# ALGORITMA FORD-FULKERSON UNTUK MEMAKSIMUMKAN *FLOW* DALAM PENDISTRIBUSIAN BARANG

**[1]Fahrun Nisa', [2]Wahyu Henky Irawan**

[1]Jurusan Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang
[2]jurusan Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang

Email: fahrunnisa33@gmail.com

**ABSTRAK**

Algoritma Ford Fulkerson digunakan untuk mencari *flow* maksimum pada jaringan yang mempunyai satu titik sumber dan satu titik tujuan. Dengan mendefinisikan jalur pendistribusian barang sebagai jaringan, maka jaringan tersebut memiliki beberapa titik sumber dan beberapa titik tujuan. Untuk menyelesaikan permasalahan *flow* maksimum pada jaringan ini maka digunakan modifikasi Algoritma Ford-Fulkerson. Pada artikel ini akan ditunjukkan langkah-langkah mencari *flow* maksimum serta hasil yang diperoleh dengan menggunakan modifikasi Algoritma Ford-Fulkerson pada data simulasi tentang pendistribusian barang dengan lima titik sumber dan lima titik tujuan. Hasil dari penelitian ini merupakan bentuk modifikasi Algoritma Ford-Fulkerson, yaitu membentuk jaringan baru dengan menambahkan satu titik sumber utama dan satu titik tujuan utama pada jaringan baru, membentuk kapasitas di busur dari titik sumber utama ke beberapa titik sumber serta membentuk kapasitas di busur dari beberapa titik tujuan ke titik tujuan utama dengan nilai kapasitas maksimum dan memberi nilai *flow* awal sebesar nol. Selanjutnya memaksimumkan *flow* dari titik sumber utama ke titik tujuan utama menggunakan Algoritma Ford-Fulkerson, yaitu dengan melakukan pelabelan titik, menggunakan prosedur balik, dan mencari lintasan peningkatan sampai semua titik yang terlabel telah teramati dan titik tujuan utama tidak terlabel sehingga iterasi dihentikan. Berdasarkan hasil perhitungan didapatkan *flow* maksimum pada jaringan yang tidak dipartisi, pada jaringan yang dipartisi serta dengan membuat program di Matlab R2010a dengan nilai 45.

**Kata kunci**: Algoritma Ford-Fulkerson, *flow* maksimum, jaringan, jaringan bipartisi

***ABSTRACT***

*Ford-Fulkerson algorithms used to find the maximum flow in a network that has a single point source and a sink point . By defining the goods distribution channels as the network , then the network has several source point and several sink point. To solve the problems of the maximum flow in the network is then used a modified Ford-Fulkerson algorithm. This article can be shown the steps looking for maximum flow and obtained results by using a modified Ford-Fulkerson algorithm on simulated data on the distribution of goods with a five source point and five sink.point. The results of this study is a modified form of the Ford-Fulkerson algorithm , which formed a new network by adding a main source point and a sink point on a major new network ,and formed capacity in an arc from primary sources point to a few source point and established capacity in the bow of some sink point to point the main goal with the value of the maximum capacity and provide initial flow value of zero . Furthermore, to maximize a flow of the main source point to a major sink point by using the Ford - Fulkerson algorithm , namely by labeling a point , using the reverse procedure and to seek increase trajectory till all the labeled points have been observed and a major sink point has not labeled so the iteration is stopped . Based on the results of the calculation, the maximum flow on a unpartitioned network on a partitioned network and by creating a program in Matlab R2010a with a value of 45.*

***Keywords**: Ford-Fulkerson Algorithm, flow maximum, network, bipartite network*

---

## PENDAHULUAN

### 1. Latar Belakang

Teori graf merupakan salah satu cabang dari ilmu matematika, dimana teori-teori yang digunakan tergolong aplikatif dan dapat digunakan untuk memecahkan masalah alam kehidupan sehari-hari. Salah satu pembahasan dalam teori graf adalah jaringan (*network*). Permasalahan dalam suatu jaringan dapat dicari penyelesaiannya yaitu salah satunya dengan mencari *flow* maksimum. Artikel ini membahas tentang pendistribusian barang yang didefinisikan sebagai jaringan.

Dalam konsep pendistribusian barang, semakin banyak barang yang dijual, maka kemungkinan barang yang laku terjual semakin banya, sehingga keuntungan yang didapatkan oleh perusahaan akan semakin banyak. Berdasarkan prinsip ekonomi "Dengan modal sekecil-kecilnya memperoleh keuntungan sebesar-besarnya", maka permasalahan matematika yang sesuai dalam kasus ini yaitu pencarian *flow* maksimum, sehingga *flow* maksimum digunakan dalam jaringan ini.

Sebuah penelitian yang dilakukan oleh [1] yaitu tentang pencarian *flow* maksimum pada jaringan yang memiliki satu titik sumber dan satu titik tujuan menggunakan Algoritma Ford-Fulkerson. Menurut Farizal, algoritma ini selalu memberikan hasil yang benar dalam mencari lintasan peningkatan (*augmenting-path*) sehingga Algoritma Ford-Fulkerson sesuai dengan permasalahan dalam artikel ini.

Dalam penelitian ini akan dicari *flow* maksimum pada jaringan pendistribusian barang, dimana jaringan tersebut memiliki beberapa titik sumber dan beberapa titik tujuan, sehingga akan dilakukan modifikasi Algoritma Ford-Fulkerson. Harapan dari adanya penelitian ini yaitu pendistribusian barang dapat dilakukan secara maksimum sehingga keuntungan yang diperoleh suatu perusahaan akan lebih banyak.

## 2. Metode Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif dengan langkah-langkah penelitiannya:

a. Menyediakan data simulasi yaitu tentang pendistribusian barang.

b. Data simulasi yang disediakan memiliki lima titik sumber dan lima titik tujuan, sehingga dilakukan modifikasi Algoritma Ford-Fulkerson agar algoritma ini dapat digunakan untuk mencari *flow* maksimum pada jaringan.

c. Menyusun *Flow* Maksimum pada jaringan.

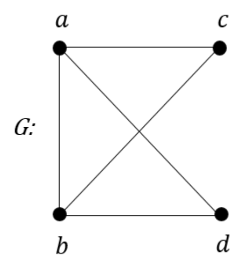
d. Membuat Program di Matlab R2010a.

## KAJIAN TEORI

### 1. Graf

Suatu graf $G$ adalah suatu himpunan tidak kosong dan berhingga dari objek-objek yang disebut sebagai *vertex* (titik) dan himpunan (mungkin kosong) pasangan tak berurutan dari titik-titik berbeda pada $G$ yang disebut *edge* (sisi).

Contoh:




**Gambar 1.** Graf $G$

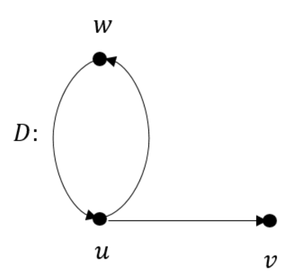
Graf $G$ pada Gambar 1 dapat dinyatakan dengan $G = (V(G), E(G))$ dengan $V(G) = \{a, b, c, d\}$ dan $E(G) = \{ab, ad, ac, bc, cd\}$ *[2]*.

Sebuah sisi graf yang menghubungkan sebuah titik dengan dirinya sendiri disebut gelung (loop). Jika terdapat lebih dari satu sisi yang menghubungkan dua titik pada suatu graf, maka sisi-sisi tersebut disebut sisi rangkap atau sisi ganda. Graf yang tidak mempunyai sisi rangkap dan tidak memiliki gelung disebut graf sederhana *[3]*.

### 2. Digraf

Graf berarah atau digraf $D$ adalah struktur yang terdiri dari himpunan tak kosong dan berhingga, yang unsurnya disebut titik, beserta himpunan pasangan berurutan dari dua titik berbeda yang disebut busur.

Contoh:



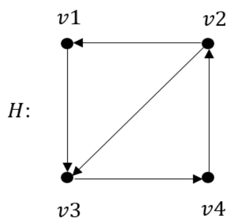

**Gambar 2.** Digraf $G$

Pada Gambar 2 titik $u$ terhubung langsung ke $v$. Di lain pihak, titik $v$ tidak terhubung langsung ke $u$ [2].

### 3. Derajat Titik pada Digraf

Misalkan $v$ adalah titik pada digraf $D$. Banyaknya titik yang terhubung langsung ke $v$ disebut derajat masuk (*indegree*) dari $v$, dan dinotasikan dengan $id(v)$. Banyaknya titik yang terhubung langsung dari $v$ disebut derajat keluar (*outdegree*) dari $v$, dan dinotasikan dengan $od(v)$ [2].

Contoh:

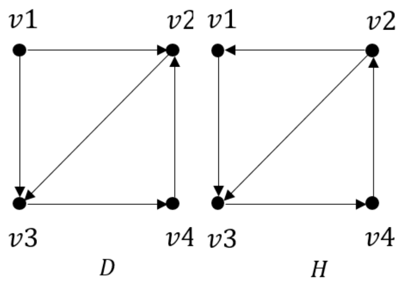



**Gambar 3.** Derajat Titik pada Graf $H$

### 4. Digraf Terhubung

Misalkan $D$ adalah suatu graf berarah. Graf berarah $D$ dikatakan terhubung kuat jika untuk setiap dua titik di $D$ ada lintasan berarah yang menghubungkan dua titik tersebut. Sedangkan graf $D$ dikatakan terhubung lemah jika $D$ tidak terhubung kuat tetapi graf tak berarah yang bersesuaian dengan graf $D$ terhubung [2].

Contoh:

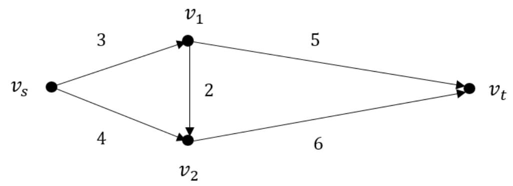



**Gambar 4.** Graf $D$ Terhubung Lemah; Graf $H$ Terhubung Kuat

### 5. Jaringan (*Network*)

Jaringan $N = (V(N), \Gamma(N))$ merupakan sebuah graf berarah sederhana terhubung lemah yang setiap unsurnya dikaitkan dengan bilangan real non negatif. Bilangan real non negatif yang dikaitkan pada busur $(V_i, V_j)$ di jaringan $N$ disebut kapasitas busur dan dilambangkan $c(V_i, V_j)$. Sebuah titik $s$ di jaringan $N$ disebut titik sumber jika $id(s) = 0$ dan sebuah titik $t$ di jaringan $N$ disebut titik tujuan jika $od(t) = 0$, sedangkan titik yang lain di jaringan $N$ disebut titik-titik antara [3].

Contoh:




**Gambar 5.** Jaringan $N$ dengan Titik Sumber $v_s$ dan Titik Tujuan $v_t$

### 6. Pemutus pada Jaringan

Misalkan $N$ sebuah jaringan dengan titik sumber $s$ dan titik tujuan $t$. Misalkan himpunan $X$ adalah himpunan bagian tak kosong dari $V(N)$ dan $X_1 = V(N) - X$. Jika $s \in X$, dan $t \in X_1$, maka himpunan busur $B(X, X_1)$ disebut sebuah pemutus$-(s,t)$ dari jaringan $N$, karena penghapusan semua busur $B(X, X_1)$ dari $N$, memutus semua lintasan berarah dari titik $s$ ke titik $t$ pada jaringan $N$ [3].

### 7. Konsep Flow pada Jaringan

Sebuah *flow* di jaringan $N$ dari titik sumber $s$ dan titik tujuan $t$ adalah sebuah fungsi yang memetakan setiap busur $(i,j)$ di $N$ dengan sebuah bilangan non negatif yang memenuhi syarat-syarat berikut:

1. $0 \leq f(i,j) \leq c(i,j)\ \forall (i,j) \in \Gamma(N)$ (disebut kapasita pembatas)
2. $\sum_{(i,j)\in O(s)} f(i,j) = \sum_{(i,j)\in I(t)} f(i,j)$ (disebut nilai *flow* $f$)
3. $\sum_{(i,j)\in O(s)} f(i,j) = \sum_{(i,j)\in I(t)} f(i,j)\ \forall x \in V(N) - \{s,t\}$ (disebut sebagai "konservasi *flow*") [3].

### 8. Flow Maksimum pada Jaringan

*Flow* $f$ bernilai $f_{s,t}$ dari titik $s$ ke titik tujuan $t$ pada jaringan $N$ dikatakan *flow* maksimum jika $f_{s,t} = \min\{c(X, X_1) | B(X, X_1)$ suatu pemutus$-(s,t)$ pada jaringan $N\}$ [3].

### 9. Algoritma Ford-Fulkerson

Secara sistematis, langkah-langkah pada Algoritma Ford-Fulkerson yaitu:

Input : Jaringan $N = (V, G)$ dengan titik sumber $s$ dan titik tujuan $t$.

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

Langkah 2 : Rutin-Pelabelan.

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

b. Pilih sebarang titik yang terlabel tetapi belum teramati, misalkan titik tersebut $v_x$. Untuk $\forall\ v_y \exists\ (y,x) \in \Gamma, v_y$ belum terlabel 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 $\forall\ v_y \exists\ (y,x) \in \Gamma, v_y$ belum terlabel 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. Diubah label $v_x$ dengan cara melingkari tanda + atau -. Sekarang titik $v_x$ terlabel dan teramati.

c. Ulangi langkah a. sampai:

1) Titik $v_t$ terlabel atau semua titik terlabel telah teramati tetapi titik $v_t$ tak terlabel.

2) Jika titik $v_t$ terlabel, lanjut ke langkah 3 atau jika semua titik yang terlabel telah teramati tetapi titik $v_t$ tak terlabel, maka iterasi dihentikan dan *flow* $f$ adalah *flow* maksimum dana jaringan $N$.

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

Langkah 4 : Tingkatkan nilai *flow* $f$ sebesar label $v_t$, berdasarkan lintasan peningkatan $P$ dengan menggunakan "Rutin Peningkatan". Rutin Peningkatan:

a. Misalkan $Z = t$ lanjutkan ke langkah b.

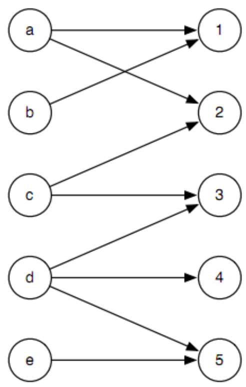
b. Jika label $v_z = (q, +, \varepsilon(t))$ tingkatkan nilai $f(q,z)$ dengan $\varepsilon(t) = i(P)$.

c. Jika $q = s$, hapus semua label. Pada tahap ini diperoleh *flow* $f$ baru dengan nilai $= i(P) +$ nilai *flow* $f$ lama. Ganti *flow* $f$ dengan *flow* $f$ yang baru, dan kembali ke langkah 1.

[3].

### 10. Jaringan Bipartisi

Sebuah jaringan $G = (V, E)$ dikatakan bipartisi jika himpunan titik $V$ dapat dipartisi menjadi dua sub-himpunan $V_1$ dan $V_2$ sehingga semua sisi dari $G$ mempunyai ujung di $V_1$ dan ujung lainnya di $V_2$, sebagaimana contoh berikut [4]:




**Gambar 6.** Jaringan Bipartisi

### PEMBAHASAN

Data simulasi yang disediakan merupakan data pendistribusian barang yang memiliki lima titik sumber dan lima titik tujuan dimana untuk mencari *flow* maksimum pada data tersebut, maka dengan memodifikasi Algoritma Ford-Fulkerson.

Bentuk modifikasi algoritma ini adalah:

Input : Jaringan $A = (V, G)$ dengan beberapa titik sumber $(s_i)$ dan beberapa titik tujuan $t(t_j)$.

Langkah 1 : Dibentuk sebuah jaringan baru dari $A$ yang dimisalkan $A^*$, dengan menambahkan satu titik sumber $s$ dan satu titik tujuan $t$ sedemikian hingga $s$ merupakan satu-satunya titik sumber di jaringan $A^*$ dan $t$ merupakan satu-satunya titik tujuan di $A^*$.

Langkah 2 : Dibuat kapasitas busur $(s, s_i)$ dan $(t_i, t)$ dengan nilai kapasitas tak hingga atau dilambangkan $\infty$, dan nilai *flow* dapat dimulai dari 0.

Langkah selanjutnya yaitu memaksimumkan *flow* $s - t$ dengan menggunakan algoritma *flow* maksimum, yaitu:

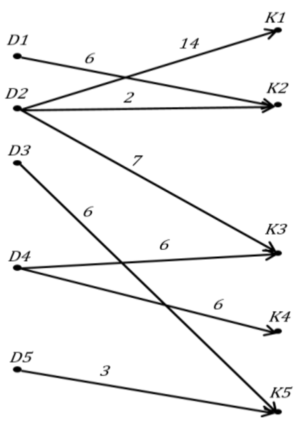
Langkah 3 : Lakukan rutin pelabelan.

Langkah 4 : Gunakan prosedur balik.

Langkah 5 : Lakukan rutin peningkatan.

Langkah 6 : Apabila titik-titik di $A^*$ yang terlabel telah teramati semua, dan titik $t$ tidak terlabel maka iterasi dihentikan.

Selanjutnya data dibentuk menjadi sebuah jaringan yang dimisalkan jaringan $M$, sebagaimana dapat dicontohkan pada gambar berikut:

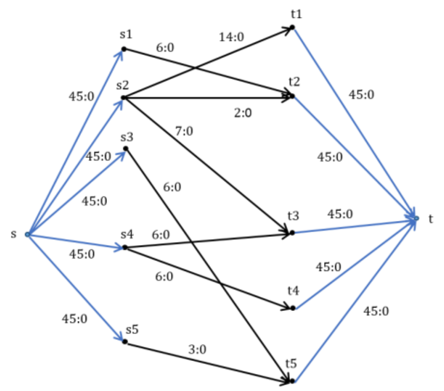



**Gambar 7.** Jaringan $M$

Jaringan $M$ pada Gambar 7 akan dicari *flow* maksimumnya menggunakan modifikasi Algoritma Ford-Fulkerson dengan langkah-langkah sebagai berikut:

Langkah 1 : Dibentuk sebuah jaringan baru $M^*$ dari $M$ dengan menambahkan satu titik sumber baru dan satu titik tujuan baru sedemikian hingga merupakan titik sumber utama dan titik tujuan utama di $M^*$ dimana titik sumber utama dan titik tujuan utama pada jaringan $M^*$ diberi warna biru.

Langkah 2 : Dibuat kapasitas busur $(s, s1), (s, s2), (s, s3), (s, s4)$ dan $(s, s5)$ serta $(t1, t), (t2, t), (t3, t), (t4, t)$ dan $(t5, t)$ yang juga diberi warna biru dengan nilai kapasitas utama yaitu stok persediaan barang dengan nilai 45 dan nilai *flow* awal 0.




**Gambar 8.** Jaringan $M^*$ dengan Nilai *Flow* Awal $f = 0$

Jaringan $M^*$ pada Gambar 8 mempunyai kapasitas di titik sumber utama dengan nilai 45, sedangkan total kapasitas di beberapa titik sumber dengan nilai 50. Sebagaimana prosedur Algoritma Ford-Fulkerson pada bab sebelumnya, agar nilai *flow* segera terpenuhi maka strategi *flow* maksimum yang dilakukan yaitu dipilih lintasan dari titik sumber utama ke titik yang mempunyai busur terhubungnya paling banyak serta titik yang mempunyai kapasitas terbesar, sehingga urutan iterasi untuk lintasan peningkatan yaitu di titik $s2, s4, s1, s3, s5$.

Langkah selanjutnya dalam pencarian *flow* maksimum menggunakan modifikasi Algoritma Ford-Fulkerson untuk iterasi pertama yaitu sebagai berikut:

1) Dari titik $s$, diberi label titik $s1, s2, s3, s4$ dan $s5$ masing-masing $s1 = (s, +, 45)$, $s2 = (s, +, 45)$, $s3 = (s, +, 45)$, $s4 = (s, +, 45)$ dan $s5 = (s, +, 45)$. Sekarang titik $s$ telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.

2) Dari titik $s2$, diberi label untuk titik $t1, t2, t3$ masing-masing $t1 = (s2, +, 14), t2 = (s2, +, 2), t3 = (s2, +, 7)$. Titik $s2$ telah terlabel dan teramati dengan label $s2 = (s, +, 45)$.

3) Dari titik $t1$, diberi label untuk titik $t$ dengan label $t = (t1, +, 45)$. Titik $t1$ telah terlabel dan teramati dengan label $t1 = (s2, +, 14)$.

Langkah 4 : Gunakan prosedur balik.

Titik $t$ dilabel dari titik $t1$, titik $t1$ dilabel dari $s2$ dan titik $s2$ dilabel dari titik $s$. Dengan demikian diperoleh lintasan peningkatan $P = (s, s2, t1, t)$ dan $i(P) = \min(45,14,45) = 14$.

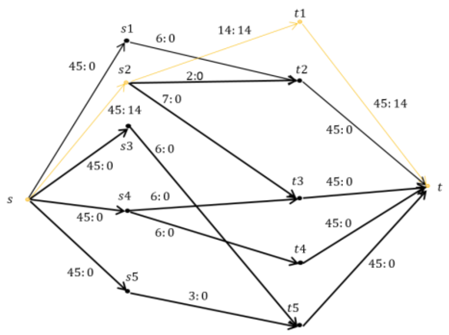
Langkah 5 : Rutin peningkatan.

Titik $t = (t1, +, 45)$ maka nilai *flow* pada busur $(t1, t)$ ditambah 14.

Titik $t1 = (s2, +, 14)$ maka nilai *flow* pada busur $(s2, t1)$ ditambah 14.

Titik $s2 = (s, +, 45)$ maka nilai *flow* pada busur $(s, s2)$ ditambah 14.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_1 = 14$. Lintasan peningkatan pada jaringan $M^*$ diberi warna orange dan ditunjukkan sebagaimana pada gambar berikut:




**Gambar 9.** Jaringan $M^*$ dengan Nilai *Flow* Baru $f_1 = 14$

Iterasi selanjutnya diulangi pencarian *flow* maksimum menggunakan Algoritma Ford-Fulkerson dari titik sumber ke titik tujuan di $M^*$ dengan kembali ke langkah 2 sampai iterasi ke 8 yaitu untuk lintasan peningkatan di titik $s5$ dan diperoleh nilai *flow* maksimum sebesar 45.

Berdasarkan jaringan yang telah terbentuk pada Gambar 7, jaringan $M$ merupakan bentuk jaringan bipartisi dimana himpunan titik-titik di jaringan $M$ dapat dipartisi menjadi dua sub-himpunan, sehingga jaringan $M$ akan dipartisi berdasarkan titik sumber dan dicari *flow* maksimumnya untuk menunjukkan bahwasanya nilai *flow* maksimumnya sama antara jaringan yang tidak dipartisi dengan jaringan yang dipartisi. Jaringan $M$ dipartisi menjadi lima karena memiliki lima titik sumber, karena mempunyai lima titik tujuan maka dengan memanfaatkan modifikasi Algoritma Ford-Fulkerson, akan ditambahkan satu titik tujuan baru dengan kapasitasnya adalah stok persediaan barang sebesar 45. Selanjutnya dilakukan iterasi dengan mencari lintasan peningkatan dari masing-masing titik sumber ke titik tujuan utama sampai tidak ditemukan lagi lintasan peningkatan, yaitu pada saat semua titik sumber telah diselidiki. Sehingga nilai *flow* maksimum pada jaringan diperoleh dari total *flow* maksimum untuk semua titik sumber di jaringan, dan didapatkan *flow* maksimum dengan iterasi sebanyak lima nilainya 45.

Pencarian *flow* maksimum pada jaringan $M$ juga dapat dilakukan dengan membuat program menggunakan Matlab R2010a dengan langkah iterasi sebagaimana prosedur yang dimiliki Algoritma Ford-Fulkerson. Tujuan membuat program ini yaitu agar pencarian *flow* maksimum dapat diketahui dengan lebih mudah, dimana hasilnya dengan iterasi sebanyak 8 nilai *flow* maksimum sebesar 45.

Berdasarkan hasil contoh pendistribusian barang dengan pencarian *flow* maksimum pada jaringan $M$ di atas didapatkan *flow* maksimum dengan nilai $f_8 = 45$, pada jaringan yang dipartisi didapatkan *flow* maksimum dengan nilai $f_5 = 45$ serta dengan program di Matlab R2010a dengan iterasi sebanyak 8 didapatkan *flow* maksimum sebesar 45.

## KESIMPULAN

Bentuk modifikasi Algoritma Ford-Fulkerson untuk menentukan *flow* maksimum pada jaringan yaitu:

a. Membuat jaringan baru dari jaringan distribusi barang dengan menambahkan satu titik sumber utama dan satu titik tujuan utama di jaringan baru tersebut.

b. Membuat kapasitas busur dari titik sumber utama ke beberapa titik sumber dan membuat kapasitas busur dari beberapa titik tujuan ke titik tujuan utama dengan kapasitas maksimum dan memberikan nilai *flow* awal pada semua busur di jaringan baru dengan nilai 0.

Setelah itu memaksimumkan *flow* dari titik sumber baru ke titik tujuan baru dengan:

c. Melakukan pelabelan titik.

d. Menggunakan prosedur balik.

e. Mencari lintasan peningkatan sampai titik-titik di jaringan yang terlabel telah teramati semua dan titik tujuan baru tidak terlabel maka iterasi dihentikan.

Dari bentuk modifikasi di atas, didapatkan *flow* maksimum pada jaringan tersebut dengan nilai $f_8 = 45$, pada jaringan yang dipartisi didapatkan *flow* maksimum dengan nilai $f_5 = 45$ serta dengan membuat program di Matlab R2010a didapatkan *flow* maksimum dengan iterasi sebanyak 8 nilainya 45.

**DAFTAR PUSTAKA**

[1] T. Farizal, "Pencarian Flow Maksimum dengan Algoritma Ford-Fulkerson," Universitas Negeri Semarang, Semarang, 2013.

[2] G. Charland and L. Lesniak, Graf and Digraf, London: Chapman & Hall/CRC, 1996.

[3] I. Budayasa, Teori Graph dan Aplikasinya, Surabaya: Unesa University Press, 2007.

[4] R. K. Ahuja, J. B. Orlin, C. Stein and R. E. Tarjan, "Improved Algorithms for Bipartite Network Flow," *SIAMJ COMPUT,* pp. 906-933, 1994.