UJM 2 (2) (2013)
UNNES Journal of Mathematics
http://journal.unnes.ac.id/sju/index.php/ujm

MENENTUKAN ALIRAN MAKSIMUM DENGAN ALGORITMA FORD-FULKERSON DAN *PREFLOW-PUSH*

Rif'ah Ulya✉**, Mulyono, Amin Suyitno**

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia
Gedung D7 lantai 1 Kampus Sekaran, Gunungpati, Semarang, 50229

Info Artikel

Sejarah Artikel:
Diterima Agustus 2013
Disetujui September 2013
Dipublikasikan Nopember 2013

Keywords :
Algoritma Ford-Fulkerson; Algoritma Preflow-Push; dan Aliran Maksimum.

Abstrak

Penelitian ini bertujuan untuk mengetahui konsep aliran maksimum berdasarkan teorema *Maximal Flow–Minimal Cut* serta mengetahui cara menentukan aliran maksimum dengan algoritma Ford-Fulkerson dan *Preflow-Push*. Metode penelitian yang digunakan adalah metode studi pustaka. Pada penelitian ini dapat disimpulkan: (1) konsep aliran maksimum berdasarkan teorema Maximal Flow–Minimal Cut menjelaskan bahwa nilai aliran $f^{*}=c(X,X_1)$ dengan $B(X,X_1)$ merupakan sebuah pemutus-(s,t) minimum di N, maka f^{*} adalah aliran maksimum di N yang nilainya selalu sama dengan kapasitas pemutus -(s,t) minimum di N; (2) algoritma Ford-Fulkerson bekerja dengan mengkonstruksi aliran baru dengan nilai yang lebih besar dari aliran yang lama, dan menggunakan teknik pelabelan Routin, pencarian aliran baru akan berhenti ketika semua titik N yang terlabel telah teramati dan titik t tidak terlabel; (3) algoritma *Preflow-Push* bekerja dengan operasi dasar *push* dan *relabel*, algoritma ini berhenti ketika tidak ada lagi titik yang aktif. Dalam penelitian ini algoritma Ford-Fulkerson dihitung secara manual, sedangkan algoritma *Preflow-Push* menggunakan alat bantu yaitu *software* GIDEN. Dari contoh penggunaan aliran maksimum dalam penelitian ini diperoleh aliran maksimum = pemutus -(s,t) minimum = 600.

Abstract

The purposes of this research were to know the maximum flow concept based of maximal flow-minimal cut theorem and knew how to determine the maximal flow with Ford-Fulkerson algorithm and Preflow-Push. Observational method that is utilized is studi's method library. This research gets to be concluded: (1) maximal flow concept based the maximal flow-minimum flow theorem explained that flow value $f^{*}=c(X,X_1)$ with $B(X,X_1)$ are a minimum cut-(s,t) in N, so f^{*} is maximal flow in N; (2) the Ford-Fulkerson algorithm worked by constructed new flow with greater value than the flow of time, and used the technique of labelling Routin, search flow will stop when the all N vertices are labeled has been observed and vertex t unlabeled; (2) Preflow-Push algorithm worked with the basic operations push and relabel, this algorithm will stopped when nothing active vertex. In this research the Ford-Fulkerson algorithm manually counted, while Preflow-Push algorithm used the GIDEN software tool. From the example of used the maximal flow in this research obtained maximum flow = minimal cut-(s,t) = 600.

✉ Alamat korespondensi:
rifahulya_4111409008@yahoo.co.id

ISSN 2252-6943

Pendahuluan

Teori graf merupakan topik yang banyak mendapat perhatian, karena model-modelnya sangat berguna untuk aplikasi yang luas, seperti masalah dalam jaringan komunikasi, transportasi, ilmu komputer, dan lain sebagainya. Secara umum, graf merupakan suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Dalam kehidupan sehari-hari, graf digunakan untuk menggambarkan berbagai macam struktur yang ada. Tujuannya adalah sebagai visualisasi objek-objek agar lebih mudah dimengerti. Beberapa contoh graf yang sering dijumpai dalam kehidupan sehari-hari antara lain: struktur organisasi, bagan alir pengambilan mata kuliah, peta, rangkaian listrik, dan lain-lain (Siang, 2004: 185).

Jaringan merupakan salah satu kajian dalam teori graf. Jaringan adalah kumpulan titik dan sisi yang saling terhubung dengan arah tertentu. Di dalam jaringan terdapat beberapa model yang bisa digunakan untuk membantu memecahkan masalah-masalah, di antaranya adalah model rentang jaringan minimum, model rute terpendek, dan model aliran maksimum.

Model aliran maksimum mempunyai tujuan untuk memaksimalkan jumlah aliran yang melewati jaringan dalam sebuah sistem jaringan. Pada model masalah rute terpendek, aliran yang membangkitkan biaya tidak dibatasi oleh kapasitas apa pun. Sebaliknya, pada model masalah aliran maksimum, aliran tersebut tidak membangkitkan biaya tetapi mempunyai batas kapasitas tertentu.

Pada sebuah jaringan dalam masalah aliran maksimum, selalu terdapat sebuah aliran yang nilainya sama dengan kapasitas pemutus minimum (*minimal cut*), yang dikenal dengan sebutan "*Teorema Maximal Flow–Minimal Cut*" (Budayasa, 2007: 235). Teorema *Maximal Flow-Minimal Cut* menjamin bahwa aliran maksimum pada sebuah jaringan tidak akan melebihi kapasitas pemutus minimum dalam jaringan tersebut. Dalam pencarian aliran maksimum, terdapat beberapa algoritma, menurut Ahuja *et al*. (1993: 166-167), algoritma yang digunakan dalam menyelesaikan masalah aliran maksimum secara umum menggunakan dua pendekatan dasar, yaitu pendekatan algoritma *Aughmenting Path* dan pendekatan algoritma *Preflow-Push*.

Algoritma yang menggunakan pendekatan *Aughmenting Path* diantaranya adalah algoritma Ford-Fulkerson. Algoritma Ford-Fulkerson ditemukan oleh Ford dan Fulkerson pada tahun 1956. Menurut Vatter (2004: 7), cara berjalan algoritma Ford-Fulkerson sangat alami, karena algoritma ini boleh dimulai dengan tidak adanya aliran pada semua busur (f_0=0). Kemudian ditemukanlah *Aughmenting Path* dari titik sumber s ke titik tujuan t pada lintasan tersebut. Algoritma ini akan efektif bagi penggunanya untuk melakukan suatu proses, tindakan atau pengambilan keputusan untuk tujuan tertentu dengan mengetahui aliran maksimum yang terdapat dalam suatu jaringan. Pada penelitian ini, algoritma *Preflow-Push* digunakan hanya sebatas untuk mencocokkan hasil perhitungan manual dari algoritma Ford-Fulkerson. Dalam hal ini, algoritma *Preflow-Push* akan dijalankan dengan alat bantu yaitu *software* GIDEN.

Rumusan masalah dalam penelitian ini adalah (1) bagaimana konsep aliran maksimum berdasarkan teorema *Maximal Flow-Minimal Cut?*; (2) bagaimana menentukan aliran maksimum dengan algoritma Ford-Fulkerson?; (3) bagaimana menentukan aliran maksimum dengan algoritma *Preflow-Push* dengan bantuan *software* GIDEN?

Penelitian ini bertujuan untuk mengetahui konsep aliran maksimum berdasarkan teorema *Maximal Flow–Minimal Cut*, mengetahui cara menentukan aliran maksimum menggunakan algoritma Ford-Fulkerson dan mengetahui cara menentukan aliran maksimum menggunakan algoritma *Preflow-Push* dengan alat bantu *software* GIDEN.

Metode Penelitian

Metode penelitian yang digunakan adalah metode studi pustaka, yaitu melakukan kajian pustaka dari berbagai sumber yang berkaitan dengan permasalahan sehingga didapat suatu ide mengenai bahan dasar pengembangan upaya pemecahan masalah.

Pembahasan

A. Teorema *Maximal Flow-Minimal Cut*

Misalkan N sebuah jaringan dengan titik sumber s dan titik tujuan t. Maka terdapat sebuah aliran maksimum pada N (Budayasa, 2007: 238-239).

Bukti:

Misalkan f sebuah aliran dengan nilai d

dari s ke t pada jaringan N.

Definisikan himpunan X sebagai berikut:

$w \in X$ sedemikian hingga $w = s$ atau ada lintasan (s,w) pada graf dasar N yang inkremennya positif. Maka ada dua kemungkinan letak titik t, yaitu $t \in X$ atau $t \in X_1 = V(N)$-X.

Jika $t \in X$, berdasarkan definisi X, terdapat lintasan P dari titik s ke titik t dengan $i(P) = \delta > 0$. Sehingga berdasarkan lemma 4.2, aliran f dapat direvisi menjadi aliran f_1, sedemikian hingga $f_1(a)=f(a)+\delta$ jika a busur maju pada P, $f_1(a)=f(a)-\delta$ jika a busur balik pada P, $f_1(a)=f(a)$ jika a busur N yang tidak terletak pada P. Aliran f_1 bernilai $d+\delta>$d (karena $\delta>0$). Jadi f_1 adalah aliran dari s ke t di N yang nilainya lebih besar dari nilai aliran f. Dikatakan, aliran f_1 adalah revisi aliran f. Selanjutnya menggunakan aliran f_1 pada N, cari himpunan X seperti sebelumnya. Jika titik tujuan t menjadi anggota X, maka berdasarkan definisi X, terdapat lintasan P_1 dari s ke t dengan $i(P_1) = \delta_1 > 0$. Berdasarkan lemma 4.2, bentuk aliran f_2 dari f_1 sedemikian hingga: $f_2(a) = f_1(a) + \delta_1$ jika a busur maju pada P_1, $f_2(a)=f_1(a)-\delta_1$ jika a busur balik pada P_1, dan $f_2(a)=f_1(a)$ untuk busur a yang lainnya. Aliran f_2 bernilai $d+\delta+\delta_1$, lebih besar dari nilai aliran f_1 karena $\delta_1>0$. Jadi aliran f_2 merupakan revisi dari aliran f_1 didasarkan atas lintasan peningkatan P_1. Proses merevisi aliran seperti itu dapat dilanjutkan sampai diperoleh suatu aliran, katakan aliran f^*, sedemikian hingga terhadap aliran f^* pada N, himpunan X tidak memuat titik t, atau $t \notin X$. Ini berarti, tidak ada lagi lintasan pada graf N dari s ke t yang inkremennya positif. Selanjutnya, akan ditunjukkan f^* adalah aliran maksimum pada N. Untuk itu cukup ditunjukkan bahwa nilai aliran f^* sama dengan kapasitas sebuah pemutus-(s,t) pada N.

Klaim 1. Jika $v \in X$, $\in u\ X_1 = V(N)$-X, dan $(v,u) \in\ (N)$, maka $f^*(v,u)=c(v,u)$.

Andaikan $f^*(v,u)<c(v,u)$. Karena (v,u) busur maju, maka $i(v,u)>0$. Selanjutnya, karena $v \in X$, maka ada lintasan P' dari s ke v dengan $i(P')>0$ dan karena $i(v,u)>0$, maka ada lintasan dari s ke u lewat v yang inkremennya positif. Berdasarkan definisi X, maka $u \in X$ kontradiksi dengan $u \notin X$.

Klaim 2. Jika $v \in X$, $u \in X_1 = V(N)$-X, dan $(v,u) \in \Gamma(N)$, maka $f^*(u,v)=0$.

Andaikan $f^*(u,v) > 0$. Karena (u,v) busur balik maka $i(u,v)=f^*(u,v) > 0$. Seperti sebelumnya, karena $v \in X$, maka ada lintasan P' dari ske v dengan $i(P')>0$ dan karena $i(u,v)>0$, maka pada graf dasar N ada lintasan dari s ke u lewat v yang inkremennya positif. Berdasarkan definisi X, maka $u \in X$ kontradiksi bahwa titik u terletak di dalam $X_1 = V(N)$ -X.

Berdasarkan klaim 1 dan klaim 2, secara berturut-turut diperoleh

$f^*(X,X_1)=c(X,X_1)$ dan $f^*(X,X_1)=0$.

Berdasarkan teorema 4.1

Nilai aliran $f^*=f^*(X,X_1)$-$f^*(X_1,X)$

$=c(X,X_1)-0=c(X,X_1)$.

Karena $B(X,X_1)$ sebuah pemutus-(s,t) pada N dan nilai $f^*=c(X,X_1)$, maka f^* adalah aliran maksimum di N dan $B(X,X_1)$ adalah sebuah pemutus-(s,t) minimum pada jaringan N.

B. Algoritma Ford-Fulkerson

Secara sistematis algoritmanya adalah sebagai berikut.

Langkah 1: misalkan f sebuah aliran dari s ke t pada N. (Boleh dimulai dengan aliran bernilai nol, yaitu $f(i,j)=0, \forall (i,j) \in \Gamma(N)$. Dilanjutkan ke Routin-Pelabelan.

Langkah 2: Routin-Pelabelan

2.1 Label $v_s=(s,+,\varepsilon(s)=\sim)$. Titik v_s telah terlabel dan belum teramati. Sebuah titik v dikatakan telah teramati jika semua titik yang dapat dilabel dari titik v sudah terlabel.

2.2 Pilih sebarang titik yang terlabel tetapi belum teramati, misalkan titik tersebut v_x. Untuk $\forall v_y\ \exists (y,x) \in \Gamma(N)$, v_y belum berlabel dan $f(y,x)>0$, maka label $v_y=(x,-,\varepsilon(y))$ dengan $\varepsilon(y)=min\{\varepsilon(x),f(y,x)\}$. Sekarang titik v_y telah terlabel, tetapi belum teramati. Untuk $\forall v_y\ \exists (x,y),\ \in \Gamma v_y$ belum berlabel dan $c(x,y)>f(x,y)$, maka label $v_y=(x,+,\varepsilon(y))$ dengan $\varepsilon(y)= min\{\varepsilon(x),c(x,y)-f(x,y)\}$. Sekarang titik v_y terlabel tetapi belum teramati, sedangkan titik v_x telah terlabel dan teramati.

2.3 Ulangi langkah 2.2 sampai:

(1) titik v_t terlabel, atau;

(2) semua titik terlabel telah teramati tetapi titik v_t tak terlabel;

(3) jika titik v_t terlabel, lanjut ke langkah 3;

(4) jika semua titik terlabel telah teramati tetapi titik v_t tak terlabel, maka BERHENTI. Aliran f adalah aliran maksimum pada jaringan N.

Langkah 3: dengan prosedur "balik", temukan lintasan peningkatan *P* dengan *i*(*P*) adalah label v_t.

Langkah 4: tingkatkan nilai aliran *f* sebesar label v_t, berdasarkan lintasan peningkatan *P* dengan menggunakan "Routine-Peningkatan":

4.1 Misal: z=t lanjutkan ke langkah 4.2.

4.2 Jika label $v_z=(q,+,\varepsilon(t))$ tingkatkan nilai $f(q,z)$ dengan $\varepsilon(t) = i(P)$. Jika label $v_z=(q,-,\varepsilon(t))$ turunkan nilai $f(z,q)$ dengan $\varepsilon(t)=i(P)$.

Jika $q=s$, hapus semua label. Pada tahap ini diperoleh aliran *f* baru dengan nilai = $i(P)$ + nilai aliran *f* lama. Ganti aliran *f* dengan aliran *f* yang baru, dan kembali ke langkah 1 (Budayasa, 2007: 240-242).

C. Algoritma *Preflow-Push*

Algoritma *Preflow-Push* dimulai dengan *preflow f* yang nilainya sama dengan kapasitas busur untuk setiap busur yang meninggalkan titik sumber *s* dan bernilai nol untuk yang lainnya. Selanjutnya inisialisasi label dengan pelabelan valid $d(s)=n$, $d(t)=0$, dan $d(v)\leq d(w)+1$ untuk setiap busur sisa (*v,w*). Algoritma *Preflow-Push* secara berulang-ulang menggunakan dua operasi dasar, yaitu *Push* dan *Relabel* yang bekerja sebagai berikut.

Push (*v,w*)

Applicability. v is active, $(v,w)\in E_f$ *and* $d(v)=d(w)+1$.

Action. Set $\delta=\min\{e(v),c_f(v,w)\}$ *and do the following.*

1. Increase $f(v,w)$ *by* δ *if (v,w) is a forward edge, otherwise decrease* $f(w,v)$ *by* δ.

2. Decrease $e(v)$ *by* δ *and increase* $e(w)$ *by* δ.

(Note: $\delta>0$ *because both* $e(v)$ *and* $c_f(v,w)$ *are positive).*

Relabel (*v*)

Applicability. v is active, and for every (v,w) E_f, $d(v)\leq d(w)$.

Action. Set $d(v)=\min_{(v,w)\in E_f}\{d(w)+1\}$ (Thulasiraman, 1992: 412).

Misalkan titik *v* (bukan titik sumber maupun titik tujuan) yang aktif ($e(v)>0$), maka pilih titik tersebut dan lakukan *push* dan *relabel* secara berulang sebagai berikut.

Langkah 1: jika ada busur (*v,w*) yang admissible $d(v)=d(w)+1$, maka lakukan push $\delta=\min\{e(v),c_f(v,w)\}$

(1) tingkatkan aliran $f(v,w)$ sebesar δ jika (*v,w*) busur maju, dan penurunan aliran $f(w,v)$ sebesar δ jika (*w,v*) busur balik,

(2) turunkan $e(v)$ sebesar δ dan tingkatkan $e(w)$ sebesar δ, dengan $\delta>0$.

Langkah 2: jika tidak ada busur (*v,w*) yang *admissible* $d(v)\leq d(w)$, maka lakukan *relabel*, dengan mengganti $d(v)$ dengan label jarak sebesar $d(v)=\min\{d(w)+1\}$.

Lakukan *push* dan *relabel* secara berulang sehingga tidak ada lagi titik yang aktif.

Pendorongan *preflow f* dari *v* ke *w* meningkatkan aliran $f(v,w)$ dan $e(w)$ dengan peningkatan sebesar $\delta=\min\{e(v),c_f(v,w)\}$, dan penurunan $f(w,v)$ dan $e(v)$ dengan nilai δ yang sama. Setelah dilakukan pendorongan *preflow f* dari *v* ke *w*, jika $c_f(v,w)=0$ dikatakan *f* jenuh (*f saturated*), selainnya dikatakan *f* tak jenuh (*f unsaturated*). Algoritma *Preflow-Push* akan berhenti ketika tidak ada lagi titik aktif.

D. Algoritma *Preflow-Push* dengan *Software* GIDEN

Langkah-langkah penggunaan *software* GIDEN untuk menyelesaikan aliran maksimum menggunakan algoritma *Preflow-Push*, adalah sebagai berikut:

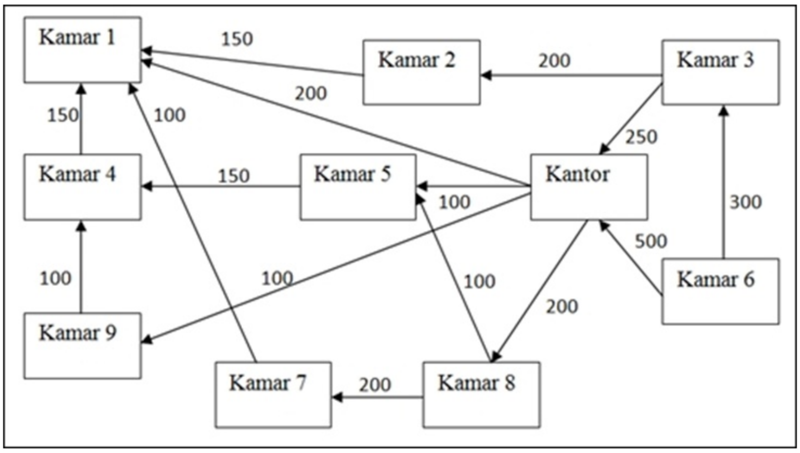
1. Buka *software* GIDEN, klik *file*, pilih *new*, kemudian gambar jaringan serta input bobot kapasitasnya.

2. Untuk mencari aliran maksimumnya, klik *solvers*, pilih *maximum flow*, pilih *Preflow-Push*.

3. Klik *trace* berulang kali untuk melihat iterasi pencarian aliran maksimum. Nilai akhir aliran maksimum akan terlihat ketika tombol *trace* berubah menjadi *reset*.

E. Contoh Penggunaan Aliran Maksimum

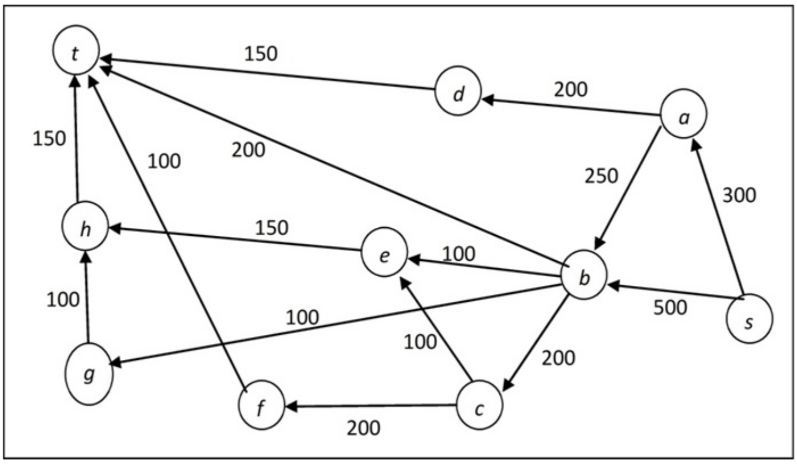
Pembangunan motel, akan dibangun sistem aliran air yang tandon airnya terletak di kamar 6 dan berakhir di kamar 1. Besarnya ukuran pipa berbeda-beda dan kapasitas aliran air (liter per menit) terlihat pada gambar berikut (Dwijanto, 2008: 148).



Gambar 1. Kapasitas aliran air (liter per menit).

Dari Gambar 1 di atas, diasumsikan bahwa kantor maupun kamar selain kamar 1 sedang dalam keadaan tidak menggunakan air, antar kamar letaknya datar, kekuatan gaya yang diberikan dalam pipa sama. Tentukan besarnya aliran air maksimum dan besar kapasitas pemutus minimumnya dengan titik tujuan kamar 1 dalam sistem jaringan aliran air pada motel ini. Gunakan algoritma Ford-Fulkerson dan *Preflow-Push*.

Misalkan, kamar 6 beri nama titik S, kamar 3 beri nama titik a, kantor beri nama titk b, kamar 8 beri nama titik c, kamar 2 beri nama titik d, kamar 5 beri nama titik e, kamar 7 beri nama titik f, kamar 9 beri nama titik g, kamar 4 beri nama titik h, dan kamar 1 adalah titik tujuan beri nama titik t. Sehingga diperoleh gambar graf N sebagai berikut.



Gambar 2. Jaringan N dengan titik sumber s dan titik tujuan t.

Penyelesaian dengan algoritma Ford-Fulkerson:

Iterasi ke 1: dimulai dengan f_0=0, dengan pelabelan Routin diperoleh lintasan peningkatan $P=(s,a,d,t)$ dengan $i(P) = \varepsilon(t)$= 150 , kemudian terapkan Routin peningkatan pada busur-busur $(s,a),(a,d),(d,t)$. Diperoleh aliran baru dengan $f_1=f_0+i(P)$=0+150=150.

Iterasi ke 2: dimulai dengan f_1=150, dengan pelabelan Routin diperoleh lintasan peningkatan $P=(s,b,t)$ dengan $i(P)=\varepsilon(t)$= 200, kemudian terapkan Routin peningkatan pada busur-busur $(s,b),(b,t)$. Diperoleh aliran baru dengan $f_2=f_1+i(P)$=150+200=350.

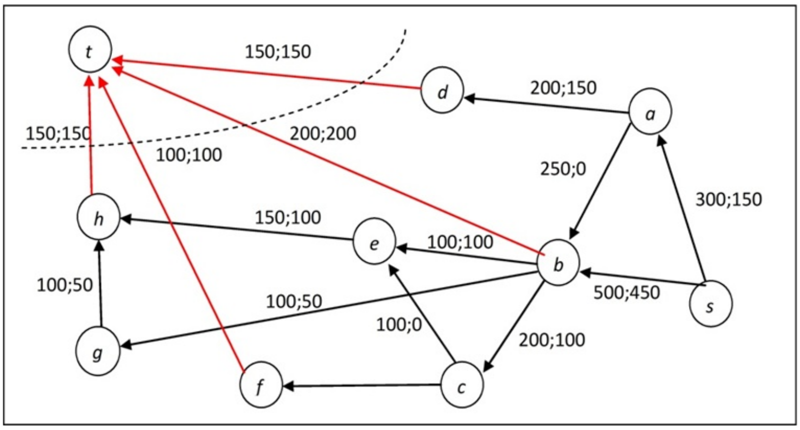
Iterasi ke 3: dimulai dengan f_2=350, dengan pelabelan Routin diperoleh lintasan peningkatan $P=(s,b,c,f,t)$ dengan $i(P)=\varepsilon(t)$= 100, kemudian terapkan Routin peningkatan pada busur-busur $(s,b),(b,c),(c,f),(f,t)$. Diperoleh aliran baru dengan $f_3=f_2+i(P)$=350+100=450.

Iterasi ke 4: dimulai dengan f_3=450, dengan pelabelan Routin diperoleh lintasan peningkatan $P=(s,b,e,h,t)$ dengan $i(P)=\varepsilon(t)$= 100, kemudian terapkan Routin peningkatan pada busur-busur $(s,b),(b,e),(e,h),(h,t)$. Diperoleh aliran baru dengan $f_4=f_3+i(P)$=450+100=550.

Iterasi ke 5: dimulai dengan f_4=550, dengan pelabelan Routin diperoleh lintasan peningkatan *P=(s,b,g,h,t)* dengan *i(P)=*ε *(t)*= 50, kemudian terapkan Routin peningkatan pada busur-busur *(s,b),(b,g),(g,h),(h,t)*. Diperoleh aliran baru dengan $f_5=f_4+i(P)$=550+50=600.

Iterasi ke 6: dimulai dengan f_5=600, dengan pelabelan Routin semua titik yang terlabel telah teramati dan titik *t* tidak terlabel, maka BERHENTI.

Dengan algoritma Ford-Fulkerson diperoleh aliran maksimum sebesar f_5=600.



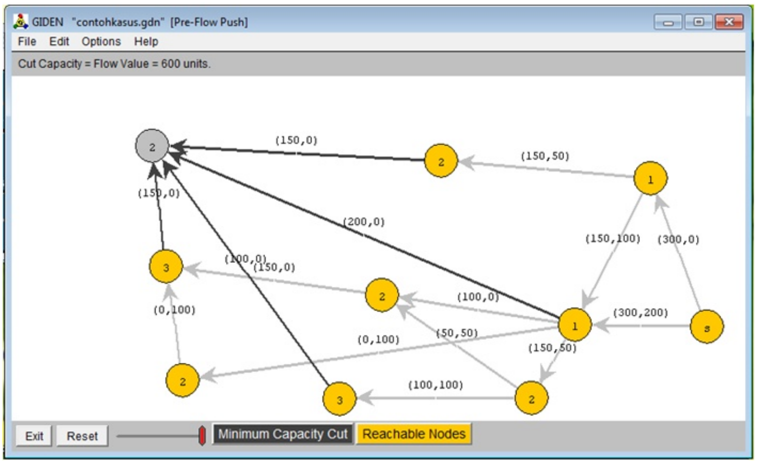
Gambar 3. Jaringan N dengan aliran maksimum = pemutus-(*s,t*) minimum.

$B(X,X_1)=(\{s,a,b,c,d,e,f,g,h\},\{t\})=\{(b,t),(d,t),(f,t),(h,t)\}$ adalah pemutus-(*s,t*) minimum dengan kapasitas

$$c(X,X_1)=c(b,t)+c(d,t)+c(f,t)+c(h,t)$$
$$=200+150+100+150=600$$

Terlihat bahwa nilai aliran baru f_5=c(X,X_1), maka f_5 adalah aliran maksimum dari kamar 6 ke kamar 1 dengan nilai aliran sebesar 600 liter per menit.

Penyelesaian dengan algoritma Preflow-Push pada software GIDEN:



Gambar 4. Hasil aliran maksimum dengan algoritma *Preflow-Push* pada *software* GIDEN.

Algoritma *Preflow-Push* dengan alat bantu *software* GIDEN, diperoleh aliran maksimum = pemutus-(*s,t*) *minimum* = 600. Dengan pemutus-(*s,t*) minimum $B(X,X_1)=(\{s,a,b,c,d,e,f,g,h\},\{t\})=\{(b,t),(d,t),(f,t),(h,t)\}$ sama dengan pemutus-(*s,t*) minimum dari perhitungan manual dengan algoritma Ford-Fulkerson.

Algoritma *Preflow-Push* digunakan untuk mencocokkan perhitungan manual dari algoritma Ford-Fulkerson, dan ternyata hasil yang di dapatkan sama walaupun dengan iterasi yang berbeda. Hal ini menunjukkan bahwa hasil pencarian aliran maksimum dengan algoritma Ford-Fulkerson dan *Preflow-Push* adalah sama.

Simpulan

Dari uraian di atas dapat disimpulkan bahwa konsep aliran maksimum selalu memenuhi teorema Maximal Flow-Minimal Cut yang menjelaskan bahwa nilai aliran $f^{*}=c(X,X_1)$ dengan $B(X,X_1)$ merupakan sebuah pemutus-(s,t) minimum di N, maka f^{*} adalah aliran maksimum di N yang nilainya sama dengan kapasitas pemutus-(s,t) minimum di N. Algoritma Ford-Fulkerson bekerja dengan mengkonstruksi aliran baru dengan nilai yang lebih besar dari aliran yang lama, dan menggunakan teknik pelabelan titik yang pada prinsipnya melabel titik-titik N dengan teknik tertentu dimulai dengan titik s, dengan $(s,+,\sim)$, kemudian dilanjutkan melabeli titik yang lain. Dengan teknik tersebut bisa melabeli titik t, maka dengan teknik prosedur balik lintasan P ditemukan, kemudian tingkatkan lintasan pemingkatan P tersebut sebesar $i(P)=\varepsilon(t)$. Pencarian aliran baru akan berhenti ketika semua titik N yang terlabel telah teramati dan titik t tidak terlabel. Algoritma *Preflow-Push* dengan *software* GIDEN digunakan untuk mencocokkan hasil perhitungan manual dari algoritma Ford-Fulkerson. Algoritma *Preflow-Push* bekerja dengan operasi dasar *push* dan *relabel*, algoritma ini akan berhenti ketika tidak ada lagi titik yang aktif. Dari contoh penggunaan aliran *maksimum* dalam penelitian ini diperoleh aliran *maksimum* = pemutus-(s,t) *minimum* = 600. Hal ini menunjukkan bahwa hasil pencarian aliran *maksimum* dengan algoritma Ford-Fulkerson dan *Preflow-Push* adalah sama.

Ucapan Terimakasih

Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah ikut membantu dalam penyelesaian artikel ini.

Daftar Pustaka

Ahuja, R.K., T.L. Magnanti & J.B. Orlin. 1993. *Network Flows*. America: Prentice-Hall.

Budayasa, I.K. 2007. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.

Dwijanto. 2008. *Program Linear*. Semarang: UNNES Press.

Siang, JJ. 2004. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi Offset.

Thulasiraman, K. & M.N.S. Swamy. 1992. *Graphs Theory and Algorithms*. Canada: Concordia University Montreal.

Vatter, V. 2004. *Graph, Flows, and the Ford-Fulkerson Algorithm*. Tersedia di: http://www.math.ufl.edu/~vatter/teaching/flow.pdf. [diakses 10 maret 2013].