



Gambar 2 Jaringan transportasi untuk langkah 1

langkah 2. Konstruksi jaringan sisa $N_f = (V, E_f)$ relatif terhadap arus f di N .

$N_f = (V, E_f)$ dikatakan suatu jaringan sisa relatif terhadap arus f jika dan hanya jika:

1. Himpunan simpul V di N_f sama dengan himpunan simpul V di N , yaitu $V = \{s, a, b, c, d, t\}$, dan
2. Untuk setiap jalur $e = (u, v)$ di N , maka N_f mempunyai :
 - (i). Untuk jalur $e_1 = (s, a) \in E$:
 Karena $f(e_1) < c(e_1) = 0 < 4$, maka terdapat jalur $e_1' = (s, a) \in E_f$ dengan $c_f(e_1') = 4$
 - (ii). Untuk jalur $e_2 = (s, c) \in E$:
 Karena $f(e_2) < c(e_2)$, maka terdapat jalur $e_2' = (s, c) \in E_f$ dengan $c_f(e_2') = 3$
 - (iii). Untuk jalur $e_3 = (c, d) \in E$:
 Karena $f(e_3) < c(e_3)$, maka terdapat jalur $e_3' = (c, d) \in E_f$ dengan $c_f(e_3') = 3$

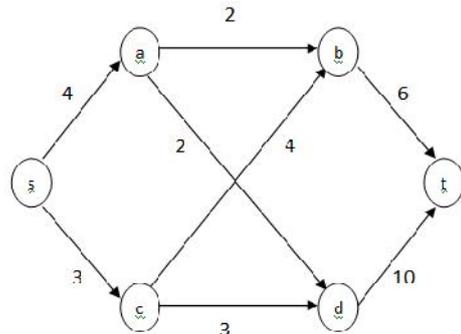
- (iv). Untuk jalur $e_4 = (a, d) \in E$:
 Karena $f(e_4) < c(e_4)$, maka terdapat jalur $e_4' = (a, d) \in E_f$ dengan $c_f(e_4') = 2$
 - (v). Untuk jalur $e_5 = (a, c) \in E$:
 Karena $f(e_5) < c(e_5)$, maka terdapat jalur $e_5' = (a, c) \in E_f$ dengan $c_f(e_5') = 2$
 - (vi). Untuk jalur $e_6 = (c, b) \in E$:
 Karena $f(e_6) < c(e_6)$, maka terdapat jalur $e_6' = (c, b) \in E_f$ dengan $c_f(e_6') = 4$
 - (vii). Untuk jalur $e_7 = (a, b) \in E$:
 Karena $f(e_7) < c(e_7)$, maka terdapat jalur $e_7' = (a, b) \in E_f$ dengan $c_f(e_7') = 2$
 - (viii). Untuk jalur $e_8 = (b, t) \in E$:
 Karena $f(e_8) < c(e_8)$, maka terdapat jalur $e_8' = (b, t) \in E_f$ dengan $c_f(e_8') = 6$
 - (ix). Untuk jalur $e_9 = (d, b) \in E$:
 Karena $f(e_9) < c(e_9)$, maka terdapat jalur $e_9' = (d, b) \in E_f$ dengan $c_f(e_9') = 5$
 - (x). Untuk jalur $e_{10} = (d, t) \in E$:
 Karena $f(e_{10}) < c(e_{10})$, maka terdapat jalur $e_{10}' = (d, t) \in E_f$ dengan $c_f(e_{10}') = 10$
3. Karena semua arus $f(e) = 0$, maka tidak ada jalur $e' = (v, u) \in E_f$.

Sehingga didapatkan suatu jaringan sisa $N_f = (V, E_f)$ dengan arus $f(e)$ yang sama seperti di N ; angka yang tertera pada setiap jalur menyatakan kapasitas sisa $c_f(e)$ di N_f , dengan $V = \{s, a, b, c, d, t\}$ dan $E_f = \{(s, a), (s, c), (a, b), (a, c), (a, d), (b, t), (c, b), (c, d), (d, b), (d, t)\}$, tidak terdapat himpunan jalur mundur di N_f seperti pada gambar 4.1.

Langkah 3. Konstruksi *layered network* relatif terhadap arus f .

Untuk jaringan N pada Gambar 2 dengan jaringan sisa pada Gambar 1, maka

didapatkan suatu *layered network* relatif terhadap arus f beserta kapasitas sisa dari jalur e adalah $c_f(i)$, seperti gambar di bawah ini.



Gambar 3. *Layered Network* Gambar 1

Iterasi 1.

Langkah 1 : $V_0 = \{s\}$ dan $i = 0$.

Langkah 2 : $T = \{v | v \notin V_j \text{ untuk } j \leq 0, \text{ dan } (u, v) \in E_f \text{ untuk suatu } u \in V_0\}$
 Ambil $j = 0$ sehingga didapatkan :
 $T = \{v | v \notin V_0 \text{ dan } (u, v) \in E_f \text{ untuk suatu } u \in V_0\}$
 $T = \{a, c\}$

Langkah 3 : $T \neq \emptyset$, maka lanjut ke langkah 4.

Langkah 4 : Simpul *sink* $t \notin T$, maka lanjut ke langkah 5.

Langkah 5 : $V_1 = T = \{a, c\}$, $E_0 = \{(u, v) \in E_f | u \in V_0 \text{ dan } v \in V_1\} = \{(s, a), (s, c)\}$.
 Jadi, jalur $\{(s, a), (s, c)\}$ adalah himpunan jalur pada LN .

Langkah 6 : $i = i + 1 = 1$ dan pergi ke langkah 2.

Iterasi 2.

Langkah 2 : $T = \{v | v \notin V_j \text{ untuk } j \leq 1, \text{ dan } (u, v) \in E_f \text{ untuk suatu } u \in V_1\}$

Ambil $j = \{0, 1\}$ sehingga didapatkan :

- Untuk $j = 0 : T_0 = \{b, c, d\}$.
 - Untuk $j = 1 : T_1 = \{b, d\}$.
- Jadi didapatkan $T = T_0 \cap T_1 = \{b, d\}$.

Langkah 3 : $T \neq \emptyset$, maka lanjut ke langkah 4.

Langkah 4 : Simpul *sink* $t \notin T$, maka lanjut ke langkah 5.

Langkah 5 : $V_2 = T = \{b, d\}$, $E_1 = \{(a, b), (a, d), (c, b), (c, d)\}$.
Jadi himpunan jalur $\{(a, b), (a, d), (c, b), (c, d)\}$ adalah jalur-jalur pada *LN*.

Langkah 6 : $i = 2$ dan pergi ke langkah 2.

Iterasi 3.

Langkah 2 : $T = \{v | v \notin V_j \text{ untuk } j \leq 2, \text{ dan } (u, v) \in E_f \text{ untuk suatu } u \in V_2\}$
Ambil $j = \{0, 1, 2\}$ sehingga didapatkan :

- Untuk $j = 0 : T_0 = \{b, t\}$.
 - Untuk $j = 1 : T_1 = \{b, t\}$.
 - Untuk $j = 2 : T_2 = \{t\}$.
- Jadi didapatkan $T = T_0 \cap T_1 \cap T_2 = \{t\}$.

Langkah 3 : $T \neq \emptyset$, maka lanjut ke langkah 4.

Langkah 4 : Simpul *sink* $t \in T$, maka didapatkan $i = i + 1 = 3$, $V_3 = \{t\}$ dan $E_2 = \{(u, v) \in E_f | u \in V_2 \text{ dan } v \in V_3\} = \{(b, t), (d, t)\}$.
Jadi himpunan jalur di E_2 adalah jalur-jalur dalam *layered network*, berhenti.

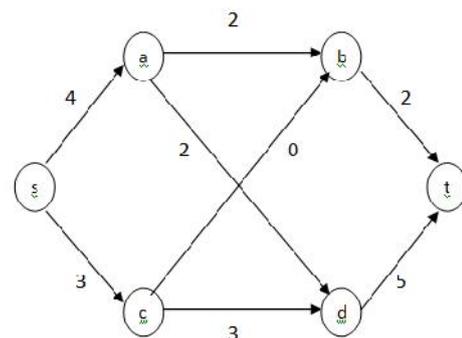
Sehingga didapat *layered network* dengan panjang $(l) = 3$, dengan kapasitas jalur $c(e)$

di *layered network* sama dengan kapasitas sisa $c_f(e)$ di N_f , dengan $V = \{s, a, b, c, d, t\}$ dan

$E = \{(s, a), (s, c), (a, b), (a, d), (b, t), (c, b), (c, d), (d, t)\}$, seperti pada Gambar 4.3. Himpunan simpul V_i , yaitu $V_0 = \{s\}$, $V_1 = \{a, c\}$, $V_2 = \{b, d\}$, dan $V_3 = \{t\}$, dinamakan *layer* dari jaringan transportasi N .

Langkah 4. Konstruksi *blocking flow* g di *layered network*.

Didefinisikan $g: E \rightarrow \mathbb{R}^+ \cup \{0\}$ dengan $g(e_i) = c_f(e_i)$, yaitu : $g(s, c) = g(c, d) = 3$, $g(s, a) = 4$, $g(a, b) = g(a, d) = g(b, t) = 2$, $g(c, b) = 0$, dan $g(d, t) = 5$.
Karena untuk setiap *path* berarah $s-t$ pada Gambar 4.3, yaitu $P_1 : s = v_0, a, b, v_3 = t$; $P_2 : s = v_0, a, d, v_3 = t$; $P_3 : s = v_0, c, b, v_3 = t$ dan $P_4 : s = v_0, c, d, v_3 = t$ di *layered network*, terdapat sekurang-kurangnya 1 (satu) jalur $e_i = (v_{i-1}, v_i)$ dengan $g(e_i)$ seperti didefinisikan di atas, maka g merupakan *blocking flow*, seperti gambar di bawah ini.



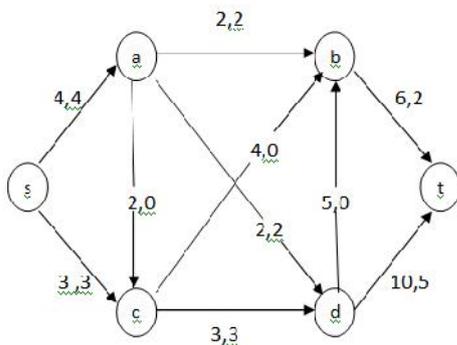
Gambar 4 *Blocking flow*

Angka yang tertera pada setiap jalur pada Gambar 4 menyatakan arus $g(e) =$ kapasitas sisa $c_f(e)$ di N_f .

Langkah 5. Berdasarkan *blocking flow* g (langkah 4), maka dapat dikonstruksi suatu arus maksimum f' sebagai berikut :

- Karena (s, a) adalah jalur maju, maka $f'(s, a) = f(s, a) + g(s, a) = 0 + 4 = 4$
- Karena (s, c) adalah jalur maju, maka $f'(s, c) = 3$
- Karena (a, b) adalah jalur maju, maka $f'(a, b) = 2$
- Karena (a, d) adalah jalur maju, maka $f'(a, d) = 2$
- Karena (b, t) adalah jalur maju, maka $f'(b, t) = 2$
- Karena (c, b) adalah jalur maju, maka $f'(c, b) = 0$
- Karena (c, d) adalah jalur maju, maka $f'(c, d) = 3$
- Karena (d, t) adalah jalur maju, maka $f'(d, t) = 5$
- Karena (u, v) adalah bukan jalur mundur, maka tidak ada $f'(v, u)$
- Untuk semua jalur (u, v) yang lain di N , maka $f'(u, v) = f(u, v)$, yaitu :
 - (i). Untuk jalur (a, c) di N , maka $f'(a, c) = f(a, c) = 0$.
 - (ii). Untuk jalur (d, b) di N , maka $f'(d, b) = f(d, b) = 0$.

Sehingga didapatkan jaringan $N = (V, E)$ dengan arus maksimum seperti di bawah ini.



Gambar 5 Jaringan dengan arus maksimum

Untuk memeriksa apakah pada Langkah 5 tersebut arus f' adalah arus maksimum, maka ulangi langkah-langkah

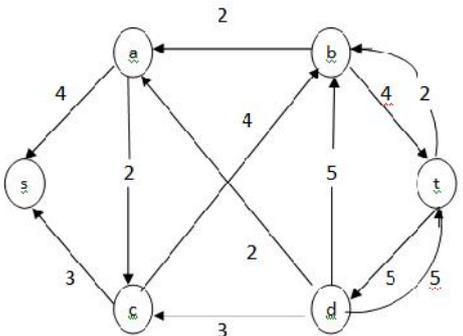
Algoritma Dinic tersebut dari Langkah 2 dan seterusnya sampai didapatkan bahwa arus f' adalah arus maksimum.

Langkah 2. Konstruksi jaringan sisa $N_f = (V, E_f)$ relatif terhadap arus f' di N .

$N_f = (V, E_f)$ dikatakan suatu jaringan sisa relatif terhadap arus f' jika dan hanya jika:

1. Himpunan simpul V di N_f sama dengan himpunan simpul V di N , yaitu $V = \{s, a, b, c, d, t\}$, dan
2. Untuk setiap jalur $e = (u, v)$ di N , maka N_f mempunyai :
 - (i). Untuk jalur $e_1 = (s, a) \in E$:
 Karena $f(e_1) = c(e_1)$, maka tidak terdapat jalur $e_1' = (s, a) \in E_f$
 Karena $f(e_1) > 0$, maka terdapat jalur $e_1' = (a, s) \in E_f$ dengan $c_f(e_1') = f(e_1) = 4$
 - (ii). Untuk jalur $e_2 = (s, c) \in E$:
 Karena $f(e_2) = c(e_2)$, maka tidak terdapat jalur $e_2' = (s, c) \in E_f$
 Karena $f(e_2) > 0$, maka terdapat jalur $e_2' = (c, s) \in E_f$, $c_f(e_2') = 3$
 - (iii). Untuk jalur $e_3 = (c, d) \in E$:
 Karena $f(e_3) = c(e_3)$, maka tidak terdapat jalur $e_3' = (c, d) \in E_f$
 Karena $f(e_3) > 0$, maka terdapat jalur $e_3' = (d, c) \in E_f$ dengan $c_f(e_3') = 3$
 - (iv). Untuk jalur $e_4 = (a, d) \in E$:
 Karena $f(e_4) = c(e_4)$, maka tidak terdapat jalur $e_4' = (a, d) \in E_f$
 Karena $f(e_4) > 0$, maka terdapat jalur $e_4' = (d, a) \in E_f$ dengan $c_f(e_4') = 2$
 - (v). Untuk jalur $e_5 = (a, c) \in E$:
 Karena $f(e_5) < c(e_5)$, maka terdapat jalur $e_5' = (a, c) \in E_f$ dengan $c_f(e_5') = 2$
 Karena $f(e_5) = 0$, maka tidak terdapat jalur $e_5' = (c, a) \in E_f$
 - (vi). Untuk jalur $e_6 = (c, b) \in E$:
 Karena $f(e_6) < c(e_6)$, maka terdapat jalur $e_6' = (c, b) \in E_f$ dengan $c_f(e_6') = 4$

- Karena $f(e_6) > 0$, maka terdapat jalur $e_6' = (b, c) \in E_f$ dengan $c_f(e_6') = 2$
- (vii). Untuk jalur $e_7 = (a, b) \in E$:
 Karena $f(e_7) = c(e_7)$, maka tidak terdapat jalur $e_7' = (a, b) \in E_f$
 Karena $f(e_7) > 0$, maka terdapat jalur $e_7' = (b, a) \in E_f$ dengan $c_f(e_7') = 2$
- (viii). Untuk jalur $e_8 = (b, t) \in E$:
 Karena $f(e_8) < c(e_8)$, maka terdapat jalur $e_8' = (b, t) \in E_f$ dengan $c_f(e_8') = 4$
 Karena $f(e_8) > 0$, maka terdapat jalur $e_8' = (t, b) \in E_f$ dengan $c_f(e_8') = 2$
- (ix). Untuk jalur $e_9 = (d, b) \in E$:
 Karena $f(e_9) < c(e_9)$, maka terdapat jalur $e_9' = (d, b) \in E_f$ dengan $c_f(e_9') = 5$
 Karena $f(e_9) = 0$, maka tidak terdapat jalur $e_4' = (b, c) \in E_f$
- (x). Untuk jalur $e_{10} = (d, t) \in E$:
 Karena $f(e_{10}) < c(e_{10})$, maka terdapat jalur $e_{10}' = (d, t) \in E_f$ dengan $c_f(e_{10}') = 5$
 Karena $f(e_{10}) > 0$, maka terdapat jalur $e_{10}' = (t, d) \in E_f$ dengan $c_f(e_{10}') = 5$



Gambar 6. Jaringan sisa dari Gambar 5

Untuk jaringan sisa $N_f = (V, E_f)$ pada Gambar 6, $V = \{s, a, b, c, d, t\}$, $E_f = \{(a, s), (c, s), (a, c), (b, a), (d, a), (b, t), (c, b), (d, c), (d, b), (d, t), (t, b), (t, d)\}$. Angka yang tertera pada setiap jalur menyatakan kapasitas jalur $c_f(e)$ di N_f . Dalam hal ini jalur-jalur $(a, c), (c, b), (d, b), (b, t)$ dan (d, t) adalah himpunan jalur

maju di N_f dan selainnya adalah himpunan jalur mundur di N_f .

Langkah 3. Konstruksi *layered network* relatif terhadap arus f'

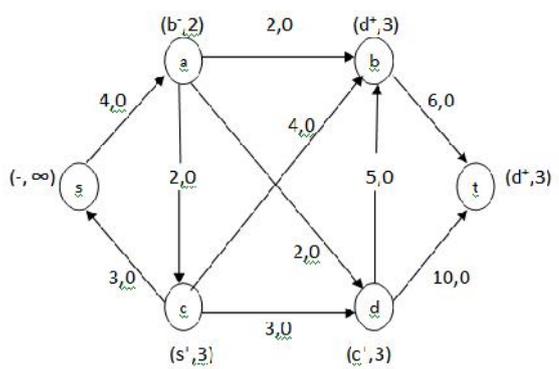
Berdasarkan Algoritma *Layered Network* didapatkan bahwa :

- Langkah 1. $V_0 = \{s\}$ dan $i = 0$
 Langkah 2. $T = \{v \mid v \notin V_0, \text{ dan } (s, v) \in E_f \text{ untuk } s \in V_0\} = \emptyset$
 Langkah 3. Karena $T = \emptyset$, maka arus f' adalah arus maksimum yang didapatkan. Berhenti.

Sehingga dapat disimpulkan bahwa arus f' yang didapatkan pada Gambar 4.5 adalah arus maksimum dengan $val(f') = 7 > val(f) = 0$.

Algoritma Pelabelan Ford-Fulkerson

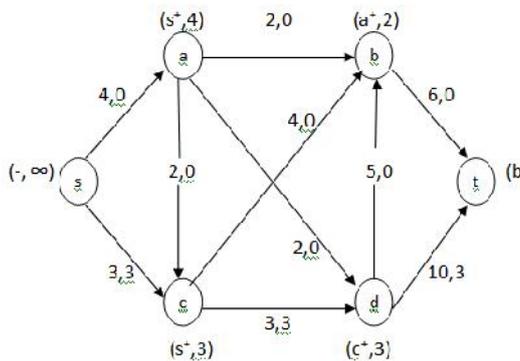
Jar² gan transportasi $N = (V, E)$ dengan kapasitas jalur $c(e)$ dan arus $f(e)$ tertera pada setiap jalur berturut-turut seperti pada gambar berikut:



Gambar 7 Jaringan transportasi dengan arus 0

1. Pilih sembarang arus f di N (boleh dipilih $f(e) = 0$ untuk setiap jalur e di N).
2. Fase 1 dimulai. Labeli s dengan $(-, \infty)$.

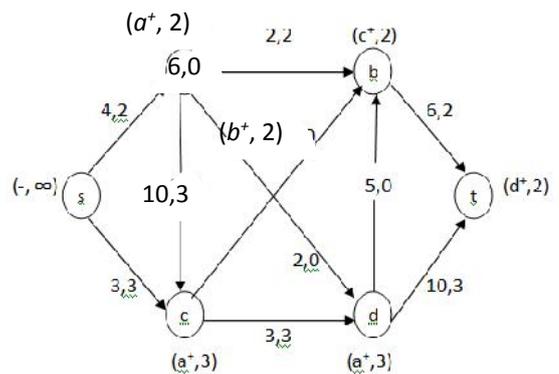
3. Simpul a, b, c, d, t tidak berlabel. Pilih simpul c kemudian beri label $(s^+, 3)$, simpul d diberi label $(c^+, 3)$, simpul a diberi label $(b^-, 2)$, simpul b diberi label $(d^+, 3)$, simpul t diberi label $(d^+, 3)$.
4. Karna $v = t$, maka lanjutkan ke langkah 5. (Fase 1 selesai).
5. Fase 2 dimulai:
 - a. Label dari c adalah (d_c, Δ_c)
Jika $d_c = s^+$, maka $f(s, c) = 3$
 - b. Label dari d adalah (d_d, Δ_d)
Jika $d_d = c^+$, maka $f(c, d) = 3$
 - c. Label dari a adalah (d_a, Δ_a)
Jika $d_a = b^-$, maka $f(a, b) = f(a, b) - \Delta_t$
 - d. Label dari b adalah (d_b, Δ_b)
Jika $d_b = d^+$, maka $f(d, b) = f(d, b) + \Delta_t$
 - e. Label dari t adalah (d_t, Δ_t)
Jika $d_t = d^+$, maka $f(d, t) = 3$
6. Karena $u = s$, hapus semua label dan fase 2 selesai dan pergi ke langkah 2.



Gambar 8 Jaringan transportasi dengan arus 3

2. Fase 1 : Labeli s dengan $(-, \infty)$.
3. Simpul a, b, c, d, t tidak berlabel. Pilih simpul a kemudian beri label $(s^+, 4)$,

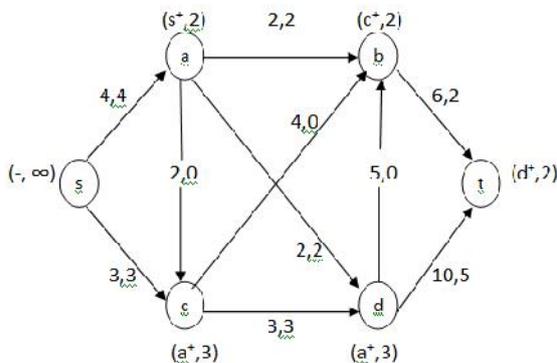
4. Karna $v = t$, maka lanjutkan ke langkah 5. (Fase 1 selesai).
5. Fase 2 dimulai :
 - a. Label dari a adalah (d_a, Δ_a)
Jika $d_a = s^+$, maka $f(s, a) = 2$
 - b. Label dari b adalah (d_b, Δ_b)
Jika $d_b = a^+$, maka $f(a, b) = 2$
 - c. Label dari t adalah (d_t, Δ_t)
Jika $d_t = b^+$, maka $f(b, t) = 2$
 - d. Label dari c adalah (d_c, Δ_c)
Jika $d_c = a^+$, maka $f(a, c) = f(a, c) + \Delta_t$
 - e. Label dari d adalah (d_d, Δ_d)
Jika $d_d = a^+$, maka $f(a, d) = f(a, d) + \Delta_t$
6. Karena $u = s$, hapus semua label dan fase 2 selesai dan pergi ke langkah 2.



Gambar 9 Jaringan transportasi dengan arus 5

2. Fase 1 dimulai. Labeli s dengan $(-, \infty)$.
3. Simpul a, b, c, d, t tidak berlabel. Pilih simpul a kemudian beri label $(s^+, 2)$, simpul d diberi label $(a^+, 2)$, simpul c diberi label $(a^+, 2)$, simpul b

- diberi label $(c^+, 2)$, simpul t diberi label $(d^+, 2)$.
4. Karena $v = t$, maka lanjutkan ke langkah 5. (Fase 1 selesai).
 5. Fase 2 dimulai :
 - a. Label dari a adalah (d_a, Δ_a)
Jika $d_a = s^+$, maka $f(s, a) = 4$
 - b. Label dari d adalah (d_d, Δ_d)
Jika $d_d = a^+$, maka $f(a, d) = 2$
 - c. Label dari c adalah (d_c, Δ_c)
Jika $d_c = a^+$, maka $f(a, c) = f(a, c) + \Delta_t$
 - d. Label dari b adalah (d_b, Δ_b)
Jika $d_b = c^+$, maka $f(c, b) = f(c, b) + \Delta_t$
 - e. Label dari t adalah (d_t, Δ_t)
Jika $d_t = d^+$, maka $f(d, t) = 5$
 6. Karena $u = s$, hapus semua label dan fase 2 selesai dan pergi ke langkah 2.



Gambar 10 Arus maksimum yang diperoleh dari Algoritma *Labeling*

2. Fase 1 dimulai. Labeli s dengan $(-, \infty)$.
3. Karena tidak terdapat simpul tidak berlabel, maka lanjut ke langkah 7.
7. Arus f adalah arus maksimum. Berhenti.

Karena arus yang keluar dari simpul s sama besarnya dengan kapasitas

jalur yang berasal dari s dan tidak ditemukan *augmenting path* yang relatif terhadap arus f_3 , maka pengerjaan selesai.

Gambar di atas adalah ilustrasi lengkap dari langkah-langkah algoritma pelabelan Ford-Fulkerson. Jadi, suatu arus maksimum yang didapatkan adalah sebesar

Membandingkan Algoritma Dinic dengan Algoritma Pelabelan Ford-Fulkerson

Algoritma Dinic dan algoritma pelabelan Ford-Fulkerson memberikan hasil arus maksimum yang sama yaitu sebesar 7. Namun, kedua algoritma tersebut memiliki kelebihan dan kekurangan masing-masing. Algoritma Dinic lebih baik digunakan pada jaringan yang memiliki jalur berjumlah genap dan berpola. Pada jaringan yang setiap jalurnya memiliki arus sebesar 0 (nol), langkah 2 diganti setelah langkah 5 karena tidak terlalu berpengaruh dalam menemukan arus maksimum. Algoritma pelabelan Ford-Fulkerson lebih baik digunakan pada jaringan yang memiliki lebih sedikit jalur.

Algoritma pelabelan Ford-Fulkerson lebih mangkus dibandingkan dengan algoritma Dinic karena algoritma pelabelan Ford-Fulkerson membutuhkan waktu yang lebih singkat dalam mencari penyelesaian masalah arus maksimum dibanding algoritma Dinic. Selain itu, algoritma pelabelan Ford-Fulkerson membutuhkan lebih sedikit ruang memori

dibandingkan algoritma Dinic. Hal ini ditunjukkan dari banyaknya halaman kertas yang digunakan oleh algoritma

pelabelan Ford-Fulkerson dan algoritma Dinic.

KESIMPULAN

1. Algoritma Dinic dan algoritma pelabelan Ford-Fulkerson dapat digunakan untuk menyelesaikan masalah arus maksimum yang dimodelkan dalam suatu jaringan transportasi. Algoritma Dinic ini menggunakan konstruksi jaringan sisa, *layered network* dan konstruksi arus baru (*blocking flow*). Algoritma pelabelan Ford-Fulkerson menggunakan 2 Fase. Fase pertama yaitu melabeli s dengan $(-,)$ dan semua simpul lainnya. Fase kedua yaitu menghitung arus yang mengalir

pada jalur. Algoritma Dinic dan algoritma pelabelan Ford-Fulkerson memberikan hasil arus maksimum yang sama yaitu sebanyak 7.

2. Algoritma pelabelan Ford-Fulkerson lebih mangkus dibandingkan dengan algoritma Dinic karena algoritma pelabelan Ford-Fulkerson membutuhkan waktu yang lebih singkat dan ruang memori yang lebih sedikit dalam mencari penyelesaian masalah arus maksimum dibandingkan dengan algoritma Dinic.

SARAN

Untuk penelitian selanjutnya disarankan adanya tindak lanjut pengerjaan untuk masalah arus maksimum seperti algoritma pelabelan Edmonds dan Karp, algoritma MPM (Malhotra, Pramodh Kumar & Maheswari), algoritma Goldberg-Tarjan untuk masalah arus maksimum algoritma yang tidak dibahas

sehingga diketahui algoritma mana yang lebih mangkus ataupun penerapan algoritma dalam menyelesaikan masalah yang berhubungan dengan arus maksimum seperti masalah transportasi barang dari lokasi sumber ke lokasi tujuan di dunia nyata.

DAFTAR PUSTAKA

- Ahuja, R. K. dan Orlin, J. B., (1989), A Fast and Simple Algorithm for the Maximum Flow Problem, *JSTOR Operation Research* **37**: 748-759.
- Ford, L. R. dan Fulkerson D. R., (1956), Maximal flow through a network, *Canadian Journal of Mathematic* **8**: 399-404.
- Foulds, L. R., (1984), *Combinatorial Optimization for Undergraduates*, Springer-Verlag, New York.
- Handoyo, A. B., (2011), Makalah IF2091, *Aplikasi Algoritma Network Flow untuk Manajemen Pendistribusian Minyak*.
- Hillier, F. S. dan Lieberman, G. J., (1994), *Pengantar Riset Operasi*, Jilid 1, Edisi ke-5, Terjemahan Gunawan, E., dkk., Erlangga, Indonesia.
- Hillier, F. S. dan Lieberman, G. J., (2005), *Introduction Operations Research*, Eight Edition, McGraw-Hill Companies, Inc., Singapore.
- Johnsonbaugh, R., (2002), *Matematika Diskrit* : Edisi ke-4, Terjemahan Djunaedi, D., dkk., PT. Prenhallindo, Jakarta.
- Mehlhorn, K., (2000), *The Maximum Flow Problem*. <http://www.mpi-sb.mpg.de/~mehlhorn/DatAlg/Maxflow.pdf>.
- Munir, R., (2007), *Matematika Diskrit*, Edisi ke-3, Informatika, Bandung.
- Nurhayati, O. D., (2013), *Algoritma dan Pemrograman Kompleksitas Algoritma*. Undip, Semarang.
- Siahaan, S., (2010), *Matematika Diskrit I*, FMIPA Unimed, Medan.
- Siang, J. J., (2006), *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*, Andi Offset, Yogyakarta.
- Strang, G., (1986), *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, United States of America.
- Taha, H. A., (2007), *Riset Operasi*, Jilid 1, Terjemahan Wirajaya D., dan Saputra, L., Binarupa Aksara, Tangerang.
- Thulasiraman, K. dan Swamy M. N. S., (1992), *Graphs: Theory and Algorithms*, John Wiley & Sons, Concordia University Montreal, Canada.
- Vatter, V., (2004), *Graphs, Flows, and the Ford Fulkerson Algorithm*. <http://www.math.ufl.edu/~vatter/teaching/summer04/flow.pdf>
- Vidyamurthy, G., (2004), *Pairs Trading Quantitative Methods and Analysis*, John Wiley & Sons, Inc., New Jersey.
- Yudhianto, A., (2003), *Algoritma Dinic untuk Masalah Arus Maksimum*, Skripsi, FMIPA, IPB, Bogor.