

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

SKRIPSI

**OLEH
FAHRUN NISA'
NIM. 11610056**



**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

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

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Sains (S.Si)**

**Oleh
Fahrin Nisa'
NIM. 11610056**

**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

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

SKRIPSI

Oleh
Fahrur Nisa'
NIM. 11610056

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 16 Juni 2015

Pembimbing I,

Pembimbing II,

H. Wahyu H. Irawan, M.Pd
NIP. 19710420 200003 1 003

Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

Mengetahui,
Ketua Jurusan Matematika

Dr. Abdussakir, M.Pd
NIP. 19751006 200312 1 001

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

SKRIPSI

**Oleh
Fahrur Nisa'
NIM. 11610056**

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Sains (S.Si)

Tanggal 26 Juni 2015

Penguji Utama : Evawati Alisah, M.Pd

Ketua Penguji : Drs. H. Turmudi, M.Si

Sekretaris Penguji : H. Wahyu H. Irawan, M.Pd

Anggota Penguji : Fachrur Rozi, M.Si

Mengetahui,
Ketua Jurusan Matematika

Dr. Abdussakir, M.Pd
NIP. 19751006 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Fahrún Nisa'
NIM : 11610056
Jurusan : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Algoritma Ford-Fulkerson untuk Memaksimumkan *Flow* dalam Pendistribusian Barang.

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 16 Juni 2015
Yang membuat pernyataan,

Fahrún Nisa'
NIM. 11610056

MOTO

“Adalah kehancuran bagi orang-orang yang menyerah” (Stalin)

يَبْنِي أَذْهَبُوا فَتَحَسَّسُوا مِنْ يُوسُفَ وَأَخِيهِ وَلَا تَأْيَسُوا مِنْ رَوْحِ اللَّهِ إِنَّهُ لَا يَأْسُ مِنْ رَوْحِ اللَّهِ إِلَّا الْقَوْمُ

الْكَافِرُونَ

“Hai anak-anakku, pergilah kamu, maka carilah berita tentang Yusuf dan saudaranya dan jangan kamu berputus asa dari rahmat Allah Swt.. Sesungguhnya tiada berputus asa dari rahmat Allah Swt., melainkan kaum yang kafir” (QS. Yusuf/12:87).



PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah, skripsi ini penulis persembahkan untuk:

Kedua orang tua tercinta

Bapak Supiaji dan Ibu Umatul Hasanah

Kedua adik tersayang

Dawam Ilhami dan Syarif Jihan Aula Firmansyah

Seluruh keluarga besar penulis



KATA PENGANTAR

Bismillahirrahmanirrahim

Assalamu'alaikum Wr. Wb.

Syukur Alhamdulillah penulis panjatkan ke hadirat Allah Swt. atas rahmat, taufiq, hidayah, serta inayah-Nya, sehingga penulis mampu menyelesaikan penyusunan skripsi yang berjudul “Algoritma Ford-Fulkerson untuk Memaksimumkan *Flow* dalam Pendistribusian Barang” sebagai salah satu syarat untuk memperoleh gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabiyullah Muhammad Saw., yang telah membimbing umatnya dari belenggu kejahiliyahan menuju kemenangan Islam yang terang benderang.

Dalam proses penyusunan skripsi ini, penulis menyadari banyak mendapatkan bimbingan, arahan, saran, doa, serta bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis sampaikan ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya kepada:

1. Prof. Dr. H. Mudjia Rahardjo, M.Si, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. drh. Bayyinatul Muchtaromah, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Abdussakir, M.Pd, selaku ketua Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. H. Wahyu H. Irawan, M.Pd, selaku dosen pembimbing I yang telah banyak memberikan bimbingan, arahan, saran, nasihat, motivasi, dan berbagai pengalaman yang berharga kepada penulis.
5. Fachrur Rozi, M.Si, selaku dosen pembimbing II yang telah banyak memberikan bimbingan dan berbagi ilmunya kepada penulis.
6. Seluruh dosen pengajar di Jurusan Matematika, terima kasih atas segala ilmu dan bimbingannya, serta kepada seluruh staf administrasi.
7. Bapak dan Ibu tercinta yang tidak pernah lelah menyalurkan doa, semangat, serta motivasi kepada penulis sampai saat ini.
8. Seluruh teman-teman di Jurusan Matematika angkatan 2011, terutama Nia Cahyani yang telah berjuang bersama-sama, terima kasih atas segala bantuannya, serta kepada Unit Kegiatan Mahasiswa Lembaga Kajian, Penelitian, dan Pengembangan Mahasiswa.
9. Semua pihak yang turut serta membantu dalam menyelesaikan skripsi.

Akhirnya penulis berharap semoga skripsi ini bermanfaat khususnya bagi penulis dan umumnya bagi pembaca.

Wassalamu'alaikum Wr. Wb.

Malang, Juni 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	
HALAMAN PENGAJUAN	
HALAMAN PERSETUJUAN	
HALAMAN PENGESAHAN	
HALAMAN PERNYATAAN KEASLIAN TULISAN	
HALAMAN MOTO	
HALAMAN PERSEMBAHAN	
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
ABSTRAK	xiii
ABSTRACT	xiv
ملخص	xv
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	5
1.6 Metode Penelitian.....	6
1.6.1 Jenis Penelitian.....	6
1.6.2 Data dan Sumber Penelitian	6
1.6.3 Tahap-tahap Penelitian.....	6
1.7 Sistematika Penelitian	7
BAB II KAJIAN PUSTAKA	
2.1 Graf.....	9
2.1.1 Lintasan pada Graf	11
2.1.2 Graf Terhubung	12

2.2 Graf Berarah atau Digraf.....	13
2.2.1 Derajat Titik pada Graf Berarah.....	14
2.2.2 Digraf Terhubung.....	16
2.3 Pengertian Jaringan	17
2.3.1 Pemutus pada Jaringan.....	18
2.3.2 Konsep <i>Flow</i> pada Jaringan	19
2.3.3 <i>Flow</i> Maksimum pada Jaringan	23
2.3.4 Algoritma <i>Flow</i> Maksimum pada Jaringan.....	27
2.4 Graf Bipartisi.....	30
2.5 Jaringan Bipartisi.....	31
2.5.1 Konsep <i>Flow</i> pada Jaringan Bipartisi.....	31
2.6 Kajian Agama Tentang Masalah dalam Kehidupan serta Kaidah Penyelesaiannya.....	32
BAB III PEMBAHASAN	
3.1 Paparan Data dan Modifikasi Algoritma Ford-Fulkerson.....	37
3.2 <i>Flow</i> Maksimum pada Data Distribusi Barang	39
3.3 <i>Flow</i> Maksimum pada Jaringan dengan Matlab	55
3.4 Kajian Keagamaan tentang Penyelesaian Masalah	56
BAB IV PENUTUP	
4.1 Kesimpulan.....	59
4.2 Saran.....	60
DAFTAR PUSTAKA	61
LAMPIRAN	
RIWAYAT HIDUP	

DAFTAR GAMBAR

Gambar 2.1 Contoh Graf G	9
Gambar 2.2 Graf G	10
Gambar 2.3 Graf G Merupakan Graf Sederhana; Graf C Bukan Graf Sederhana	11
Gambar 2.4 Jalan dan Lintasan pada Graf G	12
Gambar 2.5 Graf G Terhubung; Graf F Tidak Terhubung	12
Gambar 2.6 Contoh Digraf D	14
Gambar 2.7 Derajat Titik pada Graf H	15
Gambar 2.8 Graf D Terhubung Lemah; Graf H Terhubung Kuat	17
Gambar 2.9 Jaringan N dengan Titik Sumber v_s dan Titik Tujuan v_t	17
Gambar 2.10 Jaringan N dengan Titik Sumber v_s dan Titik Tujuan v_t	20
Gambar 2.11 $Flow$ f pada N dengan Nilai 5	21
Gambar 2.12 $Flow$ f_1 pada N dengan Nilai 6	22
Gambar 2.13 $Flow$ Maksimum dengan Nilai $Flow$ $f_2 = 9$	24
Gambar 2.14 Lintasan Peningkatan Terhadap $Flow$ f	27
Gambar 2.15 $Flow$ f' Diperoleh dengan Merevisi $Flow$ f'	27
Gambar 2.16 Contoh Graf Bipartisi	30
Gambar 2.17 Contoh Jaringan Bipartisi	31
Gambar 3.1 Jaringan M	39
Gambar 3.2 Jaringan M^* dengan Nilai $Flow$ Awal $f = 0$	40
Gambar 3.3 Jaringan M^* dengan Nilai $Flow$ Awal $f = 14$	42
Gambar 3.4 Jaringan M^* dengan Nilai $Flow$ Awal $f = 16$	43
Gambar 3.5 Jaringan M^* dengan Nilai $Flow$ Awal $f = 23$	45
Gambar 3.6 Jaringan M^* dengan Nilai $Flow$ Awal $f = 29$	46
Gambar 3.7 Jaringan M^* dengan Nilai $Flow$ Awal $f = 35$	48
Gambar 3.8 Jaringan M^* dengan Nilai $Flow$ Awal $f = 41$	49
Gambar 3.9 Jaringan M^* dengan Nilai $Flow$ Awal $f = 45$	51
Gambar 3.10 Jaringan M^* dengan Nilai $Flow$ Awal $f = 45$	52

ABSTRAK

Nisa', Fahrur. 2015. **Algoritma Ford-Fulkerson untuk Memaksimumkan *Flow* dalam Pendistribusian Barang**. Skripsi. Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) H. Wahyu H. Irawan, M.Pd. (II) Fachrur Rozi, M.Si.

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

Algoritma Ford-Fulkerson digunakan untuk mencari *flow* maksimum pada jaringan yang mempunyai satu titik sumber dan satu titik tujuan. Dengan mendefinisikan 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. Tujuan dari penelitian ini yaitu 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. Bagi penelitian selanjutnya diharapkan mampu mengkaji lebih mendalam terkait penelitian ini serta dapat dikembangkan pada jaringan penerbangan.

ABSTRACT

Nisa', Fahrur. 2015. **Ford-Fulkerson Algorithm to Maximise Flow in Distribution of Commodity**. Thesis. Department of Mathematics, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisors: (I) H. Wahyu. H. Irawan, M.Pd. (II) Fachrur Rozi, M.Si.

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

Ford-Fulkerson Algorithms is 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 points and several sink points. To solve the problems of the maximum flow in the network is then used a modified Ford-Fulkerson algorithm. The aim of this study is to show the steps of determining the maximum flow and obtained results using a modified Ford-Fulkerson algorithm on simulated data on the distribution of goods with a five source points and five sink points.

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 arc of some sink points 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 we used the Ford-Fulkerson Algorithm, namely by labeling a point, using the reverse procedure and to seek increase trajectory until all the labeled points have been observed and a major sink point has not been 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 routine in Matlab R2010a with a value of 45. For further research it is expected to examine more deeply and could be developed in the aviation network.

ملخص

نساء، فخر. ٢٠١٥. خوارزمية Ford-Fulkerson لتعظيم تدفق في توزيع السلع. البحث الجامعي. شعبة الرياضية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف: (١) الحاج وحي هاغكي إيراوان الماجستير. (٢) فخر الرازي الماجستير.

الكلمة الرئيسية: خوارزمية Ford-Fulkerson، تدفق الأكثر، الشبكة، شبكة الثنائية

خوارزميات Ford-Fulkerson الذي استُخدم لطلب التدفق الأقصى في الشبكة التي لديها نقطة المصدر و نقطة الهدف. تصويرا لنشر البضائع شبكةً فالشبكة لديها نقاط المصدر و نقاط الهدف العديدة لحل مشاكل التدفق الأقصى في هذه الشبكة فيستخدم تعديل Ford-Fulkerson. والهدف من هذا البحث هو دليل الخطوات التي تبحث عن أقصى التدفق و النتائج المحسولة عليها باستخدام خوارزمية Ford-Fulkerson للبيانات المحاكاة عن توزيع السلع مع خمس وجهة المصدر و خمس نقاط الهدف.

كان نتائج من هذه الدراسة هو شكل معدل من الخوارزمية Ford-Fulkerson، تشكيل شبكة جديدة بإضافة نقطة المصادر الرئيسية ونقطة الوجهة على شبكة رئيسية جديدة، وتشكيل القدرات في القوس من المصادر الرئيسية تشير إلى عدد المصادر وتشكيل القدرات في القوس من بعض نقطة الهدف إلى نقطة الهدف الرئيسي مع قيمة الحد الأقصى وتأتي قيمة السيل الأولي بقدر الصفر. ثم تقصية السيل من نقطة المصادر الرئيسية إلى نقطة الهدف الرئيسي باستخدام خوارزمية Ford-Fulkerson، وهي بتوصيف النقطة، وذلك باستخدام الإجراء العكسي، والسعي لتجد زيادة مساره حتى يتم تسمية جميع النقاط التي قد وصفت وكان نقطة الهدف الرئيسي لم تصف حتى ينتهي هذا التكرار. نظرا على نتائج العملية الحسابية، توجد الحد الأقصى للتدفق على الشبكة التي لا تقسيم، على الشبكة المقسمة، و إنشاء برنامج في Matlab R2010a بقيمة ٤٥. يهتم للبحث التالي أن يبحث أكثر ارتباطا عميقا هذه الدراسة، و يمكن وضعها في شبكة الطيران.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teori graf merupakan salah satu cabang dari ilmu matematika, dimana teori-teori yang digunakan tergolong aplikatif hingga saat ini dan dapat diterapkan untuk memecahkan masalah dalam kehidupan sehari-hari, sehingga teori ini sangat menarik untuk dibahas. Teori graf dapat diperlihatkan peranan dan kegunaannya dalam memecahkan berbagai masalah dengan mengkaji dan menganalisis model atau rumusan. Permasalahan yang dirumuskan dengan teori graf dibuat sederhana, yaitu diambil aspek-aspek yang diperlukan dan dibuang aspek-aspek lainnya (Purwanto, 1998:1). Teori graf membahas berbagai macam permasalahan, salah satu pembahasannya adalah mengenai jaringan (*network*).

Menurut Budayasa (2007:228) jaringan merupakan graf berarah sederhana terhubung lemah yang mempunyai titik sumber dan titik tujuan yang setiap sisinya mempunyai kapasitas. Suatu titik s di jaringan disebut titik sumber jika derajat masuk titik s sama dengan nol dan suatu titik t di jaringan disebut titik tujuan jika derajat keluar titik t sama dengan nol, sedangkan titik yang lain di jaringan disebut titik-titik antara. Permasalahan dalam suatu jaringan dapat dicari penyelesaiannya, salah satunya yaitu dengan mencari *flow* maksimum pada jaringan tersebut.

Definisi *flow* maksimum sesuai dengan namanya merupakan suatu aliran (*flow*) yang digunakan untuk mengetahui nilai *flow* yang maksimum dari seluruh lintasan di dalam suatu jaringan. Hal ini tentunya sangat umum terjadi pada bidang transportasi, produksi, dan distribusi. *Flow* maksimum pada suatu jaringan

misalnya jaringan listrik, jalur lalu lintas, dan pendistribusian barang. Kapasitas pada setiap jaringan akan membatasi jumlah nilai *flow* yang melewatinya. Misalnya sebuah truk pengangkut barang dengan kapasitas 100 kotak tidak akan muat jika dipaksa untuk mengangkut 150 kotak.

Selanjutnya, penelitian ini membahas tentang distribusi barang yang didefinisikan sebagai jaringan. Dalam konsep pendistribusian barang, semakin banyak barang yang dijual maka kemungkinan barang yang laku terjual semakin banyak, sehingga keuntungan yang didapatkan oleh perusahaan juga semakin banyak. Banyaknya barang yang didistribusikan ke toko didefinisikan sebagai *flow*. Jika barang yang didistribusikan ke toko banyak maka barang yang laku terjual juga banyak, namun jika barang yang didistribusikan ke toko sedikit maka barang yang laku terjual di toko juga sedikit. Menurut Mankiw (2006:323), berdasarkan prinsip ekonomi yang berbunyi “*Minimize input, maximize output*” maka permasalahan matematika yang sesuai dalam kasus ini yaitu pencarian *flow* maksimum, sehingga pencarian *flow* maksimum digunakan pada jaringan distribusi barang ini.

Pencarian *flow* maksimum untuk menyelesaikan permasalahan pada jaringan ini juga dijelaskan dalam al-Quran. Sebagaimana tersirat di dalam firman Allah Swt. surah al-Insyirah/94:5-6 yaitu:

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٥﴾ إِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٦﴾

“*Karena sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan*” (QS. al-Insyirah/94:5-6).

Ayat di atas menjelaskan bahwa setelah adanya kesulitan/masalah, datanglah suatu pertolongan Allah Swt. berupa kemudahan/solusi dari masalah tersebut. Pengulangan ayat di atas dimaksudkan untuk menegaskan bahwa Allah Swt. benar-benar menyediakan solusi atau penyelesaian masalah pada setiap

permasalahan yang dialami hamba-Nya. Sebagaimana dijelaskan dalam tafsir Ibnu Katsir yaitu Allah Swt. memberitahukan bahwa bersama kesulitan itu ada kemudahan. Kemudian Dia mempertegas berita tersebut. Ibnu Jarir meriwayatkan dari al-Hasan, dia berkata: “Nabi Saw. pernah keluar rumah pada suatu hari dalam keadaan senang dan gembira, dan beliau juga dalam keadaan tertawa seraya bersabda yang artinya: “*Satu kesulitan itu tidak akan pernah mengalahkan dua kemudahan, satu kesulitan itu tidak akan pernah mengalahkan dua kemudahan, karena bersama kesulitan itu pasti terdapat kemudahan, sesungguhnya bersama kesulitan itu terdapat kemudahan*”. Dengan demikian, dapat dikatakan bahwa kesulitan itu dapat diketahui pada dua keadaan, dimana kalimatnya dalam bentuk *mufrad* (tunggal). Sedangkan kemudahan (*al-yusr*) dalam bentuk *nakirah* (tidak ada ketentuannya) sehingga bilangannya bertambah banyak. Oleh karena itu, beliau bersabda, “*satu kesulitan itu tidak akan mengalahkan dua kemudahan*” (Abdullah, 2007:94).

Sebagaimana makna ayat tersebut, masalah jaringan juga dapat dicari penyelesaiannya, yaitu dengan mencari *flow* maksimumnya. Banyak penelitian yang telah dilakukan oleh praktisi pendidikan dalam pencarian *flow* maksimum ini, salah satunya suatu penelitian yang dilakukan oleh Farizal (2013) yang menjelaskan tentang pencarian aliran (*flow*) maksimum pada jaringan listrik di Kota Tegal menggunakan Algoritma Ford-Fulkerson. Menurut Farizal, Algoritma Ford-Fulkerson ini selalu memberikan hasil yang benar dalam mencari lintasan peningkatan (*augmenting-path*), sehingga pada penelitian ini juga akan digunakan Algoritma Ford-Fulkerson. Penelitian yang dilakukan oleh Farizal tersebut merupakan pencarian *flow* maksimum pada jaringan yang memiliki satu titik

sumber dan satu titik tujuan. Namun masalah muncul ketika jaringan memiliki beberapa titik sumber dan beberapa titik tujuan, sehingga untuk menyelesaikan permasalahan tersebut dapat dilakukan modifikasi dari Algoritma Ford-Fulkerson.

Dalam penelitian ini, penulis menyelesaikan masalah pendistribusian barang yang memiliki beberapa titik sumber dan beberapa titik tujuan dengan mencari *flow* maksimum menggunakan modifikasi Algoritma Ford-Fulkerson. Harapan dari penelitian ini adalah pendistribusian barang dapat dilakukan secara maksimum sehingga keuntungan yang diperoleh perusahaan lebih banyak. Oleh karena itu, judul yang diangkat pada penelitian ini yaitu “Algoritma Ford-Fulkerson untuk Memaksimumkan *Flow* dalam Pendistribusian Barang”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah pada penelitian ini yaitu bagaimana menentukan *flow* maksimum pada jaringan pendistribusian barang?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini yaitu untuk mengetahui *flow* maksimum pada jaringan pendistribusian barang.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian, maka manfaat penelitian yang dapat diambil pada penelitian ini dibedakan berdasarkan kepentingan beberapa pihak yaitu:

1. Bagi Peneliti

Sebagai tambahan pengetahuan dan wawasan tentang penelitian aplikatif graf pada masalah distribusi barang dengan menentukan *flow* maksimum.

2. Bagi Mahasiswa

Sebagai tambahan pengetahuan dan kedalaman keilmuan matematika khususnya aplikasi teori graf pada masalah distribusi barang dengan menggunakan *flow*.

3. Bagi Lembaga

Sebagai tambahan bahan literatur untuk kajian penelitian aplikatif matematika khususnya teori graf dengan menggunakan *flow* dan penerapan pada bidang lainnya.

1.5 Batasan Masalah

Penelitian ini dibatasi pada beberapa hal agar pembahasannya lebih fokus, sehingga batasan masalah pada penelitian ini adalah:

1. Data pendistribusian barang yang digunakan berupa data simulasi.
2. Data yang digunakan hanya berdasarkan banyaknya barang, tanpa memperhatikan variabel lain misalnya jarak, harga, dan *expired* barang.
3. Pendefinisian kapasitas di titik sumber utama merupakan stok barang sedangkan untuk kapasitas di beberapa titik sumber merupakan permintaan dari toko.
4. Pendefinisian nilai *flow* untuk titik sumber utama yaitu barang yang dikirim ke toko tidak akan melebihi stok barang, sedangkan nilai *flow* untuk beberapa titik

sumber yaitu barang yang dikirim ke toko tidak akan melebihi permintaan dari toko.

1.6 Metode Penelitian

1.6.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif dengan mengacu pada data pendistribusian barang yang akan dicari penyelesaiannya dengan menggunakan Algoritma Ford-Fulkerson.

1.6.2 Data dan Sumber Data

Pada penelitian ini data yang digunakan adalah data sekunder yang bersumber dari penulis. Data tersebut berupa data simulasi tentang pendistribusian barang yang meliputi stok persediaan barang yang didefinisikan sebagai kapasitas dari titik sumber utama sebanyak 45 biji, jumlah permintaan barang dari toko didefinisikan sebagai kapasitas dari agen distributor dengan total barang sebanyak 50 biji, banyaknya agen distributor didefinisikan sebagai beberapa titik sumber yaitu lima orang, banyaknya toko tempat pengiriman barang yang dikelompokkan berdasarkan wilayah distribusi tiap kecamatan didefinisikan sebagai beberapa titik tujuan yaitu lima kecamatan, dan tempat penyimpanan barang didefinisikan sebagai titik sumber utama dan titik tujuan utama yaitu suatu perusahaan.

1.6.3 Tahap-tahap Penelitian

Tahap-tahap penelitian dalam penelitian ini dapat dijelaskan dan diuraikan sebagai berikut:

1. Menyediakan data pendistribusian barang yaitu berupa data simulasi.

2. Data simulasi yang disediakan memiliki beberapa titik sumber dan beberapa titik tujuan, sehingga langkah yang dilakukan yaitu memodifikasi Algoritma Ford-Fulkerson agar algoritma ini dapat digunakan untuk mencari *flow* maksimum pada jaringan ini.
3. Membentuk sebuah jaringan pada data simulasi kemudian menyusun *Flow* Maksimum pada data tersebut dengan hasil dari modifikasi Algoritma Ford-Fulkerson serta mencari *flow* maksimum dengan mempartisi tiap jaringan berdasarkan titik sumber.
4. Membuat program di Matlab R2010a yang bertujuan agar pencarian *flow* maksimum dapat diketahui dengan lebih mudah.

1.7 Sistematika Penulisan

Penulisan skripsi ini menggunakan sistematika penulisan yang terdiri dari empat bab, dan masing-masing bab dibagi dalam subbab dengan sistematika penulisan sebagai berikut:

Bab I Pendahuluan

Berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

Bab II Kajian Pustaka

Berisi konsep-konsep dasar graf, graf berarah, jaringan (*Network*), graf bipartisi, jaringan bipartisi, dan kajian agama.

Bab III Pembahasan

Berisi pemaparan data serta pencarian *flow* maksimum.

Bab IV Penutup

Berisi kesimpulan dan saran, dimana kesimpulan merupakan jawaban langsung dari rumusan masalah yang diajukan, sedangkan saran dibuat untuk penelitian selanjutnya.

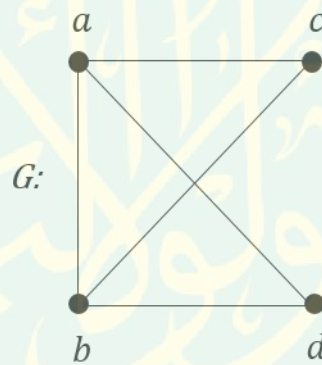


BAB II

KAJIAN PUSTAKA

2.1 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). Himpunan titik dari G dinotasikan dengan $V(G)$, sedangkan himpunan sisi dinotasikan dengan $E(G)$, sebagaimana dicontohkan pada gambar berikut.



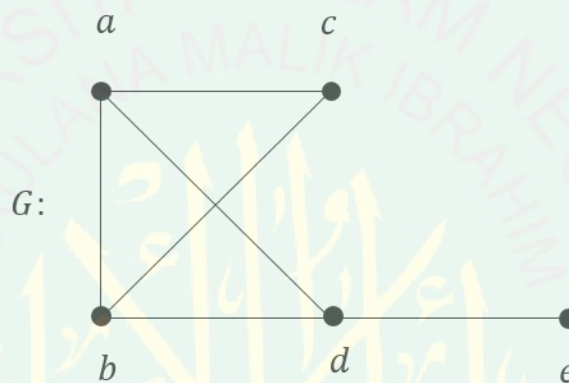
Gambar 2.1 Contoh Graf G

Graf G pada Gambar 2.1 dapat dinyatakan dengan $G = (V(G), E(G))$ dengan $V(G) = \{a, b, c, d\}$ dan $E(G) = \{ab, ad, ac, bc, bd\}$ (Chartrand dan Lesniak, 1996:1).

Selain itu Abdussakir, dkk (2009:4) mengatakan bahwa “Graf G adalah pasangan $(V(G), E(G))$ dengan $V(G)$ adalah himpunan tidak kosong dan berhingga dari objek-objek yang disebut titik, dan $E(G)$ adalah himpunan (mungkin kosong) pasangan tak berurutan dari titik-titik berbeda di $V(G)$ yang disebut sisi. Banyaknya unsur di $V(G)$ disebut order dari G dan dilambangkan dengan $p(G)$, dan banyaknya

unsur di $E(G)$ disebut ukuran dari G dan dilambangkan dengan $q(G)$. Jika graf yang dibicarakan hanya graf G , maka order dan ukuran dari G masing-masing cukup ditulis p dan q . Graf dengan order p dan ukuran q dapat disebut graf- (p, q) ”.

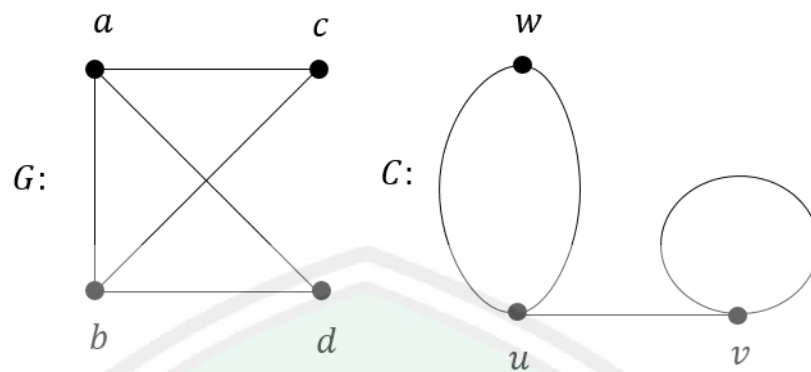
Suatu graf G yang memuat himpunan titik $V(G)$ dan himpunan sisi $E(G)$ yaitu $V(G) = \{a, b, c, d, e\}$, $E(G) = \{(a, b), (a, c), (a, d), (b, d), (b, c), (d, e)\}$. Graf G tersebut dapat digambarkan sebagai berikut.



Gambar 2.2 Graf G

Graf G mempunyai 5 titik sehingga banyak titik G adalah $p = 5$. Graf G mempunyai 6 sisi sehingga ukuran graf G adalah $q = 6$ (Abdussakir, dkk, 2009:5).

Suatu sisi graf yang menghubungkan suatu 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 (*multiple-edges*). Graf yang tidak mempunyai sisi rangkap dan tidak memiliki gelung disebut graf sederhana, sedangkan suatu graf yang memiliki sisi rangkap tetapi tidak memiliki gelung disebut graf rangkap (*multi-graph*), sebagaimana ditunjukkan pada gambar berikut.



Gambar 2.3 Graf G Merupakan Graf Sederhana; Graf C Bukan Graf Sederhana

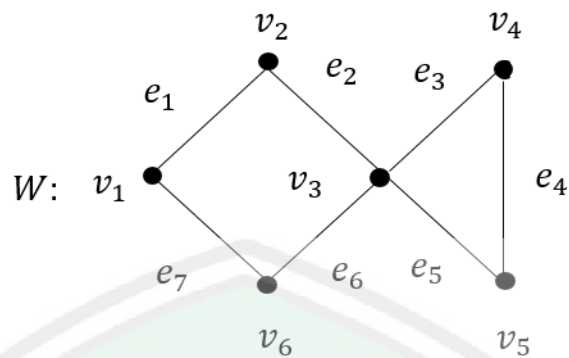
(Budayasa, 2007:3).

2.1.1 Lintasan pada Graf

Misalkan G suatu graf. Misalkan u dan v adalah titik di G (yang tidak harus berbeda). Jalan $u - v$ pada graf G adalah barisan berhingga yang berselang-seling

$$W: u = u_0, e_1, u_1, e_2, \dots, u_{k-1}, e_k, u_k = v$$

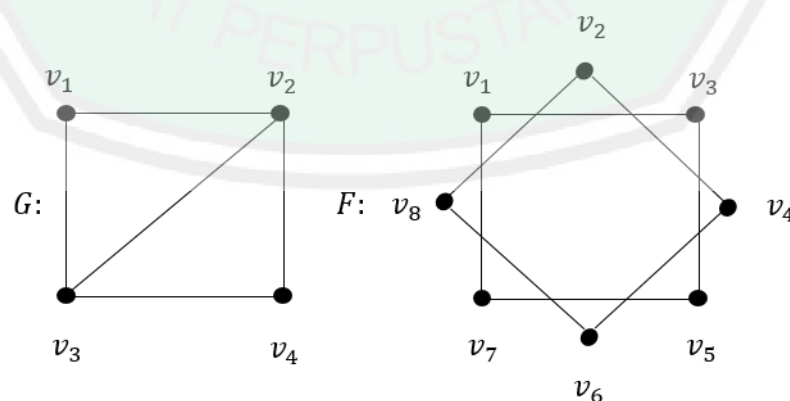
antara titik dan sisi, yang dimulai dari titik u dan diakhiri dengan titik v , dengan $e_i = u_{i-1}u_i$ untuk $i = 1, 2, 3, \dots, k$ adalah sisi di G . Titik u_0 disebut titik awal, titik u_k disebut titik akhir, titik u_1, u_2, \dots, u_{k-1} disebut titik internal, dan bilangan k disebut panjang dari W (Chartrand dan Lesniak, 1996:16). Jika $u_0 \neq u_k$ maka W disebut jalan terbuka. Jika $u_0 = u_k$ maka W disebut jalan tertutup. Jalan yang tidak mempunyai sisi disebut jalan *trivial*. Suatu jejak $u - v$ adalah jalan $u - v$ yang tidak mengulang sisi, sedangkan lintasan $u - v$ adalah jalan $u - v$ yang tidak mengulang titik, sebagaimana ditunjukkan pada gambar berikut.

Gambar 2.4 Jalan dan Lintasan pada Graf W

Gambar 2.4 menyatakan sebuah jalan dari titik v_1 sampai titik v_5 dan urutannya adalah $(v_1, e_7, v_6, e_6, v_3, e_3, v_4, e_4, v_5)$. Sedangkan lintasan dari titik v_1 sampai titik v_5 urutannya adalah $(v_1, e_1, v_2, e_2, v_3, e_5, v_5)$ (Chartrand dan Lesniak, 1996:17).

2.1.2 Graf Terhubung

Budayasa (2007:8) mengatakan bahwa “Dengan menggunakan konsep lintasan, maka dapat didefinisikan keterhubungan suatu graf. Suatu graf G dikatakan terhubung (*connected*) jika untuk setiap dua titik G yang berbeda terdapat suatu lintasan yang menghubungkan kedua titik tersebut”, sebagaimana dicontohkan pada gambar berikut.

Gambar 2.5 Graf G Terhubung; Graf F Tidak Terhubung

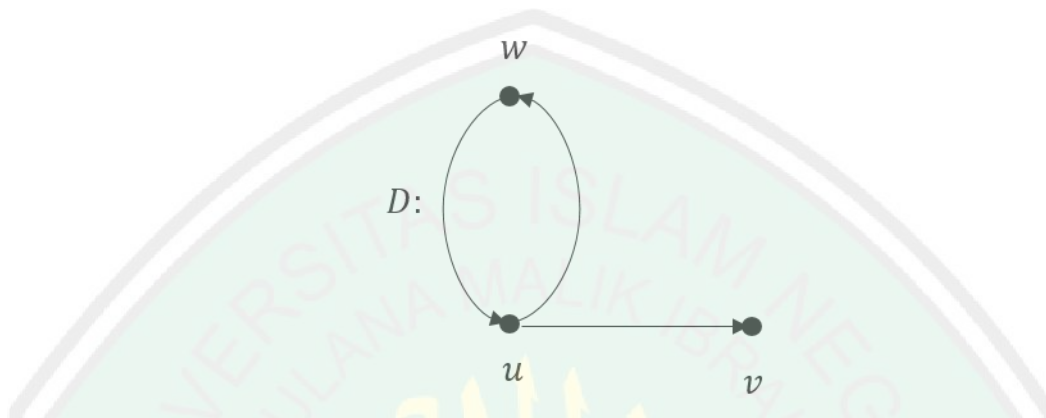
Gambar 2.5 menunjukkan bahwa graf G merupakan graf terhubung dari v_1 sampai v_4 urutannya adalah (v_1, v_2, v_3, v_4) . Sedangkan graf F merupakan graf tak terhubung karena terdapat dua titik yang tidak terhubung dari v_1 sampai v_8 .

2.2 Graf Berarah atau 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. Kata “digraf” merupakan adopsi dari kependekan kata “*directed graph*”. Seperti pada graf, himpunan titik pada digraf D dinotasikan dengan $V(D)$, dan himpunan busur dinotasikan dengan $A(D)$. Banyaknya unsur di $V(D)$, yakni $|V(D)|$ disebut order dari D dan dinotasikan dengan $p(D)$, sedangkan banyaknya unsur di $A(D)$, yakni $|A(D)|$, disebut ukuran dari D dan dinotasikan dengan $q(D)$. Jika digraf yang dibicarakan hanya digraf D , maka order dan ukuran dari D masing-masing cukup ditulis p dan q . Digraf dengan order p dan ukuran q dapat disebut digraf- (p, q) (Chartrand dan Lesniak, 1996:25).

Jika $a = (u, v)$ adalah busur pada digraf D , maka a dikatakan memasangkan u dan v . Selanjutnya, dikatakan a terkait langsung dari (*incident from*) u dan terkait langsung ke (*incident to*) v . Selain itu, dikatakan u terhubung langsung ke (*adjacent to*) v sedangkan v terhubung langsung dari (*adjacent from*) u . Dua titik u dan v pada digraf D dikatakan tidak terhubung langsung jika u tidak terhubung langsung ke v atau u tidak terhubung langsung dari v . Pada digraf D , busur (u, v) tidak sama dengan busur (v, u) . Selanjutnya busur (u, v) akan ditulis uv (Chartrand dan Lesniak, 1996:26).

Misalkan D suatu digraf dengan himpunan titik $V(D) = \{u, v, w\}$ dan himpunan busur $A(D) = \{(u, w), (w, u), (u, v)\}$, maka D dapat digambarkan sebagai berikut.



Gambar 2.6 Contoh Digraf D

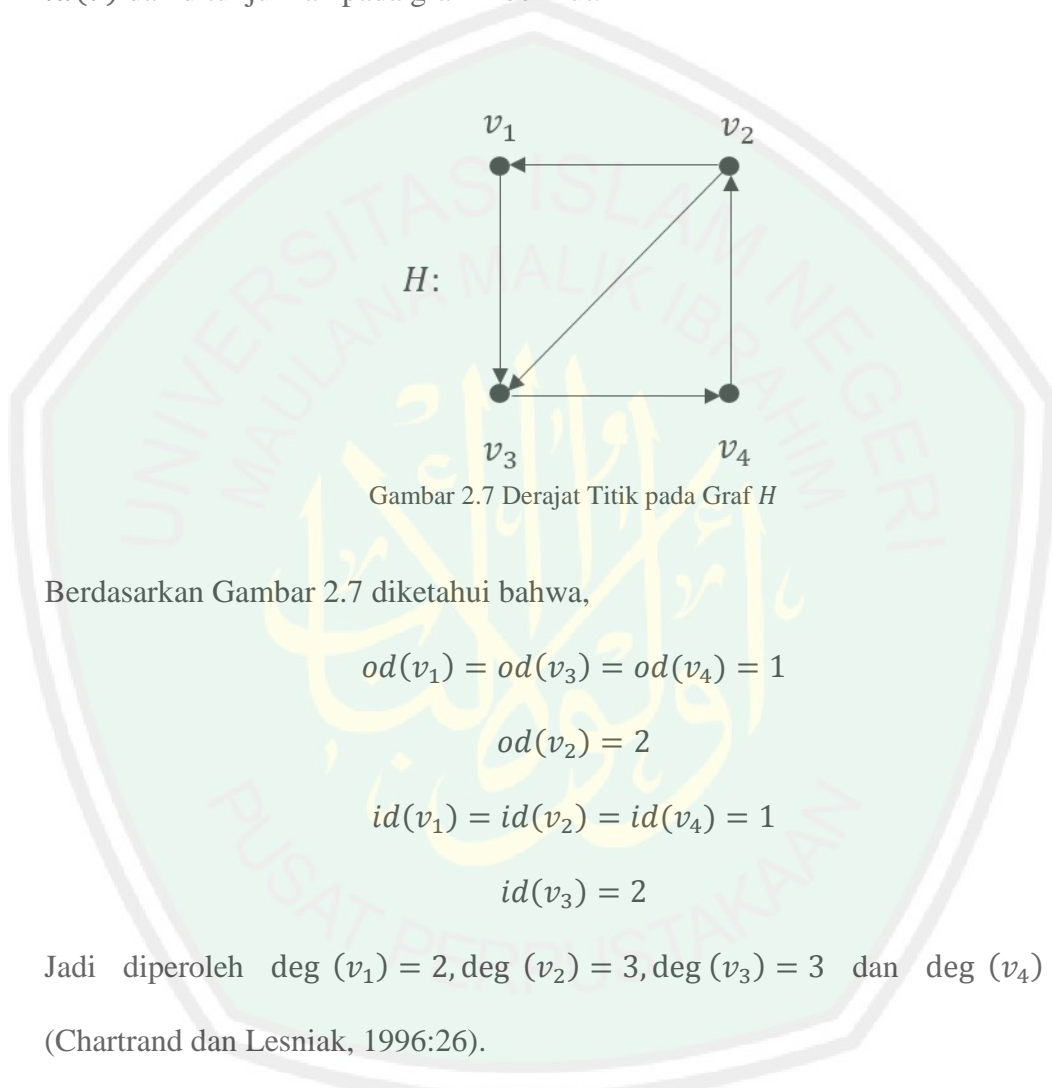
Gambar 2.6 menyatakan bahwa titik u terhubung langsung ke v , sedangkan titik v tidak terhubung langsung ke u (Chartrand dan Lesniak, 1996:25).

Konsep graf berarah sederhana serupa dengan graf sederhana, hal yang membedakan adalah sisi di graf sederhana berubah menjadi busur. Sebuah busur graf yang menghubungkan sebuah titik dengan dirinya sendiri disebut gelang (*loop*). Jika terdapat lebih dari satu busur yang menghubungkan dua titik pada suatu graf, maka busur-busur tersebut disebut busur rangkap atau busur ganda (*multiple-edges*). Graf yang tidak mempunyai busur rangkap dan tidak memiliki gelang disebut graf berarah sederhana. Sedangkan sebuah graf yang memiliki busur rangkap tetapi tidak memiliki gelang disebut graf rangkap (*multi-graph*). Contoh digraf D pada Gambar 2.6 bukan merupakan graf berarah sederhana.

2.2.1 Derajat Titik pada Graf Berarah

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 $od(v)$, sedangkan derajat titik v pada digraf D , dinotasikan dengan $deg(v)$, didefinisikan dengan $deg(v) = od(v) + id(v)$ dan ditunjukkan pada graf H berikut.



Berdasarkan Gambar 2.7 diketahui bahwa,

$$od(v_1) = od(v_3) = od(v_4) = 1$$

$$od(v_2) = 2$$

$$id(v_1) = id(v_2) = id(v_4) = 1$$

$$id(v_3) = 2$$

Jadi diperoleh $deg(v_1) = 2, deg(v_2) = 3, deg(v_3) = 3$ dan $deg(v_4) = 2$ (Chartrand dan Lesniak, 1996:26).

Teorema 1: Misalkan D digraf berorder p dan berukuran q dengan himpunan titik

$$V(D) = \{v_1, v_2, \dots, v_p\}, \text{ maka } \sum_{i=1}^p od(v_i) = \sum_{i=1}^p id(v_i) = q.$$

Bukti: Setiap menghitung derajat keluar suatu titik, maka masing-masing busur dihitung satu kali, karena setiap busur terkait langsung dari tepat satu titik. Demikian juga, setiap menghitung derajat masuk suatu titik, maka masing-

masing busur dihitung satu kali, karena setiap busur terkait langsung ke tepat satu titik (Abdussakir, dkk, 2009:89).

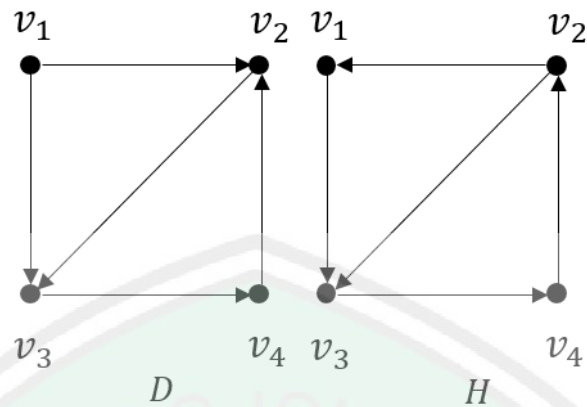
2.2.2 Digraf Terhubung

Konsep jalan pada digraf D analog pada graf G . Hal yang membedakan pada digraf adalah arah busur pada jalan, jejak, lintasan, sirkuit, dan sikel harus diperhatikan (Chartrand dan Lesniak, 1996:29). Misalkan D sebuah digraf. Misalkan u dan v adalah titik di D (yang tidak harus berbeda). Jalan $u - v$ pada graf D adalah barisan berhingga yang berselang-seling

$$W: u = u_0, a_1, u_1, a_2, \dots, u_{k-1}, a_k, u_k = v$$

antara titik dan busur, yang dimulai dari titik u dan diakhiri dengan titik v , dengan $a_i = u_{i-1}u_i$ untuk $i = 1, 2, 3, \dots, k$ adalah busur di D . Titik u_0 disebut titik awal, titik u_k disebut titik akhir, titik u_1, u_2, \dots, u_{k-1} disebut titik internal, dan bilangan k menyatakan panjang dari W . Jika $u_0 \neq u_k$ maka W disebut jalan terbuka. Jika $u_0 = u_k$ maka W disebut jalan tertutup. Jalan yang tidak mempunyai busur disebut jalan *trivial*.

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, sebagaimana ditunjukkan pada gambar berikut.

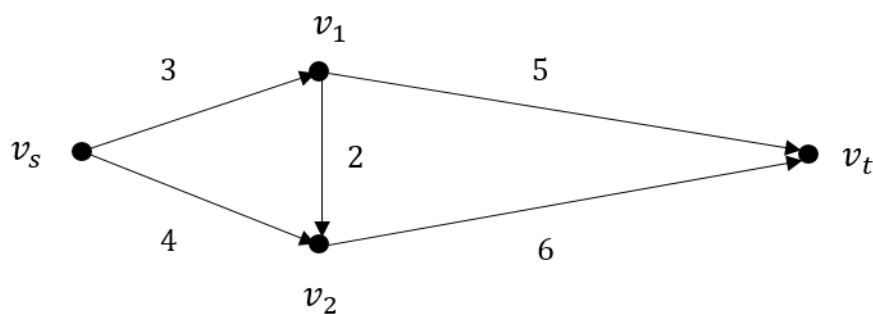


Gambar 2.8 Graf D Terhubung Lemah; Graf H Terhubung Kuat

(Chartrand dan Lesniak, 1996:29).

2.3 Pengertian Jaringan

Sebuah jaringan $N = (V(N), \Gamma(N))$ adalah 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) atau disingkat (i, j) , pada jaringan N disebut kapasitas busur (V_i, V_j) dan dilambangkan dengan $c(V_i, V_j)$ atau disingkat $c(i, j)$. 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. Sebuah contoh jaringan N dengan sumber V_s dan titik tujuan V_t dapat dilihat pada gambar berikut.



Gambar 2.9 Jaringan N dengan Titik Sumber v_s dan Titik Tujuan v_t

Titik V_1 dan V_2 adalah titik-titik antara pada jaringan N . Kapasitas busur (v_s, v_1) adalah $c(s, 1) = 3$, sedangkan kapasitas busur (v_1, v_2) adalah $c(1, 2) = 2$. Selanjutnya dapat ditulis $c(s, 2) = 4, c(1, t) = 5$ dan $c(2, t) = 6$ (Budayasa, 2007:228).

Misalkan X dan Y dua himpunan bagian $V(N)$ pada jaringan N . Himpunan semua busur N yang berawal di X dan berujung di Y dilambangkan dengan $B(X, Y)$. Total kapasitas semua busur di $B(X, Y)$ dilambangkan dengan $c(X, Y)$. Dengan demikian,

$$c(X, Y) = \sum_{a \in B(X, Y)} c(a)$$

Misalkan pada Gambar 2.9, jika $X = \{V_s, V_1\}$ dan $Y = \{V_2, V_t\}$, maka $B(X, Y) = \{(s, 2), (1, 2), (1, t)\}$ dan $c(X, Y) = c(s, 2) + c(1, 2) + c(1, t) = 4 + 2 + 5 = 11$. Misalkan $B(X, Y) = \emptyset$ dan $c(X, Y) = 0$. Jika $X = \{v_1, v_t\}$ dan $Y = \{v_2\}$, maka $B(X, Y) = \{(1, 2)\}$ dan $c(X, Y) = c(1, 2) = 2$, sedangkan $B(Y, X) = \{(v_2, v_t)\}$ dengan $c(Y, X) = c(2, t) = 6$ (Budayasa, 2007:228).

2.3.1 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 . Disebut demikian, karena penghapusan semua busur $B(X, X_1)$ dari N , memutus semua lintasan berarah dari titik s ke titik t pada jaringan N . Misalkan A adalah himpunan titik antara pada jaringan N dan A' adalah sebuah himpunan bagian A . Jika $X = \{t\} \cup A'$, maka $B(X, X_1)$ sebuah pemutus— (s, t) pada N . Jadi banyaknya pemutus— (s, t) pada jaringan N sama

dengan banyaknya himpunan bagian dari himpunan A yaitu 2^n dengan $n = |A|$. Misalkan jaringan N pada Gambar 2.9 memiliki dua titik antara yaitu v_1 dan v_2 , sehingga terdapat $2^2 = 4$ pemutus- (s, t) pada N , yaitu:

$$B(\{v_s\}, \{v_1, v_2, v_t\}) = \{(s, 1), (s, 2)\}$$

$$B(\{v_s, v_1\}, \{v_2, v_t\}) = \{(s, 2), (1, 2), (1, t)\}$$

$$B(\{v_s, v_2\}, \{v_1, v_t\}) = \{(s, 1), (2, t)\}$$

$$B(\{v_s, v_1, v_2\}) = \{(1, t), (2, t)\}$$

Setiap pemutus- (s, t) pada jaringan N mempunyai kapasitas. Pemutus- (s, t) yang mempunyai kapasitas terkecil disebut pemutus- (s, t) minimum. Masing-masing kapasitas dari keempat pemutus tersebut adalah:

$$c(\{v_s\}, \{v_1, v_2, v_t\}) = c(s, 1) + c(s, 2) = 3 + 4 = 7$$

$$c(\{v_s, v_1\}, \{v_2, v_t\}) = c(s, 2) + c(1, 2) + c(1, t) = 4 + 2 + 5 = 11$$

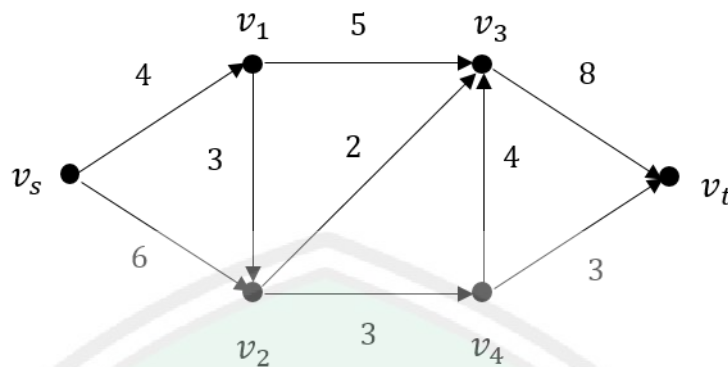
$$c(\{v_s, v_2\}, \{v_1, v_t\}) = c(s, 1) + c(2, t) = 3 + 6 = 9$$

$$c(\{v_s, v_1, v_2\}) = c(1, t) + c(2, t) = 5 + 6 = 11$$

Busur pemutus $B(\{v_s\}, \{v_1, v_2, v_t\}) = \{(s, 1), (s, 2)\}$ dengan kapasitas 7 merupakan sebuah pemutus- (s, t) minimum pada jaringan N (Budayasa, 2007:229).

2.3.2 Konsep *Flow* pada Jaringan

Misalkan N adalah sebuah jaringan dengan titik sumber s dan titik tujuan t . Jika v adalah sebuah titik di N , maka himpunan semua busur N yang keluar dari titik v (meninggalkan v) dilambangkan dengan $O(v)$ dan himpunan semua busur N yang menuju titik v dilambangkan dengan $I(v)$. Misalkan jaringan N dengan sumber s dan tujuan t pada gambar berikut.



Gambar 2.10 Jaringan N dengan Titik Sumber v_s dan Titik Tujuan v_t

Terdapat dua busur N yang keluar dari titik v_s yaitu busur (v_s, v_1) dan busur (v_s, v_2) , dan tidak ada busur N yang menuju ke titik v_s , sehingga $O(v_s) = \{(v_s, v_1), (v_s, v_2)\}$ dan $I(v_s) = \{\}$. Penulisan label titik cukup ditulis indeks label titik tersebut, titik v_s cukup ditulis titik s , busur (v_1, v_3) cukup ditulis $(1,3)$, dengan demikian, $O(v_s) = O(s) = \{(s, 1), (s, 2), \}$ dan $I(v_s) = I(s) = \{\}$. Selanjutnya diperoleh $O(1) = O(v_1) = \{(v_1, v_2), (v_1, v_3)\}$ dan $I(1) = I(v_1) = \{(v_s, v_1)\} = \{(s, 1)\}$, demikian juga $O(3) = \{(3, t)\}$ dan $I(3) = \{(1,3), (2,3), (4,3)\}$ (Budayasa, 2007:230).

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:

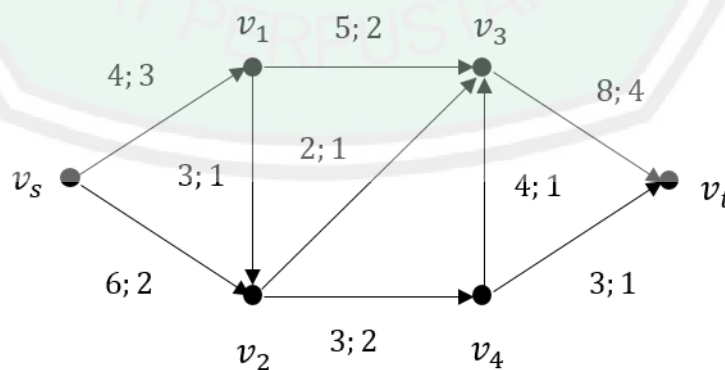
- (i) $0 \leq f(i, j) \leq c(i, j) \forall (i, j) \in \Gamma(N)$ (disebut kapasitas pembatas)
- (ii) $\sum_{(i,j) \in O(s)} f(i, j) = \sum_{(i,j) \in I(t)} f(i, j)$ (disebut nilai *flow* f)
- (iii) $\sum_{(i,j) \in O(s)} f(i, j) = \sum_{(i,j) \in I(t)} f(i, j) \forall x \in V(N) - \{s, t\}$ (disebut sebagai “konservasi *flow*”) (Budayasa, 2007:231).

Syarat (i) menyatakan bahwa nilai *flow* pada setiap busur N tidak pernah melebihi kapasitas busur tersebut. Syarat (ii) menyatakan bahwa total nilai *flow*

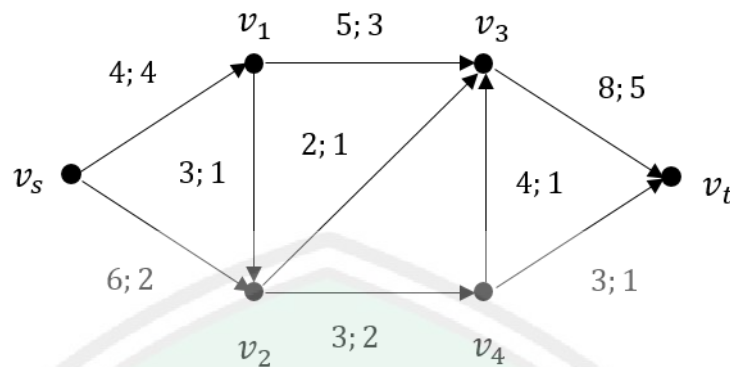
yang keluar dari titik sumber s sama dengan total nilai *flow* yang sampai di titik tujuan. Nilai yang selanjutnya disebut “nilai *flow* f ” dari s ke t pada jaringan N . Syarat (iii) menyatakan bahwa untuk setiap titik antara pada N berlaku total *flow* yang meninggalkan x sama dengan total *flow* yang menuju titik x . Jika nilai *flow* f dari titik sumber s ke titik tujuan t pada jaringan N dimisalkan $f_{s,t}$ maka syarat (ii) dan (iii) di atas dapat ditulis sebagai:

$$f(i, V) - f(V, i) = \begin{cases} f_{s,t} & \text{jika } i = s \\ 0 & \text{jika } i \neq s, t \\ -f_{s,t} & \text{jika } i = t \end{cases}$$

Misalkan jaringan N pada Gambar 2.10. Sebuah *flow* f dari titik sumber s ke titik tujuan t dengan nilai 5 dapat dilihat pada Gambar 2.11, dan sebuah *flow* f_1 dengan nilai 6 dapat dilihat pada Gambar 2.12. Urutan pertama dari pasangan bilangan yang dikaitkan pada suatu busur N menyatakan kapasitas busur, dan urutan kedua dalam pasangan tersebut menyatakan nilai *flow* yang mengalir pada busur tersebut, sebagaimana berikut.



Gambar 2.11 *Flow* f pada N dengan Nilai 5

Gambar 2.12 Flow f_1 pada N dengan Nilai 6

Nilai *flow* f_1 lebih besar dari nilai *flow* f pada jaringan N . Selanjutnya untuk *flow* f di jaringan N pada Gambar 2.11, jika $X = \{v_s, v_1\} = \{s, 1\}$ dan $X_1 = \{2, 3, 4, t\}$, maka $B(X, X_1) = \{(s, 2), (1, 2), (1, 3)\}$ adalah sebuah pemutus- (s, t) pada N dengan kapasitas $c(X, X_1) = c(s, 2) + c(1, 2) + c(1, 3) = 6 + 3 + 5 = 14$. Nilai *flow* f pada jaringan N yaitu 5, dimana nilai tersebut tidak melebihi kapasitas pemutus- (s, t) $B(X, X_1)$. Nilai *flow* f_1 pada Gambar 2.12 yaitu 6 dimana nilai tersebut tidak melebihi kapasitas pemutus- (s, t) $B(X, X_1)$. Hal ini berlaku untuk sebarang *flow* dan sebarang pemutus- (s, t) pada jaringan N .

Teorema : Misalkan N sebuah jaringan dengan titik sumber s dan titik tujuan t .

Jika f adalah sebuah *flow* dari s ke t pada N dengan nilai $f_{s,t}$ dan $B(X, X_1)$ sebuah pemutus- (s, t) pada N , maka $f_{s,t} = f(X, X_1) - f(X_1, X) \leq c(X, X_1)$.

Bukti : Dari definisi *flow*, untuk sumber s diperoleh

$$f(\{s\}, V) - f(V, \{s\}) = f_{s,t}$$

dan untuk setiap titik $x \in V - \{s, t\}$, diperoleh

$$f(\{x\}, V) = \sum_{(i,j) \in O(x)} f(i,j) = \sum_{(i,j) \in I(x)} f(i,j) = f(V, \{x\})$$

atau

$$f(\{x\}, V) - f(V, \{x\}) = 0$$

Sehingga untuk suatu pemutus (s, t) $B(X, X_1)$ diperoleh

$$\begin{aligned} f(X, V) - f(V, X) &= \sum_{x \in X} [f(\{x\}, V) - f(V, \{x\})] \\ &= [f(\{s\}, V) - f(V, \{s\})] + 0 \\ &= f_{s,t} \end{aligned} \tag{2.1}$$

Sedangkan,

$$\begin{aligned} f(X, V) - f(V, X) &= f(X, X \cup X_1) - f(X \cup X_1, X) \\ &= f(X, X) + f(X, X_1) \\ &\quad - \{f(X, X) + f(X_1, X)\} \\ &= f(X, X_1) - f(X_1, X) \end{aligned} \tag{2.2}$$

Karena nilai *flow* pada setiap busur non-negatif, maka $f(X_1, X) \geq 0$, sehingga

$$f(X, X_1) - f(X_1, X) \leq f(X, X_1) \tag{2.3}$$

Selanjutnya, karena nilai *flow* pada setiap busur N tidak melebihi kapasitas busur, maka

$$f(X, X_1) \leq c(X, X_1) \tag{2.4}$$

Dari (2.1), (2.2), (2.3) dan (2.4) disimpulkan

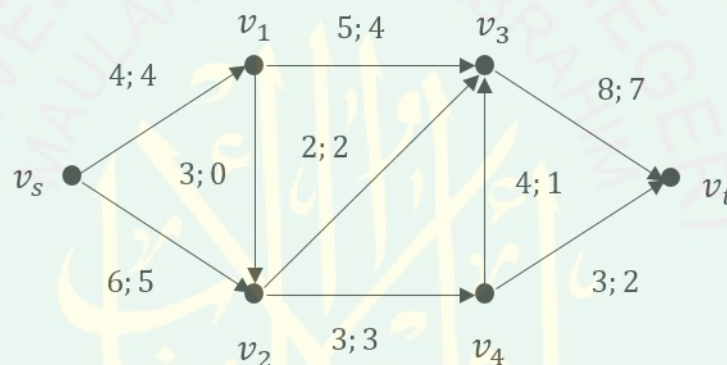
$$f_{s,t} = f(X, X_1) - f(X_1, X) \leq f(X, X_1) \leq c(X, X_1)$$

dengan demikian bukti teorema lengkap (Budayasa, 2007:234).

2.3.3 Flow Maksimum pada Jaringan

Teorema dalam subbab sebelumnya menjamin bahwa nilai sebarang *flow* pada suatu jaringan N dari titik sumber s ke titik tujuan t , tidak akan melebihi kapasitas sebarang pemutus (s, t) , sehingga menurut Budayasa (2008) sebesar-

besarnya nilai *flow* tidak akan melebihi sekecil-kecilnya nilai kapasitas pemutus (s, t) . Jika terdapat sebuah *flow* f di N yang nilainya sama dengan kapasitas suatu pemutus (s, t) , maka *flow* tersebut adalah *flow* maksimum dan pemutus (s, t) tersebut adalah sebuah pemutus (s, t) minimum. Jadi *flow* f bernilai $f_{s,t}$ dari titik sumber s ke titik tujuan t pada jaringan N dikatakan *flow* maksimum jika $f_{s,t} = \min\{c(X, X_1) | B(X, X_1) \text{ suatu pemutus } (s, t) \text{ pada jaringan } N\}$, sebagaimana ditunjukkan pada gambar berikut.



Gambar 2.13 *Flow* Maksimum dengan Nilai *Flow* $f_2 = 9$

Flow f_2 pada Gambar 2.13 bernilai 9. Jika $X = \{s, 2\}$ dan $X_1 = \{(s, 1), (2, 3), (2, 4)\}$ adalah pemutus (s, t) pada N dengan kapasitas $c(X, X_1) = c(s, 1) + c(2, 3) + c(2, 4) = 4 + 2 + 3 = 9$ merupakan nilai *flow* f_2 . Jadi f_2 adalah *flow* maksimum dan $B(X, X_1) = \{(s, 1), (2, 3), (2, 4)\}$ adalah pemutus (s, t) minimum pada jaringan N (Budayasa, 2007:235).

Dalam suatu jaringan selalu terdapat *flow* yang nilainya sama dengan kapasitas pemutus minimum, yang dikenal dengan sebutan “Teorema *Flow* Maksimum Pemutus Minimum”. Hal ini memerlukan beberapa konsep yaitu misalkan N suatu jaringan dan G adalah graf dasar N , misalkan pada graf G terdapat lintasan $P = (v_1, v_2, \dots, v_i, v_{i+1}, \dots, v_n)$, jika (v_i, v_{i+1}) sebuah busur pada N , maka

busur tersebut dinamakan busur maju terhadap P , sebaliknya jika (v_{i+1}, v_i) sebuah busur pada N , maka busur tersebut dinamakan busur balik terhadap P . Jadi suatu busur pada N termasuk busur maju atau busur balik sangat tergantung pada lintasan P pada graf dasarnya. Misalkan G graf dasar dari jaringan N pada Gambar 2.11 di atas, maka $P = (v_s, v_2, v_1, v_3, v_4, v_t)$ adalah sebuah lintasan (v_s, v_t) pada G dan P bukan lintasan berarah pada N . Sehingga terhadap P , busur-busur (v_s, v_2) , (v_1, v_3) , dan (v_t, v_4) merupakan busur-busur maju, sedangkan busur-busur (v_1, v_2) dan (v_4, v_3) merupakan busur-busur balik. Jika diperhatikan lintasan $P_1 = (v_s, v_1, v_2, v_4, v_t)$ pada G , maka semua busur N yang berkorespondensi dengan sisi lintasan P_1 merupakan busur maju terhadap P_1 . Busur (v_1, v_2) pada N merupakan busur balik terhadap lintasan P , tetapi busur tersebut merupakan busur maju terhadap lintasan P_1 (Budayasa, 2007:235).

Misalkan f adalah sebuah *flow* dari titik sumber s ke titik tujuan t pada jaringan N , dan misalkan G adalah graf dasar N . Misalkan sebuah lintasan P pada G . ‘Inkremen’ sebuah busur a pada N yang berkorespondensi dengan sebuah sisi P pada G , dilambangkan dengan $i(a)$, dan didefinisikan sebagai $i(a) = c(a) - f(a)$ jika a busur maju dan $i(a) = f(a)$ jika a busur balik. Selanjutnya inkremen lintasan P , disimbolkan $i(P)$ dan didefinisikan $i(P) = \min \{i(a) \mid a \text{ adalah busur } N \text{ yang bersesuaian dengan sisi } P\}$. Sebuah lintasan P dengan $i(P)$ positif disebut lintasan *augmentasi*. Selanjutnya, lintasan *augmentasi* P dari titik sumber s ke titik tujuan t dinamakan sebuah lintasan peningkatan. Misalkan contoh *flow* f dari titik sumber s ke titik tujuan t di jaringan N pada Gambar 2.11 di atas. Misalkan lintasan $P = (v_s, v_2, v_1, v_3, v_4, v_t)$ pada graf dasar N , karena $(s, 2)$, $(1,3)$ dan $(4, t)$ busur-busur maju dan $(1,2)$, $(4,3)$ busur-busur balik, maka:

$$i(s, 2) = c(s, 2) - f(s, 2) = 6 - 2 = 4$$

$$i(1, 3) = c(1, 3) - f(1, 3) = 5 - 2 = 3$$

$$i(4, t) = c(4, t) - f(4, t) = 3 - 1 = 2$$

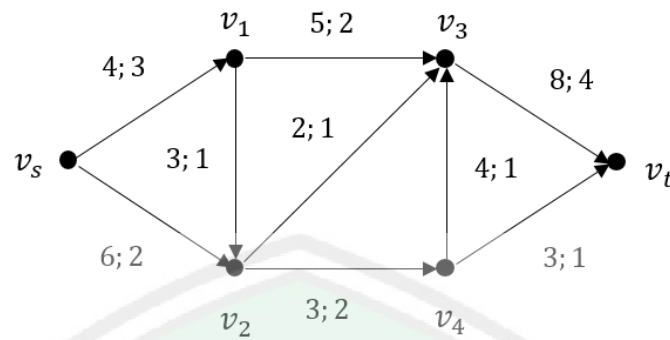
$$i(1, 2) = f(1, 2) = 1$$

$$i(4, 3) = f(4, 3) = 1$$

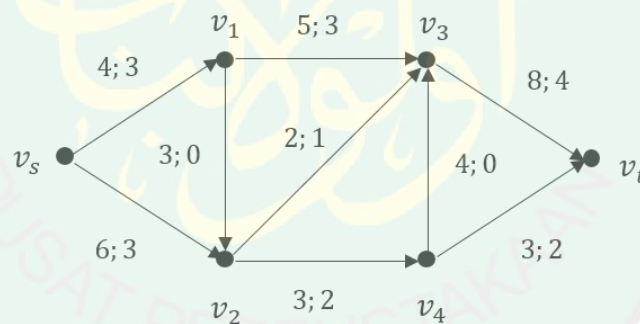
sehingga inkremen P adalah $i(P) = \min \{4, 3, 2, 1, 1\} = 1$. Lintasan dari titik sumber s ke titik tujuan t dan $i(P)$ positif, sehingga P merupakan sebuah lintasan peningkatan. Dinamakan lintasan peningkatan, karena berdasarkan lintasan tersebut dapat dibuat *flow* baru yang nilainya lebih besar dari nilai *flow* sebelumnya (Budayasa, 2007:236). Hal ini dijamin oleh lemma berikut.

Lemma: Misalkan f sebuah *flow* bernilai $f_{s,t}$ dari titik sumber s ke titik tujuan t pada jaringan N . Jika terdapat lintasan P dari titik s ke titik t dengan $i(P) = \delta > 0$, definisikan fungsi f_1 pada himpunan $\Gamma(N)$ sebagai $f_1(a) = f(a) + \delta$, jika a busur maju terhadap P , $f_1(a) = f(a) - \delta$ jika a busur balik terhadap P , dan $f_1(a) = f(a)$ jika busur a yang lainnya, maka f_1 adalah *flow* dari titik s ke titik t pada N dengan nilai $f_{s,t} + \delta$.

Bukti: Diberikan suatu ilustrasi penerapan lemma tersebut, misalnya *flow* f di jaringan N pada Gambar 2.10 di atas. Disebutkan sebelumnya terhadap *flow* jaringan f , lintasan $P = (v_s, v_2, v_1, v_3, v_4, v_t)$ pada graf dasar N adalah sebuah lintasan peningkatan dengan $i(P) = 1$. Busur-busur $(s, 2)$, $(1, 3)$ dan $(4, t)$ merupakan busur-busur maju terhadap lintasan P dan $i(P) = 1$, sehingga nilai *flow* di setiap busur tersebut ditambah 1, sebagaimana gambar berikut.

Gambar 2.14 Lintasan Peningkatan Terhadap *Flow f*

Busur-busur $(2,1)$, $(3,4)$ adalah busur-busur balik, sehingga nilai *flow* di setiap busur tersebut dikurang 1, sedangkan nilai *flow* di setiap busur yang lainnya tetap, sehingga diperoleh *flow* baru, misalnya f' , dengan nilai = nilai $f + i(P) = 5 + 1 = 6$. *Flow f'* yang diperoleh dengan merevisi *flow f* berdasarkan lintasan P , sebagaimana ditunjukkan pada gambar berikut.

Gambar 2.15 *Flow f'* Diperoleh dengan Merevisi *Flow f*

(Budayasa, 2007:237).

2.3.4 Algoritma *Flow* Maksimum pada Jaringan

Untuk sebuah jaringan selalu terdapat sebuah *flow* yang maksimum. Selanjutnya untuk mengkonstruksi sebuah *flow* maksimum pada suatu jaringan dengan satu titik sumber dan satu titik tujuan yaitu menggunakan Algoritma Ford-Fulkerson. Ide utama dari algoritma ini adalah sebagai berikut:

1. Dimulai dengan mengkonstruksi sebuah *flow* f sebarang pada jaringan N dengan titik sumber s dan titik tujuan t . Hal ini selalu dapat dilakukan, misalnya dimulai dengan *flow* nol yaitu *flow* yang mengaitkan setiap busur N dengan bilangan nol.
2. Dari *flow* f tersebut, dicari lintasan P dari titik s ke titik t pada graf dasar G dari N yang inkremennya positif. Jika tidak ada lintasan yang demikian, maka iterasi dihentikan dan *flow* f adalah *flow* maksimum di N , jika ditemukan lintasan P yang demikian, lanjut ke langkah 3.
3. Dibuat *flow* baru misalnya f_1 , dari *flow* f berdasarkan lintasan P tersebut. Nilai *flow* f_1 lebih besar dari pada nilai *flow* f , sehingga *flow* f diganti menjadi *flow* f_1 , kemudian kembali ke langkah 2.

Prosedur untuk mengkonstruksi *flow* baru yang nilainya lebih besar dari nilai *flow* yang sama sudah tersirat pada lemma sebelumnya, namun prosedur untuk mendapatkan lintasan-peningkatan P tidak tersirat dalam lemma maupun bukti teorema sebelumnya. Untuk mendapatkan lintasan P yang demikian, jika ada akan digunakan teknik pelabelan titik, yang pada prinsipnya melabel titik-titik N dengan teknik tertentu dimulai titik s , kemudian dilanjutkan melabel titik yang lain. Jika dengan teknik tersebut dapat dilabel titik t , maka dengan teknik “prosedur balik” lintasan P ditemukan, tetapi jika titik t tidak dapat dilabel, maka tidak ada lintasan P seperti itu pada N (Budayasa, 2007:240).

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 $f(i, j) = 0, \forall (i, j) \in \Gamma$) kemudian dilanjutkan ke Rutin Pelabelan.

Langkah 2 : Rutin Pelabelan.

- a. Label $v_s = (s, +, \varepsilon(s) = \sim)$. Titik v_s telah terlabel dan belum “teramati”. (Semua titik v dikatakan telah teramati jika semua titik yang dapat dilabel dari titik v sudah terlabel).
- b. Dipilih 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 $\varepsilon(y) = \min \{\varepsilon(x), f(y, x)\}$. 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 $\varepsilon(y) = \min \{\varepsilon(x), c(x, y) - f(x, y)\}$. Titik v_y terlabel, tetapi belum teramati. Diubah label v_x dengan cara melingkari tanda + atau -. Selanjutnya titik v_x terlabel dan teramati.
- c. Diulangi 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”, dicari lintasan peningkatan P dengan $i(P)$ adalah label v_t .

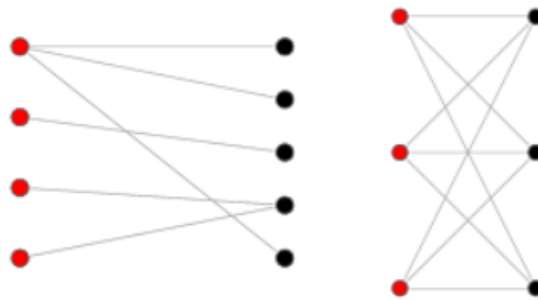
Langkah 4 : Ditingkatkan nilai $flow$ f sebesar label v_t , berdasarkan lintasan peningkatan P dengan menggunakan “Rutin Peningkatan”. Rutin Peningkatan yaitu:

- a. Misalkan $Z = t$ dilanjutkan ke langkah b.
- b. Jika label $v_z = (q, +, \varepsilon(t))$ ditingkatkan nilai $f(q, z)$ dengan $\varepsilon(t) = i(P)$.
- c. Jika $q = s$, semua label dihapus, kemudian diperoleh $flow$ f baru dengan nilai $= i(P) +$ nilai $flow$ f lama. Selanjutnya diganti $flow$ f dengan $flow$ f yang baru, dan kembali ke langkah 1.

(Budayasa, 2007:240).

2.4 Graf Bipartisi

Graf bipartisi atau sering disebut *bigraf* adalah graf yang titik-titiknya dapat dikelompokkan menjadi dua himpunan *disjoint* sehingga titik-titik pada graf tersebut yang terletak pada satu himpunan tidak ada yang *adjacent*, sebagaimana dicontohkan pada gambar berikut.

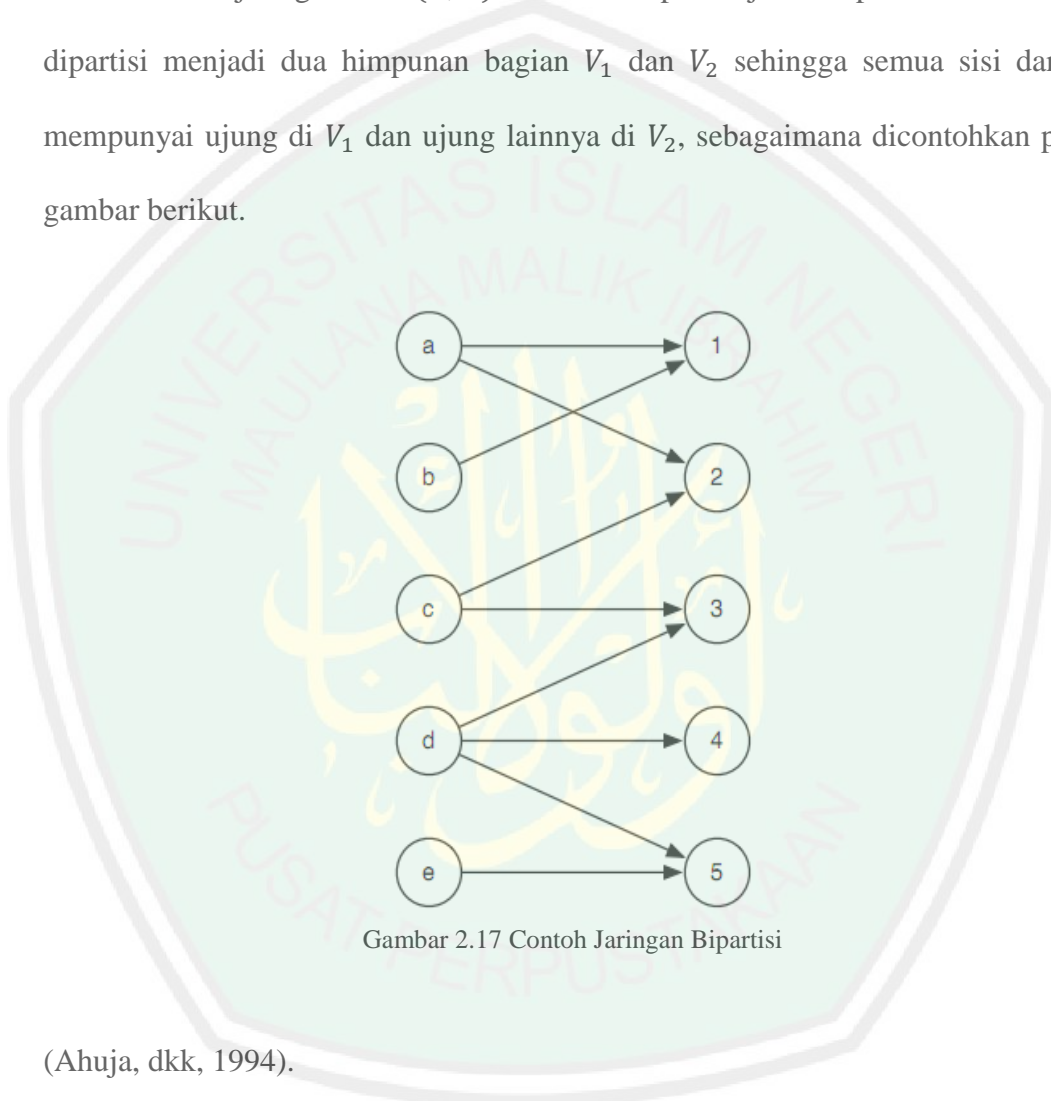


Gambar 2.16 Contoh Graf Bipartisi

(Wolfram, 2015).

2.5 Jaringan Bipartisi

Sebuah jaringan $G = (V, E)$ dikatakan bipartisi jika himpunan titik V dapat dipartisi menjadi dua himpunan bagian V_1 dan V_2 sehingga semua sisi dari G mempunyai ujung di V_1 dan ujung lainnya di V_2 , sebagaimana dicontohkan pada gambar berikut.



(Ahuja, dkk, 1994).

2.5.1 Konsep *Flow* pada Jaringan Bipartisi

Diberikan suatu himpunan *server/base-station* \mathcal{B} dan suatu himpunan *clients* \mathcal{C} dengan $|\mathcal{B}| = n$ dan $|\mathcal{C}| = m$. Setiap $B \in \mathcal{B}$ mempunyai *budget* (anggaran) $\beta(B)$ yang merupakan total kapasitas yang dikirim oleh *server* ke *clients*, sedangkan setiap *client* $C \in \mathcal{C}$ mempunyai *demand* $\alpha(C)$ yaitu total kapasitas yang dapat diterima dari semua *server*.

Setiap pasangan *server* dan *client* (B, C) terdapat kapasitas kritis $\eta(B, C)$, yang berkorespondensi dengan maksimum lebar saluran yang menghubungkan B ke C . Setiap $B \in \mathcal{B}$, diberikan nilai awal $\tau(B)$ yang menentukan kapasitas saluran (B, C) (dinotasikan $\psi_\tau(B, C)$) sebagai berikut.

$$\psi_{\tau(B,C)} := \begin{cases} \tau(B), & \tau(B) \leq \eta(B) \\ 0, & \text{lainnya} \end{cases}$$

Kapasitas dari semua saluran yang telah ditentukan, akan dicari nilai *flow* $f(B, C)$ dari masing-masing saluran (B, C) yang memenuhi:

$$f(B, C) \leq \psi_\tau(B, C)$$

$$\sum_{C \in \mathcal{C}} f(B, C) \leq \beta(B)$$

$$\sum_{B \in \mathcal{B}} f(B, C) \leq \alpha(C)$$

(Azar, dkk, 2013).

2.6 Kajian Agama Tentang Masalah dalam Kehidupan serta Kaidah Penyelesaiannya

Salah satu hukum dan ketentuan Allah Swt. bagi semua makhluk-Nya adalah ditetapkannya ujian dan cobaan. Hal ini merupakan sebuah tabiat kehidupan bahwa manusia tidak akan pernah lepas dari musibah yang akan menghampirinya. Banyak musibah yang memusnahkan pekerjaan atau harapan. Banyak derita yang ditimbulkan karena kehilangan harta, atau lainnya yang membuat hidup menjadi sesak (Jazuli, 2006:101).

Namun, bagaimanapun bentuk ujian dan cobaan yang terjadi, tentunya memiliki suatu penyelesaiannya. Sebagaimana ditegaskan di dalam al-Quran surah al-Insyirah/94:5-6:

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٥﴾ إِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٦﴾

“Karena sesungguhnya sesudah kesulitan itu pasti ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan” (QS. al-Insyirah/94:5-6).

Menurut ustadz Muhammad Abduh dalam tafsirnya menyebutkan bahwa jenis kesulitan apapun pasti dapat diatasi, selama orang yang menghadapi kesulitan tersebut memiliki jiwa yang kuat untuk mencari solusinya, menggunakan akal pikiran semaksimal mungkin, berdoa, dan tawakal kepada Allah Swt. (Amirullah, 2004:279).

Menurut Muthahhari (2000), makna global dari surah al-Insyirah/94:5-6 yaitu bahwa kesulitan itu datang sedangkan kemudahan bersamanya, atau kemudahan itu ada di dalam kesulitan. Maksud firman Allah Swt. tersebut yaitu jika seseorang ingin memperoleh kemudahan dan kebahagiaan, maka orang tersebut tidak mungkin memperolehnya begitu saja sebelum menempuh jalan yang penuh tantangan.

Dalam Tafsir Nurul Quran dikatakan, wahai Nabi Allah Swt., janganlah bersedih karena banyaknya kesulitan dan rintangan, karena semua rintangan dan kesulitan akan berubah menjadi kemudahan dan kelapangan. Kedua ayat di atas membesarkan hati kaum mukmin yang ikhlas untuk mengenal dan meyakini bahwa kesulitan dan kesukaran apapun yang dihadapi di jalan Allah Swt., maka Allah Swt. senantiasa memberi solusi/jalan keluar. Allah Swt. pasti akan memberikan kunci pembebasan pada suatu jalan yang mengantarkan pada kemudahan dan kebahagiaan. Selanjutnya dapat diartikan bahwa solusi atau penyingkapan masalah tidak semata-mata setelah “kesulitan”, tapi kemudahan memang disertakan dengan kesulitan, atau dalam setiap kesulitan yang dihadapi selalu disertakan kemudahan di dalamnya (Imani, 2006:159).

Berdasarkan beberapa pendapat ahli tafsir di atas, suatu kemudahan yang datang setelah kesulitan harus diyakini adanya. Selanjutnya, kemudahan yang dijanjikan oleh Allah Swt. dalam ayat di atas diperkuat dengan Firman Allah Swt. di dalam al-Quran surah al-Baqarah/3:286 yaitu:

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا

“Allah Swt. tidak membebani seseorang melainkan sesuai dengan kesanggupannya” (QS. al-Baqarah/3:286).

Potongan ayat surah al-Baqarah tersebut menyatakan bahwa Allah Swt. tidak membebani seseorang di luar batas kemampuan seseorang tersebut. Artinya setiap manusia diberikan ukuran/kapasitas sendiri-sendiri dalam menyelesaikan tiap masalah yang dihadapinya. Sehingga sejatinya setiap orang pasti dapat menyelesaikan masalahnya. Namun, banyak manusia yang masih bingung dalam mencari solusi atau jalan keluar dari masalah yang dialaminya, padahal telah tercantum di dalam Kitabullah sebuah kaidah untuk menyelesaikan suatu problematika/masalah sebagaimana tertera dalam firman-Nya, yaitu:

وَمَنْ يَتَّقِ اللَّهَ يَجْعَلْ لَهُ مَخْرَجًا ۖ وَيَرْزُقْهُ مِنْ حَيْثُ لَا يَحْتَسِبُ

“Barangsiapa bertakwa kepada Allah Swt. niscaya Dia akan mengadakan baginya jalan keluar. Dan memberinya rezeki dari arah yang tiada disangka-sangkanya” (QS. at-Thalaq/65:2-3).

Potongan ayat surah at-Thalaq tersebut mengandung makna bahwasanya kaidah untuk mendapatkan solusi masalah dan sempitnya rezeki yaitu dengan bertakwa kepada Allah Swt. dan tentunya dengan sebenar-benar takwa. Sebagaimana ayat di atas, kaidah ini terdiri dari tiga tahapan, yaitu:

1. Takwa
2. Solusi
3. Rezeki dari arah tak disangka

Maksud dari tiga tahapan di atas yaitu barang siapa yang benar-benar bertakwa kepada Allah Swt., maka dengan sebenar-benar takwanya itu Allah Swt. akan memberikannya jalan keluar atau solusi dari setiap permasalahan yang dihadapi manusia tersebut, dan dengan takwanya pula, Allah Swt. akan menganugerahkan rezeki dari arah yang tidak disangka-sangka oleh manusia tersebut. Demikian itulah janji Allah Swt. terhadap orang yang benar-benar bertakwa kepada-Nya.

Dalam Minhajul abidin, Al Ghazali membagi definisi takwa menjadi tiga yaitu takwa yang berarti takut, takwa bermakna taat dan takwa yang berarti membersihkan hati dari segala dosa. Makna takwa yang ketiga inilah yang sejalan dengan sabda Nabi Saw. bahwasanya takwa itu letaknya di hati. Indikasi Nabi Saw. tersebut menunjukkan bahwa orang yang bertakwa adalah orang yang mampu membersihkan hatinya dari noktah-noktah dosa. Pengosongan hati dari sifat tercela seperti *Itbaa'ulhawa* (mengikuti hawa nafsu), *Ujub* (membanggakan diri), *Riyaa* (pamer dalam ibadah), *sum'ah* (mendengarkan amalannya), *takabbur* (sombong), *Thoma'* (rakus), *hasud* (dengki), *hiqdu* (dendam) dan *hubbuddunya* (cinta dunia berlebihan) kemudian menghiasinya dengan sifat-sifat terpuji misalnya syukur, ridho, sabar, *qonaah* (merasa cukup dengan pemberian Allah Swt.), zuhud, tawakal, dan ikhlas merupakan manifestasi dari ketakwaan yang sebenarnya yang nantinya akan terpancar keluar lewat sikap dan perilaku lahiriyahnya.

Ulama menyatakan bahwasanya terdapat lima langkah yang dapat dilakukan seorang mukmin untuk mencapai derajat takwa, diantaranya:

1. *Mu'ahadah* (Mengingat Perjanjian)

Makna *mu'ahadah* yaitu mengingat-ingat kembali janji yang selalu dikatakan setiap mukmin ketika shalat. Semakin sering seorang mukmin mengingat janji dan sumpah yang diucapkannya, maka mukmin tersebut akan senantiasa menapaki kehidupan ini dengan nilai-nilai ketakwaan.

2. *Muraqabah* (Merasakan Kehadiran Allah Swt.)

Makna *muraqabah* yaitu merasakan kehadiran Allah di setiap waktu dan keadaan, serta merasakan kebersamaan-Nya di waktu sepi maupun ramai.

3. *Muhasabah* (Introspeksi Diri)

Makna *muhasabah* yaitu evaluasi diri dan meningkatkan kualitas diri dengan selalu mengambil hikmah dari setiap sesuatu yang terjadi dalam diri.

4. *Mu'aqabah* (Pemberian Sanksi)

Makna *muhasabah* yaitu mencoba memberi sanksi kepada diri sendiri ketika melakukan sebuah kekhilafan, dengan demikian maka seorang mukmin akan senantiasa meningkatkan amal ibadahnya.

5. *Mujahadah* (Bersungguh-sungguh)

Makna *mujahadah* yaitu bersungguh-sungguh kepada Allah Awt., yaitu apabila seorang mukmin terseret dalam kemalasan, santai, cinta dunia, dan tidak lagi melaksanakan amal-amal sunah serta ketaatan yang lainnya tepat pada waktunya maka orang tersebut harus memaksa dirinya melakukan amal-amal sunah lebih banyak dari sebelumnya.

BAB III
PEMBAHASAN

3.1 Paparan Data dan Modifikasi Algoritma Ford-Fulkerson

Data pendistribusian barang yang disediakan merupakan data simulasi yang memiliki lima titik sumber dan lima titik tujuan, sebagaimana ditunjukkan pada tabel berikut.

Tabel 3.1 Data Pendistribusian Barang

Titik Sumber	Titik Tujuan	Kapasitas
s_1	t_1	0
	t_2	6
	t_3	0
	t_4	0
	t_5	0
s_2	t_1	14
	t_2	2
	t_3	7
	t_4	0
	t_5	0
s_3	t_1	0
	t_2	0
	t_3	0
	t_4	0
	t_5	6
s_4	t_1	0
	t_2	0
	t_3	6
	t_4	6
	t_5	0
s_5	t_1	0
	t_2	0
	t_3	0
	t_4	0
	t_5	3

Pencarian *flow* maksimum pada data di atas, dilakukan dengan memodifikasi Algoritma Ford-Fulkerson agar algoritma ini dapat digunakan untuk menyelesaikan masalah distribusi barang ini.

Berdasarkan langkah-langkah pada Algoritma Ford-Fulkerson yang telah diuraikan pada bab sebelumnya, maka akan dijelaskan modifikasi dari Algoritma Ford-Fulkerson. Secara garis besar, Algoritma Ford-Fulkerson pada suatu jaringan dengan beberapa titik sumber dan beberapa titik tujuan adalah sebagai berikut.

Misalkan jaringan A merupakan suatu jaringan yang memiliki beberapa titik sumber dan beberapa titik tujuan, langkah-langkah yang dilakukan yaitu:

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 ∞ , dan nilai *flow* dapat dimulai dari 0.

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

Langkah 3 : Dilakukan rutin pelabelan.

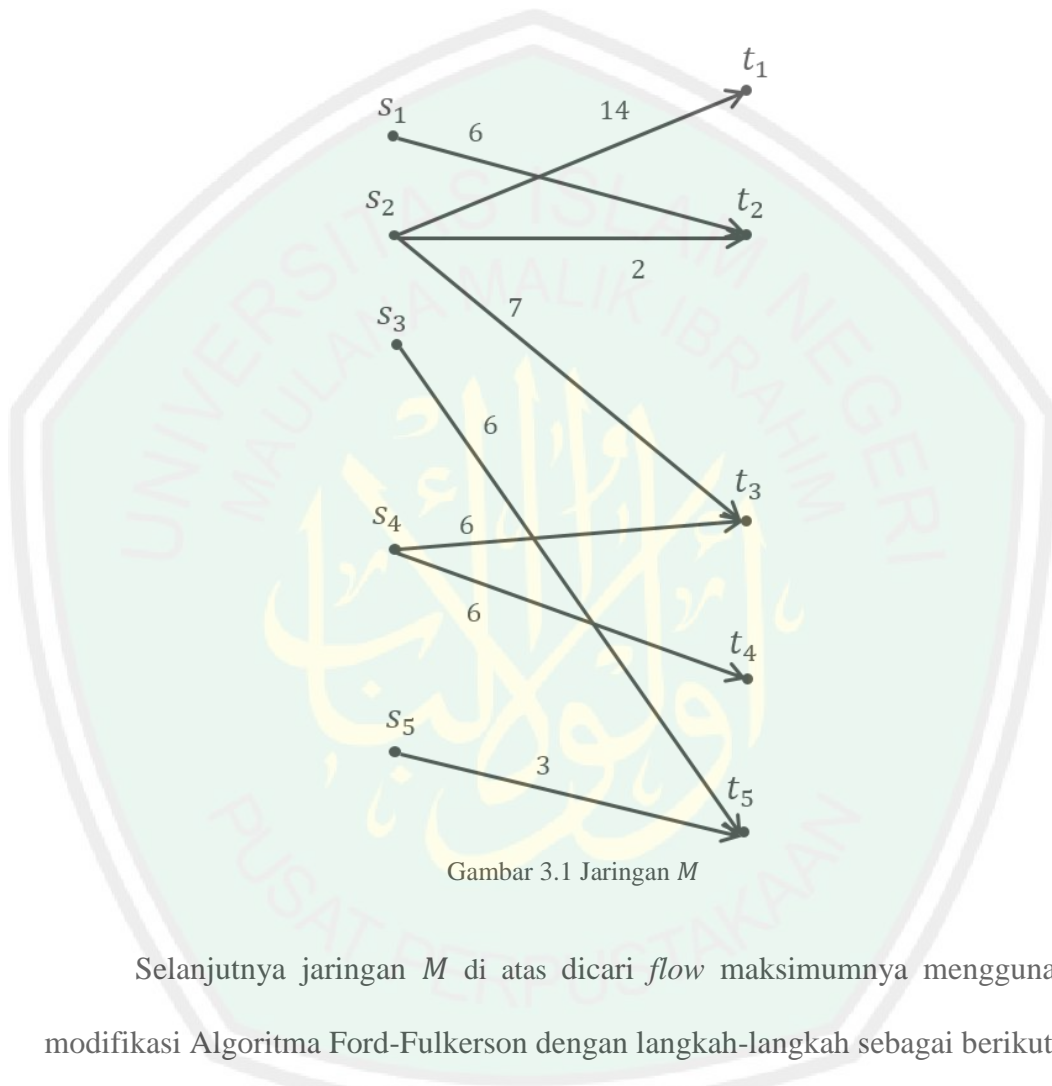
Langkah 4 : Digunakan prosedur balik.

Langkah 5 : Dilakukan rutin peningkatan.

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

3.2 *Flow* Maksimum pada Data Distribusi Barang

Data pada Tabel 3.1 dibentuk menjadi sebuah jaringan yang dimisalkan jaringan M , sebagaimana dapat dilihat pada gambar berikut.

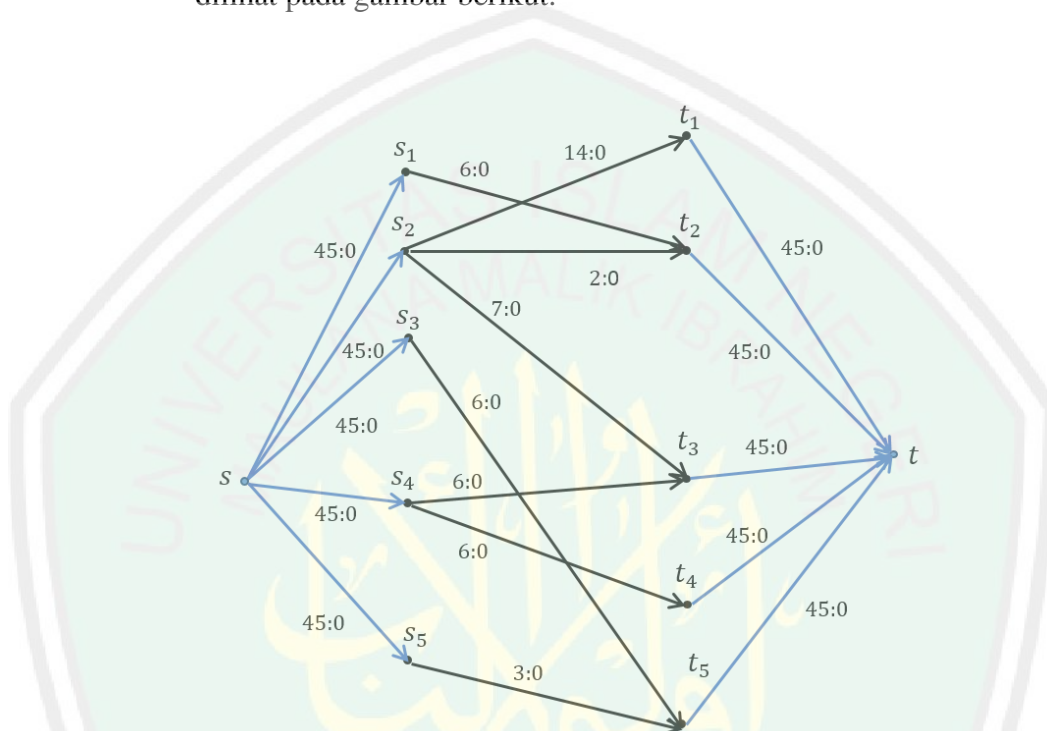


Gambar 3.1 Jaringan M

Selanjutnya jaringan M di atas 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 (s) dan satu titik tujuan baru (t) sedemikian hingga titik s merupakan titik sumber utama dan titik t merupakan titik tujuan utama di M^* dimana titik s dan titik t pada jaringan M^* diberi warna biru.

Langkah 2: Dibuat kapasitas busur $(s, s_1), (s, s_2), (s, s_3), (s, s_4), (s, s_5)$ dan $(t_1, t), (t_2, t), (t_3, t), (t_4, t), (t_5, t)$ yang juga diberi warna biru dengan nilai kapasitas utama yaitu 45 dan nilai *flow* awal 0, sebagaimana dapat dilihat pada gambar berikut.



Gambar 3.2 Jaringan M^* dengan Nilai *Flow* Awal $f = 0$

Jaringan M^* pada Gambar 3.2 mempunyai kapasitas di titik s sebesar 45, sedangkan total kapasitas di beberapa titik sumber (s_i) dengan nilai 50. Sebagaimana prosedur Algoritma Ford-Fulkerson pada bab sebelumnya, agar nilai *flow* segera terpenuhi maka strategi *flow* maksimum yang dilakukan yaitu dicari lintasan peningkatan dari titik s kemudian dipilih titik sumber s_i yang busur terhubungnya paling banyak serta titik sumber s_i yang mempunyai kapasitas terbesar, sehingga urutan iterasi untuk lintasan peningkatan yang dipilih yaitu titik s_2, s_4, s_1, s_3, s_5 . Untuk titik yang terlabel dan teramati, diberi tanda

⊕. Langkah selanjutnya, pencarian *flow* maksimum dari titik s ke titik t menggunakan modifikasi Algoritma Ford-Fulkerson untuk iterasi pertama yaitu:

Langkah 3 : Rutin Pelabelan.

- 1) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 2) Dari titik s_2 , diberi label untuk titik t_1, t_2, t_3 masing-masing $t_1 = (s_2, +, 14)$, $t_2 = (s_2, +, 2)$, $t_3 = (s_2, +, 7)$. Titik s_2 telah terlabel dan teramati dengan label $s_2 = (s, \oplus, 45)$.
- 3) Dari titik t_1 , diberi label untuk titik t dengan label $t = (t_1, +, 45)$. Titik t_1 telah terlabel dan teramati dengan label $t_1 = (s_2, \oplus, 14)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_1 , titik t_1 dilabel dari s_2 dan titik s_2 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_2, t_1, t)$ dan $i(P) = \min(45, 14, 45) = 14$.

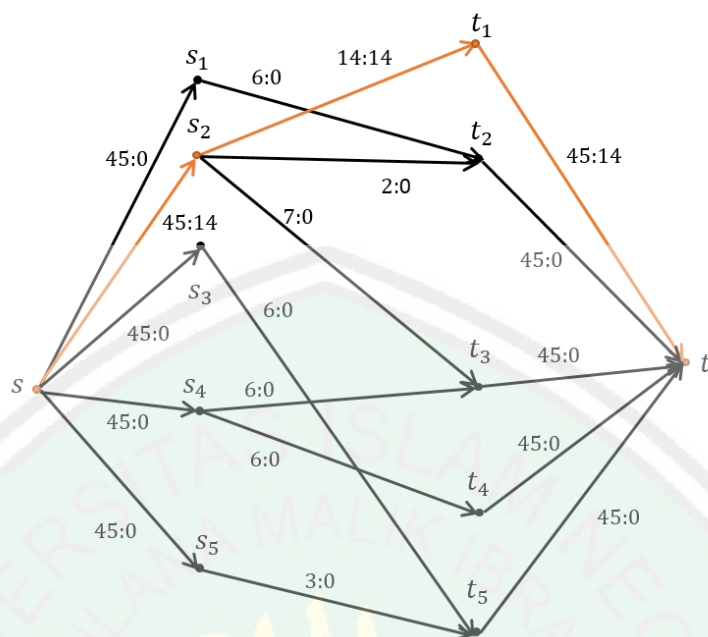
Langkah 5: Rutin peningkatan.

Titik $t = (t_1, +, 45)$ maka nilai *flow* pada busur (t_1, t) ditambah 14.

Titik $t_1 = (s_2, +, 14)$ maka nilai *flow* pada busur (s_2, t_1) ditambah 14.

Titik $s_2 = (s, +, 45)$ maka nilai *flow* pada busur (s, s_2) 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 gambar berikut.



Gambar 3.3 Jaringan M^* dengan Nilai $Flow$ Baru $f_1 = 14$

Iterasi selanjutnya diulangi algoritma $flow$ maksimum untuk mengkonstruksi $flow$ maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: $Flow$ f_1 dengan nilai 14.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45), s_2 = (s, +, 45), s_3 = (s, +, 45), s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Sekarang titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_2 , diberi label untuk titik t_1, t_2, t_3 masing-masing $t_1 = (s_2, +, 14), t_2 = (s_2, +, 2), t_3 = (s_2, +, 7)$. Titik s_2 telah terlabel dan teramati dengan label $s_2 = (s, \oplus, 31)$.

4) Dari titik t_2 , diberi label untuk titik t dengan label $t = (t_2, +, 31)$.

Titik t_2 telah terlabel dan teramati dengan label $t_2 = (s_2, \oplus, 2)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_2 , titik t_2 dilabel dari s_2 dan titik s_2 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_2, t_2, t)$ dan $i(P) = \min(31, 2, 31) = 2$.

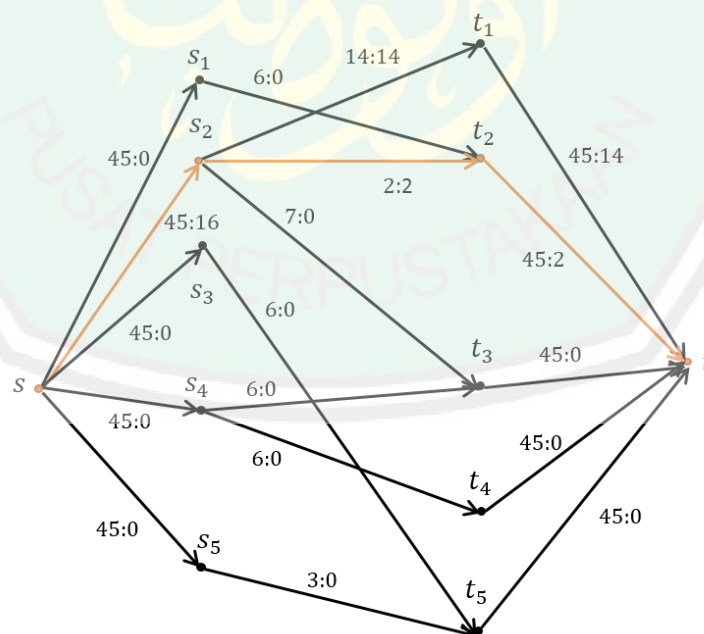
Langkah 5: Rutin peningkatan.

Titik $t = (t_2, +, 31)$ maka nilai *flow* pada busur (t_2, t) ditambah 2.

Titik $t_2 = (s_2, +, 2)$ maka nilai *flow* pada busur (s_2, t_2) ditambah 2.

Titik $s_2 = (s, +, 31)$ maka nilai *flow* pada busur (s, s_2) ditambah 2.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_2 = f_1 + 2 = 14 + 2 = 16$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.4 Jaringan M^* dengan Nilai *Flow* Baru $f_2 = 16$

Diulangi algoritma *flow* maksimum untuk mengkonstruksi *flow* maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: *Flow* f_2 dengan nilai 16.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_2 , diberi label untuk titik t_1, t_2, t_3 masing-masing $t_1 = (s_2, +, 14)$, $t_2 = (s_2, +, 2)$, $t_3 = (s_2, +, 7)$. Titik s_2 telah terlabel dan teramati dengan label $s_2 = (s, \oplus, 29)$.
- 4) Dari titik t_3 , diberi label untuk titik t dengan label $t = (t_3, +, 29)$. Titik t_4 telah terlabel dan teramati dengan label $t_3 = (s_2, \oplus, 7)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_3 , titik t_4 dilabel dari s_2 dan titik s_2 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_2, t_3, t)$ dan $i(P) = \min(29, 7, 29) = 7$.

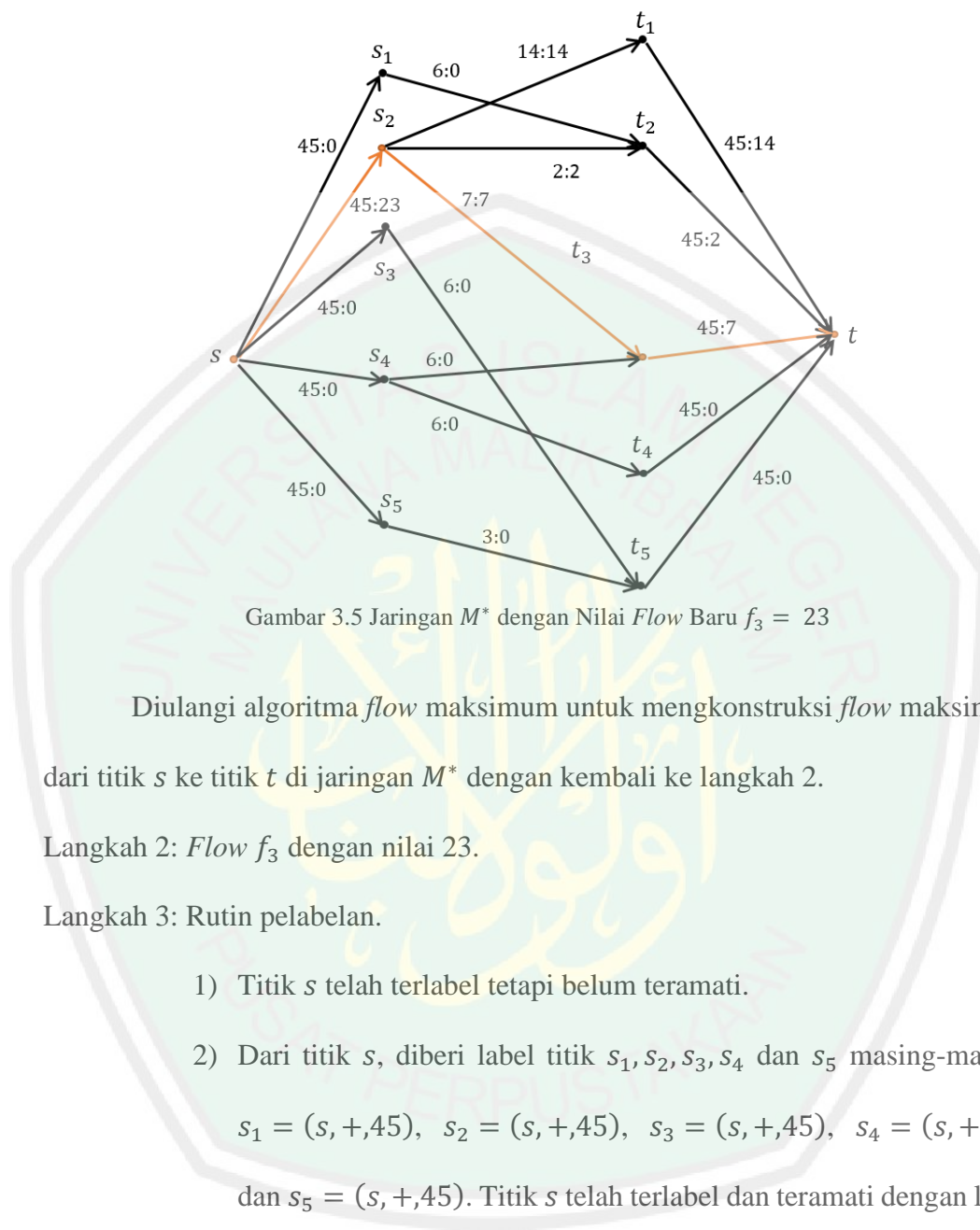
Langkah 5: Rutin peningkatan.

Titik $t = (t_3, +, 29)$ maka nilai *flow* pada busur (t_3, t) ditambah 7.

Titik $t_3 = (s_2, +, 7)$ maka nilai *flow* pada busur (s_2, t_3) ditambah 7.

Titik $s_2 = (s, +, 29)$ maka nilai *flow* pada busur (s, s_2) ditambah 7.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_3 = f_2 + 7 = 16 + 7 = 23$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.5 Jaringan M^* dengan Nilai $Flow$ Baru $f_3 = 23$

Diulangi algoritma $flow$ maksimum untuk mengkonstruksi $flow$ maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: $Flow$ f_3 dengan nilai 23.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_4 , diberi label untuk titik t_3 dan t_4 masing-masing $t_3 = (s_5, +, 6)$ dan $t_4 = (s_5, +, 6)$. Titik s_4 telah terlabel dan teramati dengan label $s_4 = (s, \oplus, 22)$.

4) Dari titik t_3 , diberi label untuk titik t dengan label $t = (t_3, +, 22)$.

Titik t_3 telah terlabel dan teramati dengan label $t_3 = s_4, \oplus, 6)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_3 , titik t_3 dilabel dari s_4 dan titik s_4 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_4, t_3, t)$ dan $i(P) = \min(22, 6, 22) = 6$.

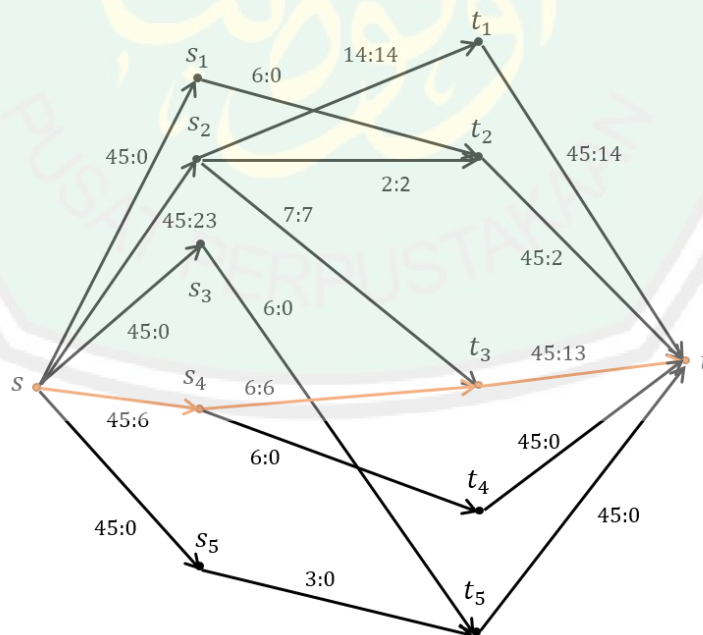
Langkah 5: Rutin peningkatan.

Titik $t = (t_3, +, 22)$ maka nilai *flow* pada busur (t_3, t) ditambah 6.

Titik $t_3 = (s_4, +, 6)$ maka nilai *flow* pada busur (s_4, t_3) ditambah 6.

Titik $s_4 = (s, +, 22)$ maka nilai *flow* pada busur (s, s_4) ditambah 6.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_4 = f_3 + 6 = 23 + 6 = 29$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.6 Jaringan M^* dengan Nilai *Flow* Baru $f_4 = 29$

Diulangi algoritma *flow* maksimum untuk mengkonstruksi *flow* maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: *Flow* f_4 dengan nilai 29.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_4 , diberi label untuk titik t_3 dan t_4 masing-masing $t_3 = (s_5, +, 6)$ dan $t_4 = (s_5, +, 6)$. Titik s_4 telah terlabel dan teramati dengan label $s_4 = (s, \oplus, 16)$.
- 4) Dari titik t_4 , diberi label untuk titik t dengan label $t = (t_4, +, 6)$. Titik t_4 telah terlabel dan teramati dengan label $t_4 = (s_4, \oplus, 16)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_4 , titik t_4 dilabel dari s_4 dan titik s_4 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_4, t_4, t)$ dan $i(P) = \min(16, 6, 16) = 6$.

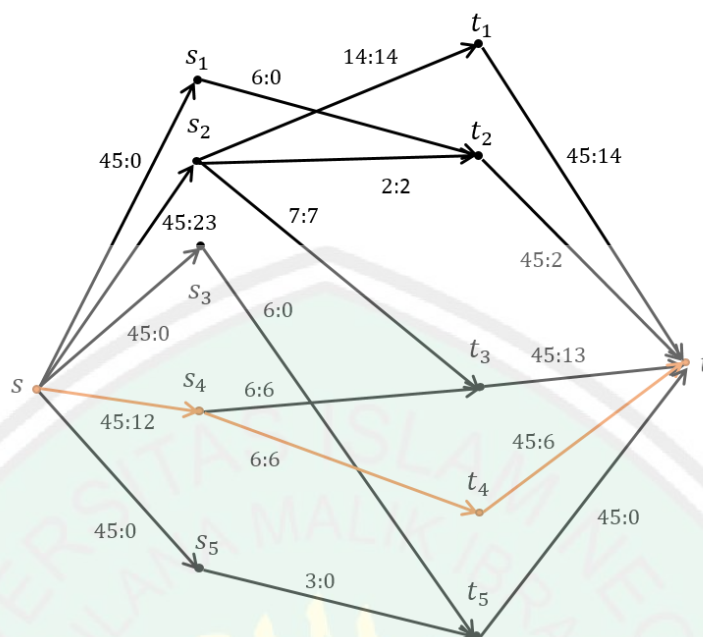
Langkah 5: Rutin peningkatan.

Titik $t = (t_4, +, 16)$ maka nilai *flow* pada busur (t_4, t) ditambah 6.

Titik $t_4 = (s_4, +, 6)$ maka nilai *flow* pada busur (s_4, t_4) ditambah 6.

Titik $s_4 = (s, +, 16)$ maka nilai *flow* pada busur (s, s_4) ditambah 6.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_5 = f_4 + 6 = 29 + 6 = 35$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.7 Jaringan M^* dengan Nilai *Flow* Baru $f_5 = 35$

Diulangi algoritma *flow* maksimum untuk mengkonstruksi *flow* maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: *Flow* f_5 dengan nilai 35.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_1 , diberi label untuk titik t_2 yaitu $t_2 = (s_1, +, 6)$. Titik s_1 telah terlabel dan teramati dengan label $s_1 = (s, \oplus, 10)$.
- 4) Dari titik t_2 , diberi label untuk titik t dengan label $t = (t_2, +, 10)$. Titik t_2 telah terlabel dan teramati dengan label $t_2 = (s_1, \oplus, 6)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_2 , titik t_2 dilabel dari titik s_1 dan titik s_1 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_1, t_2, t)$ dan $i(P) = \min(10, 6, 10) = 6$.

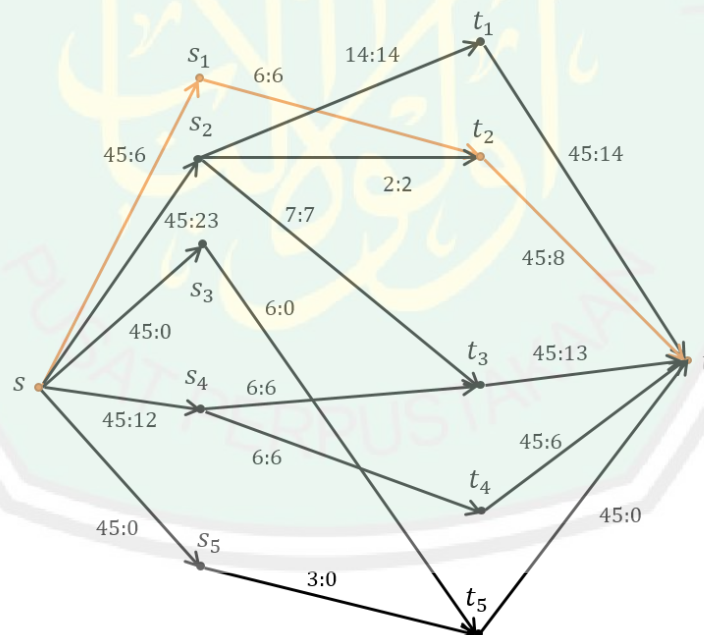
Langkah 5: Rutin peningkatan.

Titik $t = (t_2, +, 10)$ maka nilai *flow* pada busur (t_2, t) ditambah 6.

Titik $t_2 = (s_1, +, 6)$ maka nilai *flow* pada busur (s_1, t_2) ditambah 6.

Titik $s_1 = (s, +, 10)$ maka nilai *flow* pada busur (s, s_1) ditambah 6.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_6 = f_5 + 6 = 35 + 6 = 41$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.8 Jaringan M^* dengan Nilai *Flow* Baru $f_6 = 41$

Diulangi algoritma *flow* maksimum untuk mengkonstruksi *flow* maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: *Flow* f_6 dengan nilai 41.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_3 , diberi label untuk titik t_5 yaitu $t_5 = (s_3, +, 6)$. Titik s_3 telah terlabel dan teramati dengan label $s_3 = (s, \oplus, 4)$.
- 4) Dari titik t_5 , diberi label untuk titik t dengan label $t = (t_5, +, 4)$. Titik t_5 telah terlabel dan teramati dengan label $t_5 = (s_3, \oplus, 6)$.

Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_5 , titik t_5 dilabel dari s_3 dan titik s_3 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_3, t_5, t)$ dan $i(P) = \min(4, 6, 4) = 4$.

Langkah 5: Rutin peningkatan.

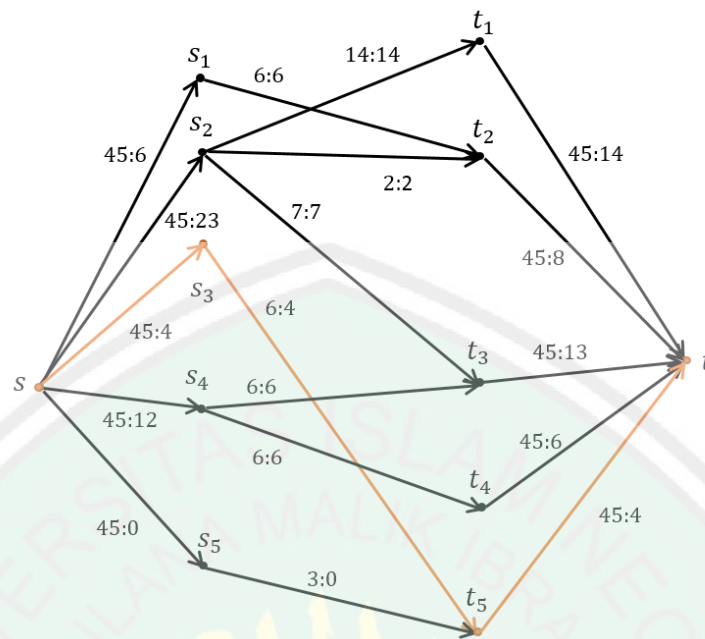
Titik $t = (t_5, +, 4)$ maka nilai *flow* pada busur (t_5, t) ditambah 4.

Titik $t_5 = (s_3, +, 6)$ maka nilai *flow* pada busur (s_3, t_5) ditambah 4.

Titik $s_3 = (s, +, 4)$ maka nilai *flow* pada busur (s, s_3) ditambah 4.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru

$f_7 = f_6 + 4 = 41 + 4 = 45$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.9 Jaringan M^* dengan Nilai *Flow* Baru $f_7 = 45$

Diulangi algoritma *flow* maksimum untuk mengkonstruksi *flow* maksimum dari titik s ke titik t di jaringan M^* dengan kembali ke langkah 2.

Langkah 2: *Flow* f_7 dengan nilai 45.

Langkah 3: Rutin pelabelan.

- 1) Titik s telah terlabel tetapi belum teramati.
- 2) Dari titik s , diberi label titik s_1, s_2, s_3, s_4 dan s_5 masing-masing $s_1 = (s, +, 45)$, $s_2 = (s, +, 45)$, $s_3 = (s, +, 45)$, $s_4 = (s, +, 45)$ dan $s_5 = (s, +, 45)$. Titik s telah terlabel dan teramati dengan label $s = (s, \oplus, 45)$.
- 3) Dari titik s_5 , diberi label untuk titik t_5 yaitu $t_5 = (s_5, +, 3)$. Titik s_5 telah terlabel dan teramati dengan label $s_5 = (s, \oplus, 0)$.
- 4) Dari titik t_5 , diberi label untuk titik t dengan label $t = (t_5, +, 0)$. Titik t_5 telah terlabel dan teramati dengan label $t_5 = (s_5, \oplus, 3)$.

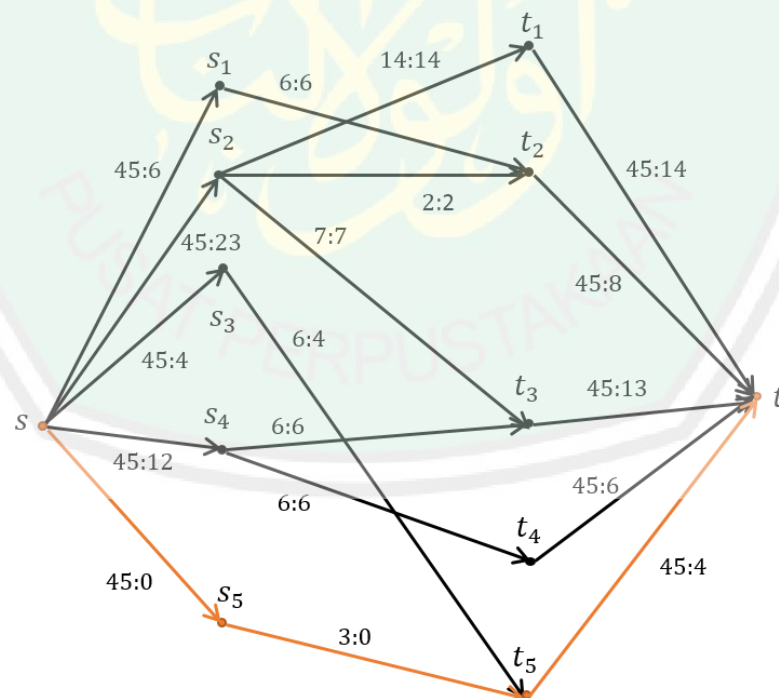
Langkah 4: Gunakan prosedur balik.

Titik t dilabel dari titik t_5 , titik t_5 dilabel dari s_5 dan titik s_5 dilabel dari titik s . Dengan demikian diperoleh lintasan peningkatan $P = (s, s_5, t_5, t)$ dan $i(P) = \min(0, 3, 0) = 0$.

Langkah 5: Rutin peningkatan.

$t = (t_5, +, 0)$; $t_5 = (s_5, +, 3)$ dan $s_5 = (s, +, 0)$. Karena pada lintasan $P = (s, s_5, t_5, t)$ diperoleh $i(P) = 0$ maka nilai *flow* pada busur (s, s_5) , (s_5, t_5) dan (t_5, t) ditambah 0 (artinya tidak bertambah). Sedangkan nilai *flow* pada busur lainnya adalah tetap.

Setelah dilakukan penghapusan terhadap semua label, diperoleh nilai *flow* baru $f_8 = f_7 + 0 = 45 + 0 = 45$ dan ditunjukkan sebagaimana gambar berikut.



Gambar 3.10 Jaringan M^* dengan Nilai *Flow* Baru $f_8 = 45$

Semua busur di jaringan M^* telah terlewati, dan semua titik di M^* yang terlabel telah teramati dan titik t tidak terlabel, sehingga iterasi dihentikan dan didapatkan *flow* f_8 adalah *flow* maksimum di jaringan M^* dengan nilai 45.

Interpretasi jaringan M^* pada Gambar 3.10 yaitu diambil suatu simulasi data pendistribusian barang dimana stok barang sebanyak 45 biji dan dibutuhkan untuk didistribusikan ke lima toko dengan total semua barang sebanyak 50 biji. Agar stok barang di perusahaan cepat habis maka digunakan strategi *flow* maksimum yaitu mendahulukan agen yang mengirim barang di toko terbanyak serta mendahulukan pengiriman barang dengan permintaan toko yang paling banyak. Selanjutnya dilakukan iterasi dengan lintasan peningkatan sampai semua agen telah menyalurkan stok barang di toko yaitu sampai iterasi kedelapan. Pada lintasan peningkatan di iterasi ketujuh, permintaan toko melebihi stok barang sehingga banyaknya barang yang disalurkan yaitu sisa dari stok barang di perusahaan. Untuk lintasan peningkatan di iterasi ketujuh stok barang di perusahaan telah habis sehingga tidak ada barang yang disalurkan ke toko terakhir.

Berdasarkan Gambar 3.1, jaringan M merupakan bentuk jaringan bipartisi dimana himpunan titik-titik di jaringan M dapat dipartisi menjadi dua himpunan bagian, sehingga dalam kasus ini 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 tersebut mempunyai lima titik tujuan sehingga digunakan modifikasi Algoritma Ford-Fulkerson yaitu dengan menambahkan satu titik tujuan baru dengan kapasitasnya adalah stok persediaan barang yaitu 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 yaitu jumlah dari *flow* maksimum untuk semua titik sumber di jaringan.

Selanjutnya jaringan M pada Gambar 3.1 dipartisi menjadi lima karena memiliki lima titik sumber. Untuk iterasi pertama dicari lintasan peningkatan yang melalui titik s_1, t_2, t dan didapat inkremen dari lintasan tersebut adalah 6, sehingga *flow* maksimum untuk s_1 yaitu $f_1 = 6$. Selanjutnya karena s_2 terhubung ke t_1, t_2, t_3 maka untuk iterasi kedua dicari lintasan peningkatan yang melalui s_2 ke t dimana untuk lintasan pertama yang melalui titik s_2, t_1, t inkremennya 14, lintasan kedua yang melalui titik s_2, t_2, t inkremennya 2 dan lintasan ketiga yang melalui titik s_2, t_3, t inkremennya 8, sehingga *flow* maksimum untuk s_2 yaitu $f_2 = 14 + 2 + 7 = 23$. Iterasi ketiga dicari lintasan peningkatan yang melalui s_3, t_5, t dan didapat inkremen dari lintasan tersebut adalah 6, sehingga *flow* maksimum untuk s_3 yaitu $f_3 = 6$. Selanjutnya karena s_4 terhubung ke t_3, t_4 maka untuk iterasi keempat dicari lintasan peningkatan yang melalui s_4 ke t dimana untuk lintasan pertama yang melalui titik s_4, t_3, t inkremennya 6 dan lintasan kedua yang melalui titik s_4, t_4, t inkremennya 4, sehingga *flow* maksimum untuk s_4 yaitu $f_4 = 6 + 4 = 10$. Iterasi kelima dicari lintasan peningkatan yang melalui s_5, t_5, t dan didapat inkremen dari lintasan tersebut adalah 0, sehingga *flow* maksimum untuk s_5 yaitu $f_5 = 0$. Semua titik sumber telah diselidiki dan didapatkan *flow* maksimum untuk masing-masing titik sumber, sehingga nilai *flow* maksimum pada jaringan M yaitu $f = \sum_{i=1}^5 f_i = f_1 + f_2 + f_3 + f_4 + f_5 = 6 + 23 + 6 + 10 + 0 = 45$ dan hasilnya sama dengan jaringan M yang tidak dipartisi.

3.3 *Flow* Maksimum pada Jaringan dengan Matlab

Pencarian *flow* maksimum di jaringan M pada Gambar 3.1 juga dapat dilakukan dengan membuat program di Matlab R2010a menggunakan modifikasi Algoritma Ford-Fulkerson, yaitu dengan menambahkan satu titik sumber baru dan satu titik tujuan baru dengan langkah-langkah iterasi sebagaimana prosedur yang dimiliki Algoritma Ford-Fulkerson. Tujuan membuat program ini yaitu agar pencarian *flow* maksimum dapat diketahui dengan lebih mudah dan cepat. Untuk *coding* pencarian *flow* maksimum pada jaringan diperoleh dengan iterasi sebanyak delapan sebagaimana dapat dilihat di Lampiran 1, sedangkan untuk hasil yang ditampilkan di *command window* didapatkan nilai *flow* maksimum dari titik sumber utama yaitu 45 dan *flow* maksimum yang menuju ke titik tujuan utama nilainya 45 sebagaimana dapat dilihat di Lampiran 2.

Berdasarkan hasil contoh pendistribusian barang dengan tiga cara pencarian *flow* maksimum pada jaringan M , didapatkan *flow* maksimum dengan nilai $f_8 = 45$ yang artinya dengan iterasi sebanyak 8 nilainya 45, pada jaringan yang dipartisi didapatkan *flow* maksimum dengan nilai $f_5 = 45$ yang artinya dengan iterasi sebanyak 5 nilainya 45, serta dengan membuat program di Matlab R2010a dengan iterasi sebanyak 8 didapatkan nilai *flow* maksimum sebesar 45. Hal ini dimaksudkan bahwa dengan cara yang berbeda serta dengan jumlah iterasi yang berbeda didapatkan hasil yang sama yaitu nilai *flow* maksimum sebesar 45.

3.4 Kajian Keagamaan tentang Penyelesaian Masalah

Permasalahan jaringan dalam penelitian ini dilakukan dengan mencari *flow* maksimum, dimana untuk memudahkan dalam mencari *flow* maksimum maka digunakan Algoritma Ford-Fulkerson. Algoritma ini umumnya digunakan untuk mencari *flow* maksimum pada jaringan yang memiliki satu titik sumber dan satu titik tujuan, namun data simulasi yang disediakan memiliki beberapa titik sumber dan beberapa titik tujuan, sehingga untuk menyelesaikan masalah distribusi barang tersebut digunakan modifikasi dari Algoritma Ford-Fulkerson. Hal ini menunjukkan bahwa semua masalah dapat diselesaikan meskipun dengan beberapa kesulitan, sebagaimana terdapat dalam surah al-Insyirah/94:5-6 yaitu sesudah kesulitan akan terdapat kemudahan, sesungguhnya sesudah kesulitan ada kemudahan. Pengulangan dua kali pada ayat tersebut menandakan bahwasanya Allah Swt. benar-benar menjanjikan kemudahan setelah kesulitan, sama halnya dalam mengerjakan permasalahan dalam jaringan.

Menyelesaikan permasalahan matematika, misalnya pencarian *flow* maksimum pada jaringan dibutuhkan sebuah algoritma untuk mempermudah dalam mencari nilai *flow* yang maksimum. Algoritma yang digunakan yaitu Algoritma Ford-Fulkerson dimana dalam algoritma tersebut terdapat langkah-langkah penyelesaian agar didapatkan nilai *flow* yang maksimum. Untuk memperoleh hasil yang benar dalam mendapatkan nilai *flow* maksimum, maka harus melakukan langkah-langkah dengan benar sesuai prosedur yang dimiliki Algoritma Ford-Fulkerson dimana langkah-langkahnya adalah mengkonstruksi nilai *flow* sebarang di jaringan, dari *flow* tersebut kemudian dicari lintasan dari titik sumber ke titik tujuan yang inkremennya positif dan jika tidak ditemukan lintasan yang demikian

maka *flow* tersebut merupakan *flow* maksimum, namun jika ditemukan lintasan yang inkremennya positif maka lanjut ke langkah berikutnya, dan langkah terakhir yaitu mengkonstruksi *flow* baru dimana nilainya lebih besar dari *flow* sebelumnya kemudian kembali ke langkah kedua. Langkah-langkah pada Algoritma Ford-Fulkerson tersebut saling berkaitan dan berurutan, sehingga dalam perhitungannya pun harus sesuai dengan prosedur yang ada di Algoritma Ford-Fulkerson.

Sebagaimana hal di atas, untuk mencari penyelesaian masalah dalam kehidupan sehari-hari juga terdapat sebuah kaidah, yaitu mengacu pada firman Allah Swt. surah at-Thalaq/65:2-3. Potongan ayat tersebut menjelaskan bahwa dengan takwa yang sebenarnya maka seseorang akan mendapatkan solusi atau jalan keluar dari peliknya kehidupan dunia serta mendapat tambahan kebahagiaan yaitu mendapatkan rezeki dari arah yang tidak diduga oleh manusia. Allah Swt. memberikan kemudahan bagi hambanya selama ia bertakwa kepada-Nya dan tentunya dengan sebenar-benar takwa. Ulama menyatakan bahwasanya terdapat lima langkah yang dapat dilakukan dalam kehidupan ini untuk menggapai ketakwaan, langkah-langkah yang dimaksud adalah *mu'ahadah*, *muroqobah*, *muhasabah*, *mu'aqobah* dan *mujahadah*. Kelima langkah tersebut memiliki kaitan antara satu dengan lainnya yaitu dengan *mu'ahadah* seorang mukmin dapat beristiqomah di atas syari'at Allah Swt., dengan *muroqobah* maka seorang mukmin dapat merasakan keagungan Allah Swt. baik di waktu sepi maupun di waktu ramai, dengan *muhasabah* maka seorang mukmin dapat terbebas dari kejahatan hawa nafsu dan dapat memenuhi hak-hak Allah Swt. dan hak-hak sesama manusia, dengan *mu'aqobah* maka seorang mukmin dapat memisahkan diri dari penyimpangan, serta dengan *mujahadah* maka seorang mukmin dapat memperbaiki

aktivitas diri sekaligus menyingkirkan kelalaian dan kemalasan. Sehingga dengan melakukan kelima langkah-langkah di atas maka seorang mukmin dapat menyelesaikan setiap masalah yang dihadapinya.

Ilmu matematika telah memberikan banyak kemudahan dalam menyelesaikan masalah dalam kehidupan sehari-hari. Agama Islam juga memberikan solusi dari setiap permasalahan, dimana dalam hal ini al-Quran menyimpan jawaban dari penyelesaian masalah melalui ayat dan makna yang tersirat maupun tersurat di dalamnya.



BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan pembahasan pada bab sebelumnya, maka untuk menentukan *flow* maksimum pada jaringan pendistribusian barang digunakan modifikasi dari Algoritma Ford-Fulkerson, karena pada jaringan tersebut terdapat lima titik sumber dan lima titik tujuan, dimana bentuk modifikasinya 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.

Berdasarkan contoh data pendistribusian barang dengan lima titik sumber dan lima titik tujuan kemudian dibentuk sebuah jaringan yang dimisalkan jaringan M dan dicari *flow* maksimum menggunakan modifikasi Algoritma Ford-Fulkerson. Jaringan yang telah dimodifikasi dimisalkan jaringan M^* dan 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.

4.2 Saran

Bagi penelitian selanjutnya diharapkan dapat mengkaji lebih mendalam terkait jaringan distribusi barang ini, serta dapat dikembangkan untuk pencarian *flow* maksimum pada jaringan penerbangan.



DAFTAR PUSTAKA

- Abdullah. 2005. *Tafsir Ibnu Katsir*. Jakarta: Pustaka Imam Asy-Syafi'i.
- Abdussakir, Azizah, N.N, dan Nofandika, F.F. 2009. *Teori Graf*. Malang: UIN-Malang Press.
- Ahuja, R.K, Orlin, J.B, Stein, C, dan Tarjan, R.E. 1994. Improved Algorithms For Bipartite Network Flow. *SIAM J. COMPUT*, 906-933.
- Amiruddin, A. 2006. *Tafsir Al Quran Kontemporer*. Bandung: Khazanah Intelektual.
- Azar, Y. Madry, A. Moscibroda, T. Panigrahi, D. dan Srinivasan, A. 2013. Maximum Bipartite Flow in Networks with Adaptive Channel Width, (Online): 1-19, (<http://research.microsoft.com/pubs/147615/icalp2009.pdf>), diakses 25 Juni 2015.
- Budayasa, I.K. 2007. *Teori Graf dan Aplikasinya*. Surabaya: Unesa University Press.
- Charland, G. dan Lesniak, L. 1996. *Graf and Digraf*. London: Chapman & Hall/CRC.
- Farizal, T. 2013. *Pencarian Aliran Maksimum dengan Algoritma Ford-Fulkerson*. Skripsi tidak diterbitkan. Semarang: Universitas Negeri Semarang.
- Imani, A.K.F. 2006. *Tafsir Nurul Quran*. Jakarta: Al-Huda.
- Jazuli, A.S. 2006. *Kehidupan dalam Pandangan Al-Qur'an*. Jakarta: Gema Insani Press.
- Mankiw, N.G. 2006. *Principles of Economics 7th Edition*. Jakarta: Salemba Empat.
- Muthahhari, M. 2000. *Tafsir Surat-Surat Pilihan*. Bandung: Pustaka Hidayah.
- Purwanto. 1998. *Teori Graph*. Malang: IKIP MALANG.
- Wolfram. 2015. *Wolfram MathWorld*, 26 (1). (Online), (<http://mathworld.wolfram.com/BipartiteGraph.html>) diakses 25 Juni 2015.

LAMPIRAN

Lampiran 1. Coding untuk Pencarian *Flow* Maksimum di M-File

```

%Program Distribusi Kaldu
clc, clear

SS=[45 0;45 0;45 0;45 0;45 0];
ST=[6 0;14 0;2 0;7 0;6 0;6 0;6 0;3 0];
TT=[45 0;45 0;45 0;45 0;45 0];
caril=0;

% JALUR 1
cari=min([SS(2,1)-SS(2,2) ST(2,1)-ST(2,2) TT(1,1)-TT(1,2)]);
SS(2,2)=cari;
ST(2,2)=cari;
TT(1,2)=cari;
caril=caril+cari;

% JALUR 2
cari=min([SS(2,1)-SS(2,2) ST(3,1)-ST(3,2) TT(2,1)-TT(2,2)]);
SS(2,2)=SS(2,2)+cari;
ST(3,2)=ST(3,2)+cari;
TT(2,2)=TT(2,2)+cari;
caril=caril+cari;

% JALUR 3
cari=min([SS(2,1)-SS(2,2) ST(4,1)-ST(4,2) TT(3,1)-TT(3,2)]);
SS(2,2)=SS(2,2)+cari;
ST(4,2)=ST(4,2)+cari;
TT(3,2)=TT(3,2)+cari;
caril=caril+cari;

% JALUR 4
cari=min([SS(4,1)-SS(4,2) ST(6,1)-ST(6,2) TT(3,1)-TT(3,2)]);
SS(4,2)=SS(4,2)+cari;
ST(6,2)=ST(6,2)+cari;
TT(3,2)=TT(3,2)+cari;
caril=caril+cari;

% JALUR 5
cari=min([SS(4,1)-SS(4,2) ST(7,1)-ST(7,2) TT(4,1)-TT(4,2)]);
SS(4,2)=SS(4,2)+cari;
ST(7,2)=ST(7,2)+cari;
TT(4,2)=TT(4,2)+cari;
caril=caril+cari;

% JALUR 6
cari=min([SS(1,1)-SS(1,2) ST(1,1)-ST(1,2) TT(2,1)-TT(2,2)]);
SS(1,2)=SS(1,2)+cari;
ST(1,2)=ST(1,2)+cari;
TT(2,2)=TT(2,2)+cari;
caril=caril+cari;

```

```

% JALUR 7
cari=min([SS(3,1)-SS(3,2) ST(5,1)-ST(5,2) TT(5,1)-TT(5,2)]);
SS(3,2)=(45-caril)-SS(3,2);
ST(5,2)=(45-caril)-ST(5,2);
TT(5,2)=(45-caril)-TT(5,2);
caril=caril+cari;

% JALUR 8
cari=min([SS(5,1)-SS(5,2) ST(8,1)-ST(8,2) TT(5,1)-TT(5,2)]);
SS(5,2)=0;
ST(8,2)=0;
TT(5,2)=TT(5,2)+0;

SS
ST
TT

flowmaksimum1=sum(SS(:,2))
flowmaksimum2=sum(TT(:,2))

```

Lampiran 2. Hasil Pencarian *Flow* Maksimum di *Command Window*

```
SS =  
  
45 6  
45 23  
45 4  
45 12  
45 0  
  
ST =  
  
6 6  
14 14  
2 2  
7 7  
6 4  
6 6  
6 6  
3 0  
  
TT =  
  
45 14  
45 8  
45 13  
45 6  
45 4  
  
flowmaksimum1 =  
  
45  
  
flowmaksimum2 =  
  
45
```