Aplikasi Algoritma Network Flow untuk Manajemen Pendistribusian Minyak

Abdurrosyid Broto Handoyo/13510107[1]

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[1]13510107@std.stei.itb.ac.id

***Abstract*—Manajemen pendistribusin minyak merupakan masalah yang perlu diselesaikan dengan lebih optimal. Pendekatan melalui algoritma maxflow seperti metode Ford-Fulkerson dan algoritma Edmonds-Karp cukup reliable untuk mengoptimasi aliran minyak. Aliran minyak diabstraksi sebagai sebuah graf network flow. Kemudian setelah kalkulasi dengan algoritma Edmonds-Karp, kita langsung dapat mensetting flow minyak pada pipa-pipa tertentu agar flow menjadi maksimal.**

***Index Terms*—Edmonds-Karp, Ford-Fulkerson, Network Flow, Pendistribusian minyak**

I. PENDAHULUAN

Pada industri perminyakan, pendistribusian minyak merupakan salah satu hal yang sangat penting, karena hasil minyak yang dihasilkan mesin penambang, harus langsung dapat disimpan pada suatu tempat penyimpanan agar nantinya dapat diolah.

Pengaliran minyak dari tempat tambang menuju tempat penyimpanan biasanya dialirkan melalui sebuah rangkaian pipa. Antara tempat suplai minyak dengan tempat penyimpanan tersedapat beberapa terminal yang menghubungkan antar pipa. Suatu pipa juga memiliki kapasitas masih-masing dalam menghantarkan aliran air tiap detiknya. Perbedaan kapasitas ini terjadi akibat perbedaan jarak antar terminal, tekstur tanah, dan sebagainya.

Pipa yang mengalirkan minyak tersebut, memang kapasitasnya tidak dapat terlalu besar, karena makin besar, maka makin mudah untuk bocor dan sebagainya. Sehingga, permasalahannya menjadi bagaimana kita dapat mengoptimalkan kuantitas minyak yang dapat dialirkan bila kita menggunakan rangkaian pipa yang tersedia tersebut? Kemudian, bila kita akan menambah pipa, maka kita harus menambah pipa pada pasangan terminal manakah agar minyak yang dialirkan dapat lebih besar lagi? Atau pada pipa manakah yang akan kita tambah kapasitasnya agar, lebih besar aliran yang mengalir dari suatu network pipa yang sudah ada?

Makalah ini akan membahas dan mencoba menjawab mengenai hal-hal tersebut menggunakan pendekatan teori graf.

II. NETWORK FLOW

Untuk menyelesaikan permasalahan mengenai sistem pipa untuk mengalirkan minyak ini, kita dapat menggunakan pendekatan teori graf dalam Matematika Diskrit. Rangkaian pipa yang terhubung tersebut kita abstraksikan sebagai sebuah graf. Suatu terminal (termasuk tempat tambang minyak yang menjadi sumber minyak dan tempat tujuan aliran minyak tersebut atau tempat penyimpanan) merupakan simpulnya (vertex). Pipa yang menghubungkan antar terminal merupakan edge nya. Karena kapasitas setiap pipa yang menghubungkan terminal tersebut berbeda kapasitas dalam menghantarkan minyak, maka graf kita ini termasuk weighted graph.

Pada bahasan kali ini, kita akan mendefinisikan sebuah flow network. Sebuah flow network G = (V, E) adalah sebuah graf berarah dengan tiap edge (x,y) ϵ E memiliki kapasitas yang tidak negative c(x,y) ≥ 0. Jika edge (x,y) tidak berada pada himpunan E, maka diasumsikan c(x,y) = 0. Pada model kali ini terdapat 2 vertex khusus, yaitu sebuah source s dan sebuah sink t. Dalam bahasan kita kali ini, kita asumsikan juga untuk setiap rangkaian pipa, terdapat vertex v ϵ V, sehingga terdapat s → v → t.

Kita akan mendefinisikan sebuah flow dengan lebih formal. Sebuah flow pada G adalah fungsi f : V x V → R yang memenuhi 3 properti:

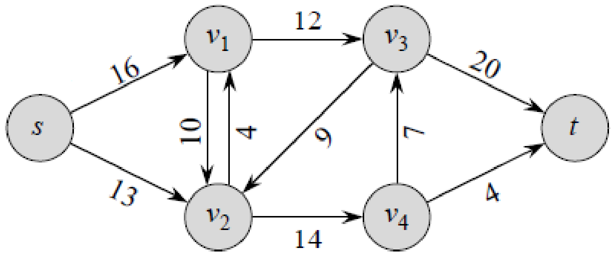
- Capacity Constraint
  Untuk semua (x,y) ϵ E, f(x,y) ≤ c(x,y). Dengan kata lain, flow yang mengalir pada sebuah edge tidak dapat melebihi kapasitasnya.
- Skew Symmetry
  Untuk semua (x,y) ϵ E, f(x,y) = - f(y,x).
- Flow Conservation
  Untuk semua x ϵ V, maka berlaku

$$\sum_{y \in V} f(x,y) = 0$$

Besarnya f(x,y) dapat positif, negative atau nol, dikaakan sebagai flow dari vertex x ke vertex y. Nilai dari sebuah flow f didefinisikan sebagai

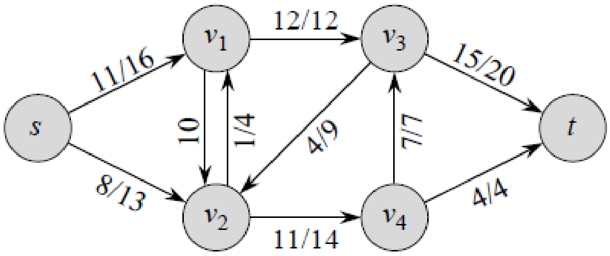
$$|f| = \sum_{y \in V} f(s,y)$$

Berikut ini (Gambar 2.1) adalah sebuah contoh graf flow network yang dapat memodelkan permasalahan pendistribusian minyak dari sebuah sumber s dan tujuannya t.



Gambar 2.1 Contoh graf flow network

Sebuah edge pada graf ini menunjukkan besarnya maksimal aliran minyak yang dapat dialirkan dari vertex x ke vertex y. Kita akan menentukan maksimal besarnya minyak yang dapat dialirkan dari s ke t.



Gambar 2.2 Flow maksimal dari graf pada gambar 2.1

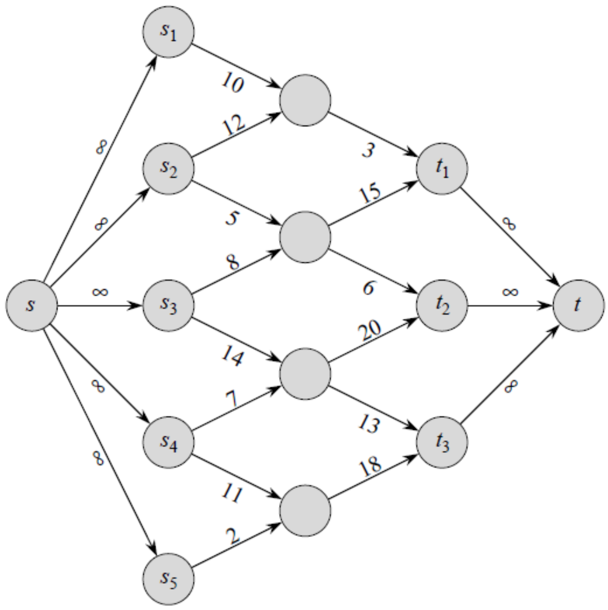
Pada gambar 2.2 dapat kita lihat flow f pada G adalah 19. Yaitu hanya 19 liter per satuan waktu tertentu saja yang dapat dialirkan dari sumber s ke tujuannya t.

Tiap edge pada graf (Gambar 2.2) dilabelkan dengan f(x,y)/c(x,y). Dari s ke v_1 kita hanya mentransfer 11 liter dari maksimal 16 liter minyak. Dari s ke v_2 kita hanya mentransfer 8 liter dari maksimal 13 liter minyak. Begitu pula seterusnya.

Pada model flow yang kita buat ini, dapat kita lihat bahwa pipa yang dapat mengalirkan minyak dari v_1 ke v_2 tidak dipakai untuk mengalirkan minyak sama sekali, karena kalau kita perhatikan lagi dengan seksama maka menggunakan pipa ini ataupun tidak, maka tetap saja hanya 19 liter saja, jumlah maksimal yang dapat dialirkan dari s ke t.

Namun pada implementasinya, aliran minyak tentu saja tidak hanya dari sebuah source dan tujuan dari aliran tersebut tidak hanya dari sebuah tempat penyimpanan. Maka dari itu, kita harus memodelkan graf flow network dengan sedikit berbeda.

Graf tersebut dapat berisi set m source$\{s_1, s_2, s_3, \dots, s_m\}$ dan n sink $\{t_1, t_2, t_3, \dots, t_n\}$. Kita dapat mereduksi permasalahan multiple source dan multiple sink(tujuan) menjadi sebuah graf flow network biasa dengan menambahkan 2 vertex. Kita tambahkan supersource ss dan sisi berarah (ss, s_i) dengan kapasitas c(ss, s_i) = ∞ untuk setiap i = 1, 2, 3, …, m. Kita juga menambahkan supersink std an sisi berarah (t_i, st) dengan kapasitas c(t_i, st) = ∞ untuk setiap i = 1, 2, 3, …, n.



Gambar 2.3 Graf network flow multiple source dan sink dengan sebuah supersource dan supersink

Dengan merepresentasikan graf multiple source-multiple sink menjadi single source-single sink, maka operasi untuk mencari maksimum flow akan menjadi lebih mudah.

III. Pendekatan Metode Ford-Fulkerson dan Algoritma Edmonds-Karp

Bagaimana kita mencari maksimum flow dari sebuah graf flow network? Pada bab ini kita akan membahas mengenai Metode Ford-Fulkerson dan Algoritma Edmons-Karp. Kita memakai istilah metode untuk Ford-Fulkerson dibandingkan algoritma karena Ford-Fulkerson dapat diimplementasikan menjadi beberapa cara dengan kompleksitas yang berbeda-beda.

Metode Ford-Fulkerson:

```
Inisialisasi semua flow f menjadi 0
While masih ada augmenting path p do
   buat augmenting flow f pada p
→ f
```

Beberapa definisi yang penting pada bahsan kali ini adalah Residual Network, Augmenting Path dan Cut of Flow Network.

Bila diberikan sebuah flow dan network flow, Residual Network terdiri dari sisi-sisi yang dapat menerima flow tambahan. Secara lebih formal, kita memiliki flow network G = (V,E) dengan sumber s dan tujuan t. Misalkan juga f adalah flow pada G, besarnya flow tambahan yang dapat ditambahkan dari x ke y tanpa melebihi kapasitas c(x,y) adalah Residual Capacity (x,y).

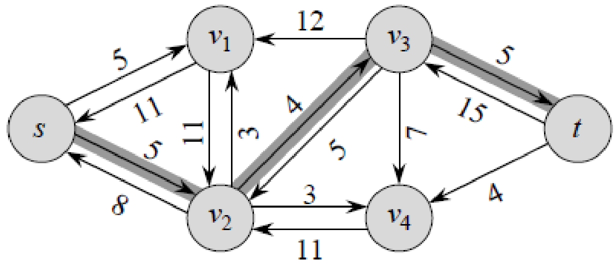
$$c_f(x,y) = c(x,y) - f(x,y)$$

Residual network dari G setelah dilalirkan flow f adalah $G_f = (V, E_f)$ dengan

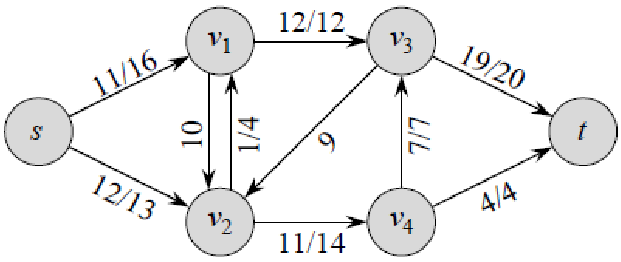
$$E_f = \{(x,y) \in V \, x \, V : \, c_f(x,y) > 0\}$$

Contoh, jika c(x,y) = 16 dan f(x,y) = 11, maka kita masih dapat menambahkan f(x,y) sebesar $c_f(x,y)$ = 5 sebelum melebihi kapasitasnya.

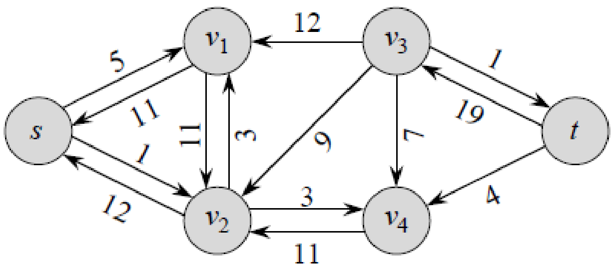
Misalkan pada gambar 2.2 adalah flow yang terjadi pada saat ini. Bila kita menambahkan sebuah flow lagi seperti pada gambar 3.1, maka kita mendapatkan flow network yang baru gambar 3.2. Residual network dari gambar 3.2, dapat kita lihat pada gambar 3.3.



Gambar 3.1



Gambar 3.2



Gambar 3.3

Augmenting path p adalah suatu simple path dari s ke t pada residual network G_f. Bagian edge graf yang diberikan bayangan pada gambar 3.1 adalah sebuah augmenting path. Pada path tersebut, kita hanya bisa mengalirkan maksimal 4 satuan saja agar tidak melanggar kapasitas dari masing-masing edge padaflow tersebut.

Metode ford Fulkerson akan terus mencari augmenting path hingga kita telah menemukan maksimum flow dari graf tersebut. Teorema max-flow min-cut yang pembuktiannya akan kami lampirkan, menyatakan flow telah maksimum jika residual networknya sudah tidak memiliki augmenting path lagi.

Untuk membantu pembuktian dari teorema tersebut, kita akan mendefinisikan beberapa istilah. Sebuah cut (S,T) dari flow network G = (V,E) adalah sebuah partisi V pada S dan T = V – S sedemikian sehingga $s \in S$ dan $t \in T$.

Lemma 3.1.

Misalkan f adalah flow pada flwo network G dengan sumber s dan tujuan t, dan misalkan (S,T) adalah cut dari G. Maka, net flow yang melewait (S,T) adalah $f(S,T) = |f|$

Proof:

Kita perhatikan bahwa f(S – s, V) = 0, karena flow tersebut tidak memiliki sumber s.

f(S,T) = f(S, V) – f(S, S)
= f(S, V)
= f(s, V) + f(S – s, V)
= f(s, V)
= $|f|$

Corollary 3.1

Nilai semua flow f pada flow network G dibatasi dari atas oleh kapasitas semua cut pada G.

Proof:

Misalkan (S,T) menjadi cut G dan f menjadi flow apapun, berdasarkan Lemma 3.1,

$|f|$ = f(S, T)
$= \sum_{x \in S} \sum_{y \in T} f(x,y)$
$\leq \sum_{x \in S} \sum_{y \in T} c(x,y)$
= c(S, T)

Teorema 3.1

Teorema maxflow min-cut, jika f adalah sebuah flow dari sebuah flow network G = (V,E) dengan sumber s dan tujuan t, maka beberapa kondisi berikut adalah equivalent:

1. f adalah maksimum flow pada G
2. Residual network G_f sudah tidak memiliki augmenting path lagi
3. $|f| = c(S,T)$ untuk beberapa cut (S, T)

Proof:

(1) → (2)

Kita akan membuktikan menggunakan kontradiksi. Asumsikan f adalah maksimum flow pada G tetapi masih ada G_f yang memiliki augmenting path p. Maka, jumlah flow yang terjadi f + f_p, dengan f_p adalah flow pada augmenting path pada p. Maka kontradiksi dengan asumsi kita tadi.

(2) → (3)

Misalkan G_f tidak memiliki augmenting path, dengan kata lain G_f tidak memiliki path dari s ke t. Definisikan

S = { $v \in V$: ada path dari s ke t pada G_f}

Dan juga definisikan T = V – S. Maka partisi (S, T) adalah sebuah cut. Dan kita dapatkan $s \in S$ dan $t \notin S$ karena tidak ada path dari s ke t. Untuk tiap pasang x dan y, $x \in S$ dan $y \in T$, berlaku f(x,y) = c(x,y), karena jika tidak demikian $(x,y) \in E_f$, yang akan menempatkan y pada S. Karena demikian, maka $|f| = f(S,T) = c(S,T)$ (dengan Lemma 3.1)

(3) → (1)

Berdasarkan corollary 3.1, $|f| \leq c(S,T)$ untuk semua cut (S,T). Kondisi $|f| = c(S,T)$ menunjukkan bahwa f adalah maksmum flow.

Karena telah terbuktinya Teorema maxflow min-cut (Teorema 3.1), maka kita telah dapat mengimplementasikan Metode Ford-Fulkerson.

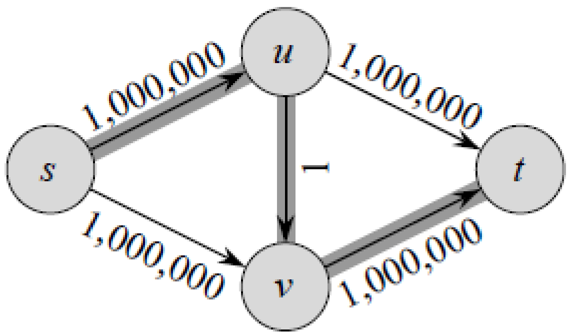
Ford Fulkerson(G,s,t)

1. **For** setiap edge $(x, y) \in E[G]$ **do**
2.   f[x,y] ← 0
3.   f[y,x] ← 0
4. **While** masih ada path p dari s ke t pada residual network G_f **do**
5. $c_f(p) \leftarrow \min\{c_f(x,y): (x,y)\ pada\ p\}$
6.   **For** setiap edge pada path p **do**
7. $f[x,y] \leftarrow f[x,y] + c_f(p)$
8. $f[y,x] \leftarrow -f[x,y]$

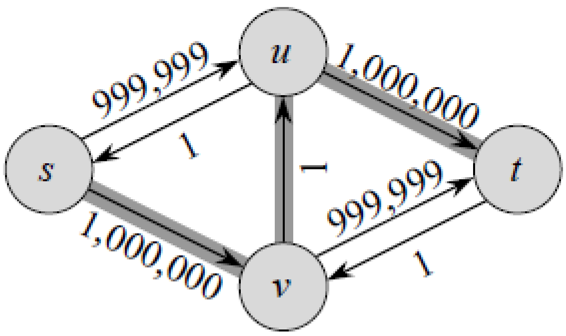
Baris 1 – 3 menginisialisasi semua flow. Sedangkan baris 4 – 8, terus mencari augmenting path p pada Gf dan augment flow f pada p dengan residual capacity $c_f(p)$.

Kita akan menganalisis running time dari algoritma ini, kompleksitas dari algoritma ini ditentukan berdasarkan bagaimana kita menentukan augmenting path p. Implementasi algoritma ini adalah $O(E|f^*|)$ dengan f^* adalah maksimum flow yang ditemukan aleh algoritma kita.

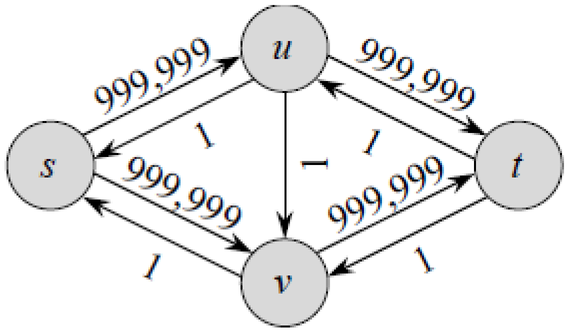
Setiap iterasi pada while loop memakan waktu O(E). Untuk mencari path dari s ke t juga memakan waktu O(E) juga jika kita menggunakan breatdh first search ataupun depth first search. Sedangkan besarnya f^* bergantung pada masing-masing graf yang berbeda. f^* ini bisa sangat besar pada graf tertentu, contohnya pada gambar 3.4. Pada gambar tersebut kita perhatikan bahwa bila kita memilih edge yang kapasitasnya 1 (gambar 3.5 dan 3.6) pada path yang kita pilih, maka kita akan mencari augmenting path sebesar 2000000 kali, padahal sebenarnya dapat dilakukan sebanyak 2 kali saja.



Gambar 3.4



Gambar 3.5



Gambar 3.6

Dengan menggunakan algoritma Edmond-Karp, kita dapat mencari maksimum flow dengan lebih cepat. Algoritma Edmond-Karp menggunakan metode Ford-Fulkerson, namun untuk mencari augmenting pathnya kita akan menggunakan algoritma shortest path dari s ke t pada residual network.

Teorema 3.2

Jika algoritma Edmond-Karp dijalankan pada sebuah flow network G = (V,E) dengan sumber s dan tujuan t, maka total flow augmentation yang akan dilakukan oleh algoritma adalah O(VE)

Proof:

Edge (x,y) pada sebuah residual network G_f disebut critical pada augmenting path p jika kapasitas residual p aalah kapasitas residual (x,y). Dengan kata lain $c_f(p) = c_f(x,y)$.

Setelah kita augment flow dari sebuah augmenting path, semua critical edge dari sebuah network akan hilang dari residual network tersebut. Ini disebabkan karena critical aliran yang dialirkan pada simulasi augmenting path p akan menghabiskan seluruh kapasitas (x,y). Setiap edge pun hanya bisa menjadi critical edge sebanyak |V|/2 – 1 kali saja.

Karena ada |E| buah edge yang dapat menjadi critical edge setiap membuat augmenting path, maka eksekusi flow augmentation dilakukan dengan kompleksitas O(VE). Maka teorema 3.2 terbukti.

Karena setiap iterasi pada metode Ford-Fulkerson dilakukan dengan kompleksitas O(E), maka Kompleksitas algoritma ini adalah $O(VE^2)$ dan lebih reliable dari pada breath first search sederhana.

Untuk algoritma shortest path dari s ke t, karena permasalahan ini termasuk single source shortest path, maka kita dapat menggunakan algoritma Dijkstra untuk implementasinya.

IV. MENGIMPLEMENTASIKAN MAXFLOW PADA PENGALIRAN MINYAK

Algoritma amxflow untk menghitung nilai maksimum flow yang mengalir pada sebuah graf dapat kita implementasikan pada aliran minyak. Kita akan menyetting pipa-pipa agar minyak mengalir melalui saluran yang kita buka. Kemudian kita harus bisa mengatur debit minyak yang mengalir pada sebuah pipa.

Contohnya pada gambar 3.2 yang menjadi maximum flow pada maxflow, kita harus mengaktifkan debit air yang melalui sebuah edge yang menghubungkan antara x dan y berdasarkan nilai f(x,) yang tertulis f(x,y)/c(x,y).

Jadi prosedur dalam implementasinya adalah

1. Kita ambil data dari sebuah tempat pertambangan dan seluruh pipa yang terkait dari sumbernya dan tempat penyimpanannya. Perhatikan bahwa sumber dan tujuan dari network tersebut biasanya tidak satu, sehingga kita harus memodelkan network tersebut seperti pada gambar 2.3.
2. Untuk sebuah waktu tertentu, kita harus mengkalkulasi debit minyak yang dihasilkan dari setiap source dalam waktu diskrit tertentu dan mengkalkulasikannya flow network yang terbaik untuk waktu diskrit tersebut.
3. Ubah settingan tiap pipa dan terminal agar aliran minyak menjadi seperti hasil maximum flow pada graf tersebut.
4. Waktu diskrit berikutnya kita akan menggunakan langkah yang 2 dan 3 terus selama mesin penambang aktif.

V. KESIMPULAN

Keuntungan dengan menggunakan algoritma ini dalam implementasinya adalah aliran minyak akan selalu menjadi maksimal sehingga produktivitas dari sebuah perusahaan minyak tersebut semakin besar. Kontrol otomatis pada tiap pipa tidak melebihi beban akan mengatasi masalah tekanan yang berlebihan pada pipa tertentu agar mengalirkan minyak sesuai dengan yang dibutuhkan saja. Setting otomatis akan membuat berapapapun hasil minyak yang diperoleh, kita akan selalu mengalirkannya dengan cepat.

Kelemahan jika kita menggunakan desain ini adalah, kita perlua mendesain system otomatisasi dari setiap terminal pada network minyak tersebut. Yang dalam hal ini dibutuhkan riset lebih lanjut dalam hal system control fisik dari setiap terminal tersebut.

REFERENCES

[1] Rosen, Kenneth H. “Discrete Mathematics and Its Application 6th edition”. 2007. McGraw-Hill

[2] Cormen, et.al. “Introduction to Algorithm”. 1994. The MIT Press

[3] Ford, L. R.; Fulkerson, D. R. "Maximal flow through a network". 1956. *Canadian Journal of Mathematics* **8**: 399–404.

[4] Jack Edmonds and Richard M. Karp (1972). "Theoretical improvements in algorithmic efficiency for network flow problems". *Journal of the ACM* 19 (2): 248–264.

[5] “Flow Network” URL : “http://en.wikipedia.org/wiki/Flow_network”

[6] “Ford-Fulkerson Algorithm” URL : “http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm”

[7] “Maxflow min-cut Theorem” URL : “http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem”

[8] “Edmonds-Karp Algorithm” URL : “http://en.wikipedia.org/wiki/Edmonds-Karp_algorithm”

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

ttd

Abdurrosyid Broto Handoyo/13510107