

Optimisasi Distribusi Gas Alam “City Gas” Dengan Algoritma Dinic

Ali Akbar - 13514080

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

aliakbr@students.itb.ac.id

Abstrak—Gas Alam adalah salah satu sumber energi terbesar di dunia. Saat ini gas alam dimanfaatkan untuk berbagai aspek. Hingga lahirlah ide untuk memanfaatkan gas alam untuk rumah tangga sebuah kota atau disebut dengan City Gas. Namun pendistribusian gas alam ini masih belum merata dan optimal di setiap daerahnya. Dengan algoritma *maximum flow* salah satunya Algoritma Dinic kita dapat mendistribusikan sebuah gas pada banyak pipa aliran gas namun meninggalkan hasil yang optimal pada akhir aliran gasnya.

Kata Kunci—algoritma Dinic, *blocking flow*, graf aliran, distribusi gas alam.

I. PENDAHULUAN

Gas alam adalah salah satu sumber daya alam terbesar yang ada di Indonesia. Sumber energi ini dapat menggantikan sumber daya alam lain yang cukup sulit untuk didapat seperti minyak bumi misalnya. Gas alam saat ini dimanfaatkan untuk bahan bakar, pembangkit listrik, dan dapat menjadi komoditas energy yang dapat diekspor. Karena kelimpahan gas alam ini muncul lah ide untuk mengganti bahan bakar untuk keperluan domestik (masyarakat) dengan sumber ini. Sehingga lahirlah ide untuk membuat “City Gas” yaitu pendistribusian gas alam ke rumah-rumah tangga untuk keperluan hidup rumah tangga tersebut. Dengan city gas ini maka sebuah kota dapat terhindar dari permasalahan “mafia gas” di beberapa daerah.

Saat ini di Indonesia beberapa kota telah memulai untuk menerapkan system City Gas ini, contohnya Surabaya dan Jambi. Namun masih banyak kekurangan dalam sistem pendistribusian gas alam ini sehingga gas belum terdistribusi secara merata ke tiap daerah. Contohnya seperti di Kota Jambi hanya daerah sekitar kelurahan Kenali saja yang merasakan gas alam untuk rumah tangga ini. Maka dari itu harus ada cara untuk membuat penyaluran gas merata dengan sumber tertentu sehingga gas alam dapat tersalurkan secara merata pada daerah kota. Bukan hanya itu, management dan dasar pendesainan jaringan gas juga harus diperhitungkan. Permasalahan lain adalah sumber gas untuk kota tersebut juga belum memadai sehingga tidak meratanya aliran gas

pada kota Untuk meningkatkan aliran dari sumber maka perlu adanya penggabungan banyak sumber gas di beberapa tempat sehingga aliran gas menjadi merata. Namun akibat permasalahan saluran pipa tadi maka penyaluran gas antar sumber pastinya akan memiliki jumlah aliran maksimum. Permasalahan ini dapat didekati dan diselesaikan dengan menggunakan teori jaringan aliran dengan memanfaatkan algoritma – algoritma untuk mendapatkan *maximum flow* .

Salah satu algoritma untuk menyelesaikan permasalahan *maximum flow* adalah Algoritma Dinic dimana algoritma tersebut menggunakan *blocking flow* sebagai cara untuk menemukan aliran maksimum dalam sebuah graf aliran. Algoritma dinic ini mirip dengan algoritma Edmonds – Karp yang bekerja dengan kompleksitas $O(VE^2)$. Sementara Algoritma Dinic bekerja dengan kompleksitas $O(V^2E)$. Namun dengan bantuan algoritma konstruksi *blocking flow* yang lebih mangkus dan handal maka kompleksitasnya dapat berkurang dan komputasi menjadi lebih cepat seperti ketika Algoritma Dinic ini digabung dengan Algoritma MPM (Malhotra, Pramod Kumar, Maheswari) dalam konstruksi *blocking flow*. Kompleksitas Algoritma Dinic akan menjadi $O(n^3)$. [4]

Dalam makalah ini penulis akan mencoba untuk memanfaatkan Algoritma Dinic dan untuk melakukan pendekatan dalam penyelesaian masalah ini. Sebenarnya masih ada algoritma lain untuk membantu menyelesaikan permasalahan ini. Namun dalam makalah ini penulis akan menjelaskan cara kerja Algoritma Dinic dan implementasinya pada optimisasi distribusi gas alam.

II. LANDASAN TEORI

A. Graf

Graf adalah representasi dengan bentuk simpul dan sisi. Sisi adalah penghubung antara tiap simpul sementara simpul adalah titik atau node dari graf. Graf secara matematis didefinisikan sebagai pasangan himpunan (V,E) dengan V adalah simpul dan E adalah sisi dari graf [6]. Graf memiliki banyak bentuk, bisa hanya berupa graf tak berarah, berarah, berbobot, tak berbobot, planar dan

tidak planar, dan sebagainya. Graf banyak digunakan dalam memodelkan sesuatu. Misalnya memodelkan hubungan antara daerah A dengan himpunan daerah S. Memodelkan langkah-langkah, dan sebagainya. Graf berarah dapat memodelkan jaringan gas pada City Gas. Gambar dibawah adalah contoh sebuah graf yang sangat sederhana.

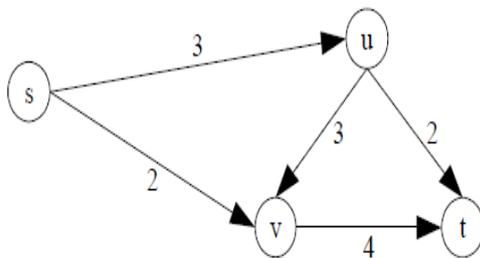


Gambar Graf

B. Network Flow

Network Flow atau aliran jaringan adalah jaringan yang merepresentasikan sebuah rangkaian aliran pada sebuah tempat. Jaringan tersebut dapat direpresentasikan dalam bentuk graf berarah yang memiliki bobot tertentu. Bobot dalam Graf Network Flow adalah kapasitas maksimum aliran dari sisi tersebut. Jika kita dekati, maka sisi-sisi pada graf jaringan aliran ini dapat memodelkan saluran pipa gas antara rumah satu dengan rumah lain (rumah dianggap sebagai simpul / *vertex*) ataupun stasiun gas satu dengan yang lain.

Pada bagian ini, kita definisikan graf berarah berbobot $G = (V, E)$ sebagai jaringan aliran. V adalah banyaknya simpul dan E adalah sisi dari graf. Bobot dari graf dinyatakan sebagai $c(x,y)$ yang berarti kapasitas (bobot) dari sisi (x,y) . Selain itu dalam model ini kita dapat merepresentasikan jaringan gas dalam bentuk aliran dari sebuah sumber s melalui rumah-rumah atau sumber-sumber (direpresentasikan sebagai simpul) dan sebuah tempat tujuan t . Sehingga dengan menggabungkan definisi diatas kita dapat mendefinisikan sebuah Graf Jaringan Aliran sebagai $G = ((V,E), c, s, t)$ dengan c adalah kapasitas, s adalah sumber, t adalah tujuan, dan (V,E) adalah konstruksi dari Graf yang memodelkannya (contoh graf aliran dapat dilihat pada gambar 1.1) . Definisi graf G akan kita pakai selama pembahasan ini.



Gambar 1.1[2]

Ada beberapa definisi yang harus kita ketahui sebelum membahas bagaimana cara kerja algoritma dinic

a. Arus dan Kapasitas

Arus kita definisikan sebagai f sehingga arus yang

mengalir dari sisi (x,y) disebut sebagai $f(x,y)$. Dalam Graf Aliran terdapat beberapa property seperti *capacity constraint* yang menunjukkan bahwa sebuah arus $f(x,y) \leq c(x,y)$ dengan $c(x,y) \geq 0$. Apabila $c(x,y) = 0$ maka sisi (x,y) bukanlah bagian E pada G . Pada aliran $f(x,y)$ juga berlaku hukum $f(x,y) = -f(y,x)$ yang disebut juga sebagai *Skew Simetry*. Maksud dari *skew simetry* adalah aliran dari x ke y merupakan lawan dari aliran dari y ke x . Selain itu ada properti Konservasi Aliran yaitu untuk semua x pada V di G berlaku $\sum f(x,y) = 0$ kecuali untuk sumber s dan t (karena merupakan awal dan akhir) dan $\sum f(x,y) = \sum f(y,z)$ untuk x, y, z bagian dari V . Maksud dari Konservasi aliran adalah total aliran pada sebuah simpul adalah 0 dan total aliran tiap simpul itu sama. Untuk lebih jelasnya dapat dilihat pada graf G pada gambar 1.2

b. Augmenting path

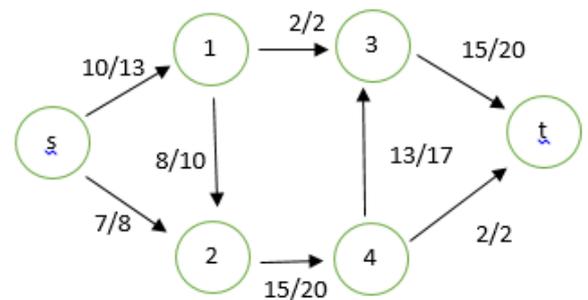
Augmenting path adalah jalur yang dapat dibentuk dari sebuah graf aliran G dari sumber s ke t dimana pada sisi sisi yang dilewati oleh aliran tersebut arusnya masih dapat ditingkatkan lagi (masih belum aliran maksimum). Atau definisi yang lebih singkatnya lagi adalah aliran yang dapat dibuat dari s ke t dengan graf residu G' dimana G' adalah graf residu dari G . Contoh augmented path dari graf residu G' dapat dilihat pada gambar 1.4 (graf residu G' pada gambar 1.3) dimana augmented path adalah garis yang berwarna biru.

c. Graf Residu

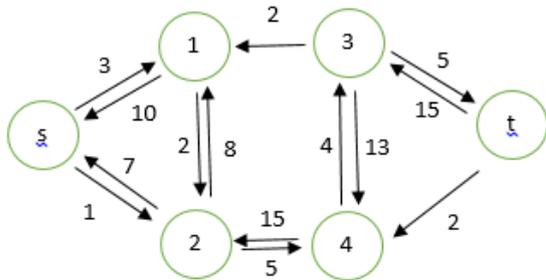
Graf Residu adalah graf aliran yang menunjukkan kapasitas sisa setelah adanya aliran dari s ke t . Misal kita definisikan sebuah graf residu G' dari graf aliran G . Jika graf aliran g memiliki aliran yaitu $f(x,y)$ dan kapasitas $c(x,y)$ maka kapasitas residu (disini didefinisikan d) pada G' adalah

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

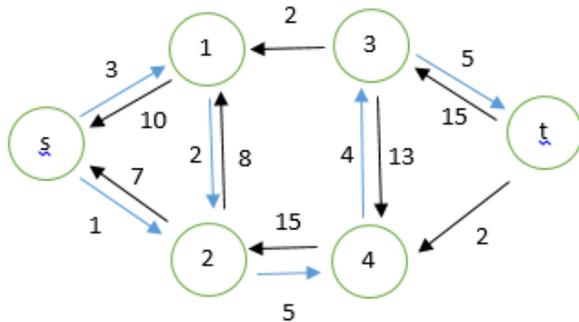
agar dapat memahami apa itu graf residu dan augmenting path dapat dilihat gambar dibawah ini :



Gambar 1.2



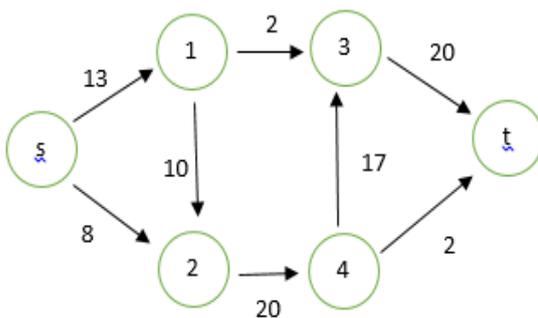
Gambar 1.3



Gambar 1.4

d. Blocking Flow

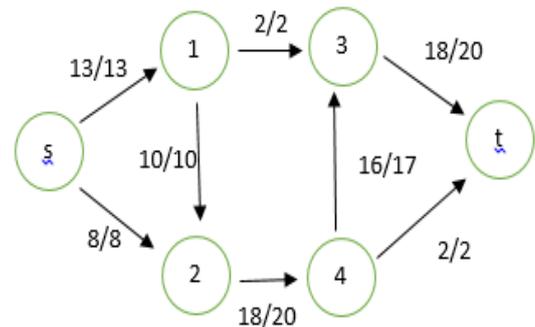
Blocking flow adalah aliran maksimum yang dapat mengalir pada sebuah graf residu G' sehingga terbentuk augmenting path dengan aliran maksimum. Untuk membentuk blocking flow terdapat begitu banyak algoritma. Seperti pendekatan dengan layered network sehingga dapat terbentuk blocking flow. Atau algoritma yang lebih cepat seperti algoritma MPM (tidak dibahas pada makalah ini). Dalam algoritma menggunakan layered network Graf aliran G dibagi menjadi layer-layer yang berisi himpunan simpul-simpul berdasarkan kejaukannya dari s . Untuk menemukan blocking flow kita menggunakan graf residu sebagai tinjauan penemuan aliran maksimumnya. Misal graf G pada gambar 1.2 dibuat dengan aliran 0 sehingga terbentuk graf residu G' (gambar 1.5)



Gambar 1.5

Pada graf residu G' tersebut dapat kita buat 4 buah layer dari simpul-simpulnya yaitu $V_0 = s$, $V_1 = 1$,

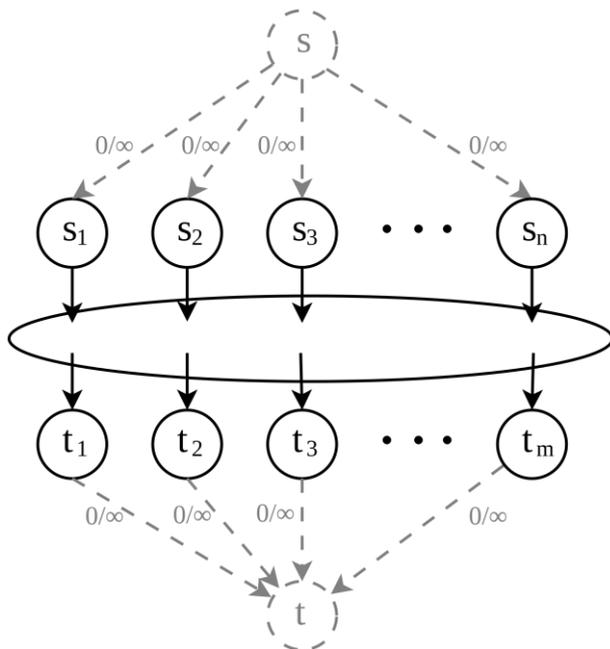
$V_2 = 3, 4$, $V_3 = t$. Kita akan mencari aliran maksimum dengan mendefinisikan bahwa ada sebuah kapasitas residu $g(e)$ dengan e adalah sisi dari graf G' yang sama nilainya dengan $c(e)$ pada graf G' . Lalu kita mencari *path* yang mungkin (augmenting path) dari s ke t pada graf residu G' . Misal kita definisikan $P = \{P_1, P_2, P_3, \dots\}$ adalah himpunan *path* dari s ke t (atau dari layer V_0 ke V_3). Untuk saat graf 1.5 kita dapat buat $P_1 = (s, 1, 3, t)$ lalu $P_2 = (s, 1, 2, 4, t)$ lalu $P_3 = (s, 2, 4, t)$ dan $P_4 = (s, 2, 4, 3, t)$. Lalu isi $g(e)$ pertama dengan kapasitas maksimum dari $g(e)$ dan jalan dari layer V_0 ke V_1 . Karena berdasarkan himpunan *path* P pada G' sekurang-kurangnya ada satu sisi $e_1 = (V_{j-1}, V_j)$ maka dapat dibentuk sebuah blocking flow seperti dibawah (diasumsikan sumber memiliki source yang cukup).



Gambar 1.6

Dalam penentuan jalur aliran. Kita menggunakan *path* dengan jalur terpendek dari s ke t . Maka dari itu sebelum membuat *blocking flow* ada beberapa metode seperti membuat Graf Level dimana tiap simpul diberi label $dist(v)$ yaitu jalur terpendek untuk mencapai v dari s . Untuk menemukan jalur terpendek kita dapat gunakan algoritma Dijkstra dan algoritma lainnya.

Dalam implementasi ke sebuah rancangan besar. Rancangan graf aliran nantinya dapat membentuk sebuah jaringan multi sumber yaitu dari sebuah sumber ke tujuan t maka t juga dapat menjadi sumber untuk yang lain dengan jumlah tujuan s lebih dari satu. Sistem seperti disebut dengan *multi-source graph*. Biasanya graf multi sumber digunakan untuk sebuah graf multi tujuan atau *multi sink*. Untuk menyelesaikan permasalahan optimasi pada *multi-source* maka graf harus didekati menjadi bentuk graf aliran satu sumber karena lebih mudah diselesaikan. Gambar dibawah adalah gambar graf *multisource*.



Gambar Graf Multi-source (diambil dari https://en.wikipedia.org/wiki/Maximum_flow_problem)

B. Algoritma Dinic

Algoritma Dinic memanfaatkan konstruksi *blocking flow* sehingga dapat dicari aliran maksimum pada sebuah *network flow*. Algoritmanya secara umum adalah sebagai berikut :

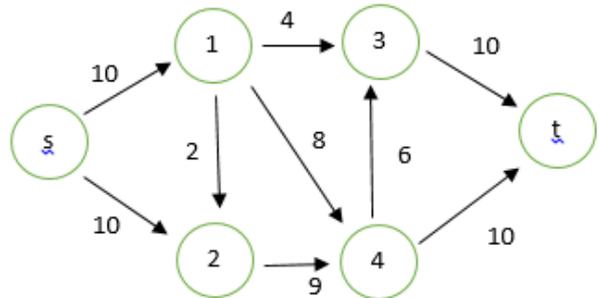
- Diketahui Graf Aliran $G = ((V,E), c, s, t)$
- Untuk e bagian dari E dan untuk seluruh anggota E inialisasi $f(e) = 0$.
- Bentuk graf residu dari flow yang ada (misal setelah inialisasi, definisikan sebagai G_f sebagai residu dari $f = 0$, atau jika telah melewati langkah d maka G_f adalah graf residu setelah dibentuk *blocking flow*).
- Bentuklah augmenting path $s - t$ pada G_f . Jika tidak bisa maka keluarkan nilai f . Namun jika bisa maka bentuk *blocking flow* dan hitung $|f|$ dari *blocking flow*. Lalu kembali ke langkah c.

Jika dibuat dalam bentuk proses komputasi maka dalam bentuk notasi algoritmik Algoritma Dinic berbentuk :

```

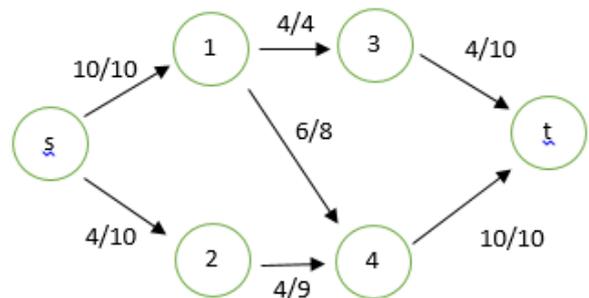
procedure_Dinic(input G : Graf Aliran ((V,E), c, s, t),
output f : integer)
ALGORITMA
  Inialisasi flow f = 0
  Bentuk graf residu Gf dari G dengan flow f
  If ( ada augmenting path s - t pada Gf ) then
    Cari blocking flow pada Gf
    Hitung nilai f
    Dinic(Gf, f1)
    → f + f1
  else
    → f
  
```

Berikut adalah simulasi algoritma Dinic pada Graf G pada gambar 1.7



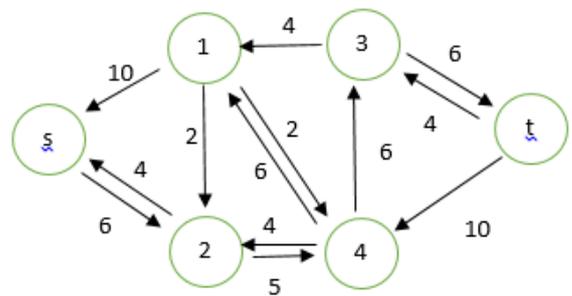
Gambar 1.7

- Langkah 1 : Inialisasi f dengan 0
- Langkah 2 : Kita dapatkan sebuah graf residu G_f . Untuk G dan f saat ini G_f akan menghasilkan graf yang sama dengan G .
- Langkah 3 : Augmented path dapat ditemukan pada G_f saat ini. Blocking flow yang terbentuk dari G_f saat ini adalah sebagai berikut



Gambar 1.8

- Untuk saat ini kita memiliki $f = 14$ (4 dari jalur $s,1,3,t$ lalu 6 dari jalur $s,1,4,t$ dan 4 dari jalur $s,2,4,t$).
- Kita kembali ke langkah 2 dimana kita sekarang memiliki graf residu yaitu



Gambar 1.9

- Lanjutkan ke langkah 3 hingga G_f tidak memiliki augmented path dari s ke t dan nilai f akhir adalah maximum flow dari graf aliran G .

Dari graf G nantinya kita akan dapat $|f| = 15$ sehingga t maksimum mendapatkan 15 unit dari sumber s .

III. IMPLEMENTASI DALAM JARINGAN GAS KOTA

Dalam implementasi jaringan gas kota jaringan gas dapat dipandang sebagai sebuah poligraf berarah *multi-source graph* dengan *multi sink*. Algoritma Dinic dapat membantu mengoptimalkan jaringan gas kota yang telah didesain oleh para teknisi dengan beberapa cara, yaitu :

- a. Memaksimalkan sumber gas dengan cara menghubungkannya dengan sumber yang lain. Dengan membuat sumber tujuan sebagai sink maka kita dapat membuat graf *single-source* sebanyak sumber yang memasok sink. Dengan Algoritma Dinic kita dapat mengoptimalkan gas yang ada pada sumber tujuan.
- b. Mengoptimalkan aliran dengan mengatur aliran pada rute pipa gas yang telah didesain oleh *engineer* dengan berbagai pertimbangan sehingga memberikan nilai yang optimal
- c. Seperti yang disinggung sebelumnya jaringan gas kota dapat dipandang sebagai jaringan graf *multi-source* dan *multisink*. Dengan pendekatan graf *single-source* pada subgraf tersebut maka kita dapat menerapkan algoritma Dinic dan mengoptimalkan aliran pada tiap sink. Sebenarnya sink itu sendiri juga dapat menjadi sumber apabila masih terhubung dengan graf lain. Sehingga nantinya antar daerah dapat dioptimalkan aliran gasnya dengan system seperti itu.
- d. Apabila jaringan gas City Gas rupanya hanya mampu memakai sebagian dari sumber P akibat keterbatasan dana pengadaan pipa. Maka gas sisa dapat dikemas dan didistribusikan lagi melalui jalur transportasi biasa ke daerah-daerah. Untuk mendistribusikan gas ini kita kembali gunakan Algoritma Dinic.

IV. KESIMPULAN

Algoritma Dinic dapat membantu pendistribusian gas alam pada sebuah kota. Kelebihan pada pemanfaatan Algoritma Dinic pada Optimisasi Jaringan Gas “City Gas” pada sebuah kota adalah minim kerugian akibat kesalahan dalam pengaliran gas ke suatu daerah. Algoritma Dinic juga lebih fleksibel karena kebagusan algoritma bertambah dengan penggabungan dengan algoritma lain serta lebih mangkus dibandingkan dengan Algoritma Edmonds-Karp atau Metode Ford-Fulkerson.

Sementara kelemahan dari penggunaan Algoritma Dinic pada optimisasi jaringan gas “City Gas” adalah Algoritma Dinic hanya memodelkan optimisasi pada sebuah aliran yang tidak memiliki galat. Sementara pada pengaliran gas terdapat galat-galat akibat system control dan bahan pipa sehingga aliran menjadi tak akurat. Algoritma ini hanya mampu melakukan pendekatan pada suatu debit aliran yang diskrit. Selain itu algoritma ini tergolong algoritma lama dan ada algoritma baru seperti Algoritma Push Relable Algorithm yang lebih mangkus dari Algoritma Dinic[7]. Untuk paper selanjutnya diharapkan Algoritma-

algoritma yang digunakan untuk menyelesaikan sebuah permasalahan menggunakan algoritma yang paling mangkus yang ada untuk masa kini.

V. UCAPAN TERIMAKASIH

Penulis mengucapkan puji dan syukur kepada Allah SWT karena berkat rahmatnya penulis dapat menyelesaikan tulisan ini. Penulis juga berterimakasih kepada dosen pengajar Matematika Diskrit Bapak Ir.Rinaldi Munir dan Ibu Harlili, M.Sc karena berkat pengajarannya penulis memiliki pengetahuan tentang graf yang menjadi pokok bahasan disini.

Penulis juga berterima kasih atas dukungan teman-teman dalam penyelesaian tulisan ini.

REFERENCES

- [1] http://www.cs.bgu.ac.il/~dinitz/Papers/Dinitz_alg.pdf (diakses pada 10 Desember 2015)
- [2] http://www.cs.cornell.edu/~wdtseng/icpc/notes/graph_part5.pdf diakses pada 10 Desember 2015
- [3] <http://www.cse.unt.edu/~tarau/teaching/AnAlgo/Dinic's%20algorit hm.pdf> diakses pada 10 Desember 2015
- [4] <http://iitg.ac.in/rinkulu/> “MPM Algorithm” oleh R.Inkulu (diakses di pada 10 Desember 2015)
- [5] Handoyo, Abdurrosyid Broto. 2011. *Aplikasi Algoritma Network Flow Untuk Manajemen Pendistribusian Minyak* . Makalah IF2120 Matematika Diskrit – Sem. I Tahun 2011/2012. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.
- [6] Munir, Rinaldi, *Diktat Kuliah IF 2120, Matematika Diskrit, Edisi Keempat, Program Studi Teknik Informatika, STEI ITB*, 2006, pp. VIII-2 – VIII-18, IX-1 – IX-9.
- [7] Yudhianto, Agung. *Algoritma Dinic Untuk Masalah Arus Maksimum*. 2003. (diakses pada 10 Desember 2015 di <http://repository.ipb.ac.id/handle/123456789/33525>)

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, 8 Desember 2015



Ali Akbar, 13514080