



**PENERAPAN ALGORITMA *SUCCESSIVE SHORTEST PATH*
DAN ALGORITMA *CAPACITY SCALING*
DALAM PENYELESAIAN *MINIMUM COST FLOW PROBLEM***

SKRIPSI

Oleh

**Shelda Amy Shara Huluq
NIM 121810101058**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**



**PENERAPAN ALGORITMA *SUCCESSIVE SHORTEST PATH*
DAN ALGORITMA *CAPACITY SCALING*
DALAM PENYELESAIAN *MINIMUM COST FLOW PROBLEM***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

**Shelda Amy Shara Huluq
NIM 121810101058**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**

HALAMAN PERSEMBAHAN

Dengan menyebut nama Allah SWT dan shalawat serta salam kepada junjungan Nabi Muhammad SAW, skripsi ini saya persembahkan untuk:

1. Kedua orang tuaku tercinta, Ayahanda Slamet Khusnul Huluq, Ibunda Nurhida Yuniati, adikku tercinta Shella Amy Niar Huluq dan seluruh keluarga besarku yang sangat aku sayangi, yang senantiasa memberikan doa, perhatian, kasih sayang, semangat, dukungan moril maupun materil untuk meraih cita-cita;
2. Bapak Kusbudiono, S.Si., M.Si., dan Bapak Ahmad Kamsyakawuni, S.Si., M.Kom., yang telah sabar dan ikhlas meluangkan waktu, pikiran, dan perhatian sehingga skripsi ini dapat terselesaikan dengan baik;
3. Bapak Ziaul Arif, S.Si., M.Sc., dan Bapak Rusli Hidayat, M.Sc., yang telah meluangkan waktu, pikiran, dan perhatian untuk menguji dan memberikan kritik saran sehingga skripsi ini dapat terselesaikan dengan baik;
4. Guru dan dosen-dosenku, yang telah memberikan ilmu dan membimbing dengan penuh kesabaran;
5. Almamater tercinta Jurusan Matematika Fakultas MIPA Universitas Jember, SMA Negeri 2 Genteng, SMP Negeri 1 Sempu, SD Negeri 2 Gendoh, dan TK Aisyiyah Bustanul Atfhal Gendoh.

MOTO

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap.”
(terjemahan Surat *Al-Insyirah* Ayat 6-8)^{*)}

“Sesungguhnya Allah tidak merubah keadaan suatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri.”
(terjemahan Surat *Ar-Rad* Ayat 11)^{**)}

*) Departemen Agama Republik Indonesia. 2014. *Al-Qur'an dan Terjemahannya*. Bandung: CV Penerbit Diponegoro.

***) Departemen Agama Republik Indonesia. 2014. *Al-Qur'an dan Terjemahannya*. Bandung: CV Penerbit Diponegoro.

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Shelda Amy Shara Huluq

NIM : 121810101058

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Penerapan Algoritma *Successive Shortest Path* dan Algoritma *Capacity Scaling* dalam Penyelesaian *Minimum Cost Flow Problem*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Desember 2016

Yang menyatakan,

Shelda Amy Shara Huluq

NIM 121810101058

SKRIPSI

**PENERAPAN ALGORITMA *SUCCESSIVE SHORTEST PATH*
DAN ALGORITMA *CAPACITY SCALING*
DALAM PENYELESAIAN *MINIMUM COST FLOW PROBLEM***

Oleh

Shelda Amy Shara Huluq
NIM 121810101058

Pembimbing

Dosen Pembimbing Utama : Kusbudiono, S.Si., M.Si

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom

PENGESAHAN

Skripsi berjudul ”Penerapan Algoritma *Successive Shortest Path* dan Algoritma *Capacity Scaling* dalam *Penyelesaian Minimum Cost Flow Problem*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas MIPA Universitas Jember.

Tim Penguji :

Dosen Pembimbing Utama,

Dosen Pembimbing Anggota,

Kusbudiono, S.Si., M.Si.

Ahmad Kamsyakawuni, S.Si., M.Kom.

NIP 197704302005011001

NIP 197211291998021001

Dosen Penguji I,

Dosen Penguji II,

M. Ziaul Arif, S.Si., M.Sc.

Drs. Rusli Hidayat, M.Sc.

NIP 198501112008121002

NIP 196610121993031001

Mengesahkan
Dekan,

Prof. Dr. Sujito, Ph.D.

NIP 196102041987111001

RINGKASAN

Penerepan Algoritma *Successive Shortest Path* dan Algoritma *Capacity Scaling* dalam Penyelesaian *Minimum Cost Flow Problem*; Shelda Amy Shara Huluq; 121810101058; 2016; x halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Network flow adalah sebuah graf berarah yang tiap sisinya memiliki kapasitas atau bobot dan pada sisi tersebut terdapat arus (*flow*) yang mengalir antara dua simpul yang mengapit sisi tersebut. *Minimum cost flow problem* merupakan model permasalahan pencarian biaya minimum yang digunakan untuk mendistribusikan suatu barang dari produsen menuju ke konsumen dalam suatu aliran jaringan. *Minimum cost flow problem* adalah masalah penentuan arus distribusi agar biaya yang dikeluarkan minimum.

Penelitian ini menggunakan data primer dari PT. Amita Bara Sejahtera Jember. Data tersebut terdiri dari 50 simpul dan 54 sisi. Simpul 1 merupakan Depo dan merupakan *source* (simpul awal), sedangkan simpul 2 sampai simpul 50 merupakan *sink* (simpul tujuan), dan sisi diasumsikan sebagai jalan dengan bobot sisi sebagai biaya distribusi dan kapasitas maksimum pengiriman. Jumlah aliran tidak boleh melebihi kapasitas maksimum pengirimannya. Hasil penelitian berupa rute, total biaya minimum, dan *running time* program.

Penerapan algoritma *successive shortest path* dan algoritma *capacity scaling* pada data pendistribusian gas LPG 3 Kg PT. Amita Bara Sejahtera Jember menghasilkan total biaya minimum algoritma *successive shortest path* sebesar Rp 326.206,-, dan total biaya minimum algoritma *capacity scaling* sebesar Rp 326.525,-. Total biaya minimum yang dihasilkan algoritma *successive shortest path* relatif lebih kecil dari pada algoritma *capacity scaling* sehingga dapat dikatakan bahwa algoritma *successive shortest path* lebih baik dari pada algoritma

capacity scaling. Sedangkan, jika dilihat dari running time yang diperoleh, *capacity scaling* lebih baik daripada algoritma *successive shortest path*.



PRAKATA

Puji syukur kehadiran Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Penerapan Algoritma *Successive Shortest Path* dan Algoritma *Capacity Scaling* dalam Penyelesaian *Minimum Cost Flow Problem*”. Shalawat serta salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW.

Skripsi ini diajukan untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Penulisan skripsi ini tidak lepas dari hambatan dan kesulitan, namun berkat bimbingan, bantuan, nasihat, saran, dan kerjasama dari berbagai pihak, segala hambatan dan kesulitan tersebut dapat diatasi dengan baik. Oleh karena itu, penulis ingin menyampaikan rasa terimakasih yang sebesar-besarnya kepada:

1. Bapak Kusbudiono, S.Si., M.Si., selaku Dosen Pembimbing Utama dan Bapak Ahmad Kamsyakawuni, S.Si., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penyusunan skripsi ini;
2. Bapak Prof. I Made Tirta, M.Sc., Ph.D., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
3. Bapak M. Ziaul Arif, S.Si., M.Sc., selaku Dosen Penguji I dan Bapak Drs. Rusli Hidayat, M.Sc., selaku Dosen Penguji II
4. Bapak/Ibu Dosen Jurusan Matematika Fakultas MIPA yang telah membimbing, mendidik dan menyalurkan banyak ilmu selama perkuliahan;
5. Bapak/Ibu Guru sejak TK hingga SMA yang dengan sabar telah mendidik dan menyalurkan banyak ilmu;
6. seluruh keluarga besar Jurusan Matematika Angkatan 2012 Bathics'12 dan keluarga besar Kos Kalimantan X No. 135 yang senantiasa memberikan support dan semangat;

7. seluruh karyawan di lingkungan Fakultas Matematika dan Ilmu Pengetahuan Alam;
8. semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu hingga terselesaikannya skripsi ini.

Semoga segala bantuan yang telah diberikan menjadi amal baik dan mendapat barokah dari Allah SWT. Penulisan skripsi ini tidak lepas dari kekurangan, baik aspek kualitas maupun kuantitas dari materi yang disajikan dan diteliti, serta masih jauh dari kata sempurna. Sehingga penulis membutuhkan banyak kritik dan saran yang bersifat membangun demi kemajuan pendidikan di masa mendatang. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Desember 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBING	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Graf	4
2.1.1 Definisi Graf	4
2.1.2 Jenis-jenis Graf	5
2.2 <i>Network Flow</i>	6
2.3 <i>Minimum Cost Flow Problem</i>	7
2.4 <i>Algoritma Successive Shortest Path</i>	8
2.5 <i>Algoritma Capacity Scaling</i>	9
BAB 3. METODE PENELITIAN	12
3.1 Data Penelitian	12

3.2 Langkah -langkah Penelitian	12
BAB 4. HASIL DAN PEMBAHASAN	17
4.1 Hasil	17
4.1.1 Penyelesaian <i>Minimum Cost Flow Problem</i> pada Data Kecil.....	17
4.1.2 Penyelesaian <i>Minimum Cost Flow Problem</i> Menggunakan Program MATLAB.....	78
4.2 Pembahasan	83
BAB 5. PENUTUP	85
5.1 Kesimpulan	85
5.2 Saran	85
DAFTAR PUSTAKA	86
LAMPIRAN	87

DAFTAR TABEL

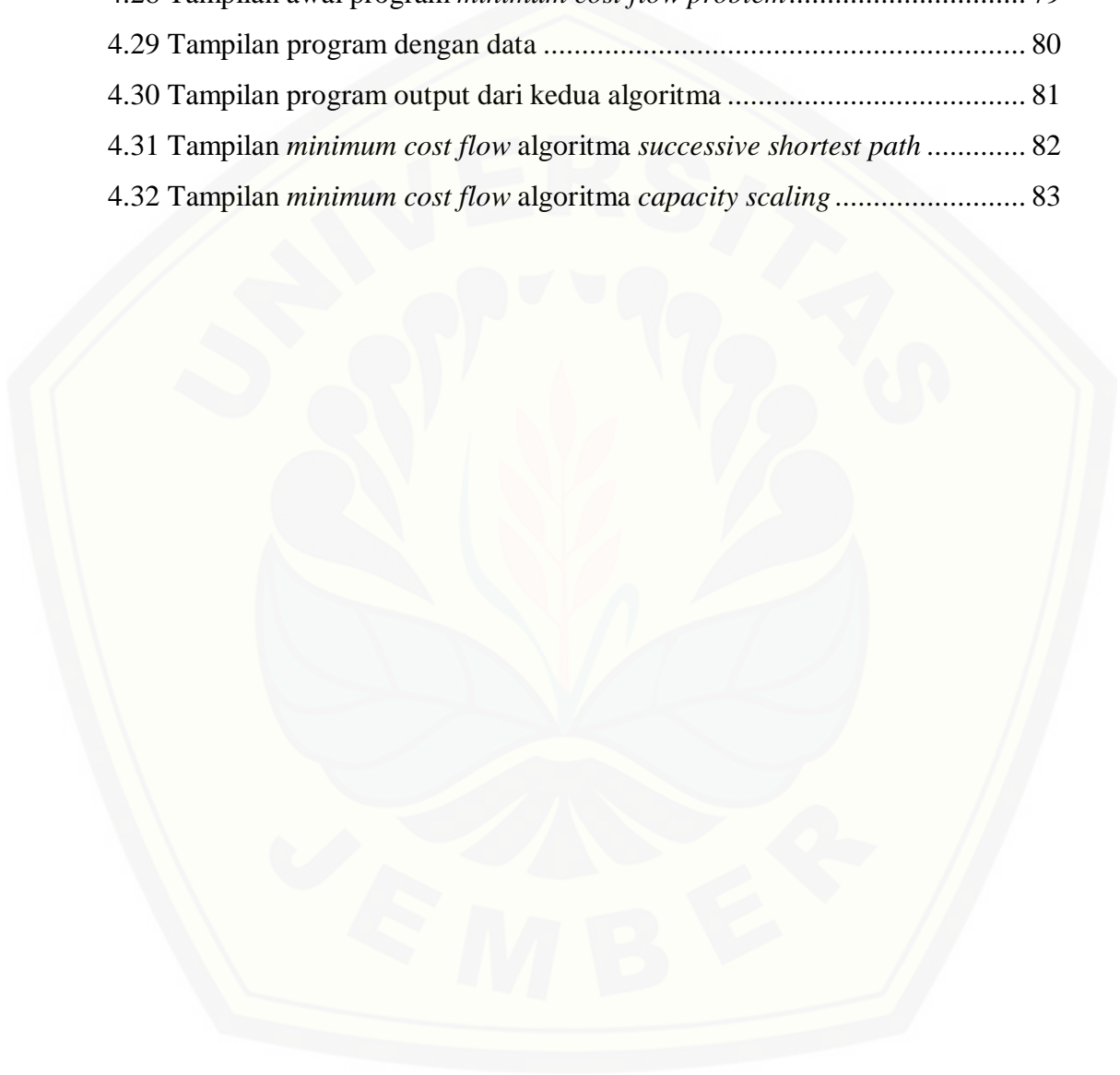
	Halaman
4.1 Hasil <i>running</i> percobaan dari kedua algoritma.....	84



DAFTAR GAMBAR

	Halaman
2.1 Dua buah graf (a) (G_1) graf ganda, (b) (G_2) graf semu	5
3.1 Skema metode penelitian.....	16
4.1 Jaringan awal	17
4.2 Jaringan sisa setelah dialirkan 132 unit melalui $1 \rightarrow 2$	22
4.3 Jaringan sisa setelah dialirkan 84 unit melalui $1 \rightarrow 3$	25
4.4 Jaringan sisa setelah dialirkan 140 unit melalui $1 \rightarrow 4$	28
4.5 Jaringan sisa setelah dialirkan 64 unit melalui $1 \rightarrow 5$	31
4.6 Jaringan sisa setelah dialirkan 140 unit melalui $1 \rightarrow 2 \rightarrow 6$	33
4.7 Jaringan sisa setelah dialirkan 75 unit melalui $1 \rightarrow 2$	37
4.8 Jaringan sisa setelah dialirkan 75 unit melalui $1 \rightarrow 3$	39
4.9 Jaringan sisa setelah dialirkan 75 unit melalui $1 \rightarrow 4$	41
4.10 Jaringan sisa setelah dialirkan 75 unit melalui $1 \rightarrow 2 \rightarrow 6$	43
4.11 Jaringan sisa setelah dialirkan 38 unit melalui $1 \rightarrow 2$	45
4.12 Jaringan sisa setelah dialirkan 38 unit melalui $1 \rightarrow 4$	47
4.13 Jaringan sisa setelah dialirkan 38 unit melalui $1 \rightarrow 5$	49
4.14 Jaringan sisa setelah dialirkan 38 unit melalui $1 \rightarrow 2 \rightarrow 6$	51
4.15 Jaringan sisa setelah dialirkan 19 unit melalui $1 \rightarrow 2$	53
4.16 Jaringan sisa setelah dialirkan 19 unit melalui $1 \rightarrow 4$	55
4.17 Jaringan sisa setelah dialirkan 19 unit melalui $1 \rightarrow 5$	57
4.18 Jaringan sisa setelah dialirkan 19 unit melalui $1 \rightarrow 2 \rightarrow 6$	59
4.19 Jaringan sisa setelah dialirkan 5 unit melalui $1 \rightarrow 3$	61
4.20 Jaringan sisa setelah dialirkan 5 unit melalui $1 \rightarrow 4$	63
4.21 Jaringan sisa setelah dialirkan 5 unit melalui $1 \rightarrow 5$	65
4.22 Jaringan sisa setelah dialirkan 5 unit melalui $1 \rightarrow 2 \rightarrow 6$	67
4.23 Jaringan sisa setelah dialirkan 3 unit melalui $1 \rightarrow 3$	69

4.24 Jaringan sisa setelah dialirkan 3 unit melalui $1 \rightarrow 4$	71
4.25 Jaringan sisa setelah dialirkan 3 unit melalui $1 \rightarrow 2 \rightarrow 6$	73
4.26 Jaringan sisa setelah dialirkan 2 unit melalui $1 \rightarrow 5$	75
4.27 Jaringan sisa setelah dialirkan 1 unit melalui $1 \rightarrow 3$	77
4.28 Tampilan awal program <i>minimum cost flow problem</i>	79
4.29 Tampilan program dengan data	80
4.30 Tampilan program output dari kedua algoritma	81
4.31 Tampilan <i>minimum cost flow</i> algoritma <i>successive shortest path</i>	82
4.32 Tampilan <i>minimum cost flow</i> algoritma <i>capacity scaling</i>	83



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Teori graf merupakan salah satu cabang matematika yang banyak manfaat karena teori-teorinya dapat diterapkan untuk memecahkan masalah dalam kehidupan sehari-hari (Purwanto,1998). Manfaat teori graph dalam kehidupan sehari-hari salah satunya adalah pemanfaatan teori graf untuk menyelesaikan masalah aliran jaringan atau biasa disebut *network flow*. *Network flow* adalah suatu model yang berkaitan dengan memaksimalkan atau meminimumkan suatu jaringan dengan syarat tertentu sehingga mendapatkan hasil yang optimal. Ada beberapa macam model permasalahan yang termasuk dalam masalah *network flow* diantaranya adalah masalah rute terpendek (*shortest path problem*), masalah diagram pohon minimum (*minimum spanning tree problem*), masalah aliran maksimum (*maximum flow problem*), dan masalah aliran biaya minimum (*minimum cost flow problem*). *Minimum cost flow problem* merupakan model permasalahan pencarian biaya minimum yang digunakan mendistribusikan suatu barang dari produsen menuju ke konsumen dalam suatu aliran jaringan. Persoalan distribusi menjadi sangat penting karena merupakan jalan untuk mencapai keberhasilan penjualan, dan kepuasan pelanggan (Woodward, 1982). Pada prinsipnya, *minimum cost flow problem* dapat digunakan sebagai unsur perencanaan pengoptimalan distribusi produk.

Minimum cost flow problem adalah masalah penentuan arus distribusi agar biaya yang dikeluarkan minimum. *Minimum cost flow problem* merupakan model permasalahan pencarian biaya minimum yang digunakan untuk mendistribusikan suatu barang dari produsen menuju ke konsumen dalam suatu aliran jaringan. Permasalahan *minimum cost flow* merupakan permasalahan yang mencari biaya minimum untuk mengirimkan suatu barang/objek dari suatu titik s ke titik t . Titik s

disebut dengan titik sumber (*supply*), sedangkan titik t disebut titik tujuan (*demand*) (Rossen, 2000).

Banyak penelitian yang telah dilakukan berkaitan dengan penyelesaian masalah *minimum cost flow*, diantaranya adalah penelitian Fajar (2012) membahas tentang penerapan algoritma *cost scaling* pada permasalahan *minimum cost flow*. Firdaus (2013) membahas tentang penerapan algoritma *successive shortest path* pada permasalahan *minimum cost flow* yang merupakan studi kasus dari KUD Lemahbang, Sumber Rejo, Sukorejo, Pasuruan untuk mencari biaya minimal pendistribusian pupuk dari KUD menuju kios-kios, dari penelitian tersebut dihasilkan biaya distribusi yang lebih murah dibandingkan dengan biaya distribusi yang selama ini dikeluarkan oleh KUD. Saputri (2013) membahas tentang algoritma *capacity scaling* pada permasalahan *minimum cost flow problem*.

Menurut Firdaus (2013) kelebihan algoritma *successive shortest path* dibandingkan dengan algoritma *cost scaling* yaitu mudah dipahami dan iterasinya lebih pendek. Sedangkan algoritma *capacity scaling* baik digunakan pada jaringan yang memuat kapasitas sisi cukup besar (Ahuja, 1993). Ide dasar algoritma *capacity scaling* ini adalah menambahkan aliran sepanjang lintasan yang memiliki kapasitas sisa cukup besar. Menurut Saputri (2013) biaya minimum yang dihasilkan algoritma *capacity scaling* sama dengan biaya minimum yang dihasilkan oleh algoritma penghapusan siklus, sehingga algoritma *capacity scaling* tersebut dapat dijadikan alternatif dalam menyelesaikan permasalahan *minimum cost flow*. Berdasarkan uraian diatas, penulis tertarik untuk melakukan penelitian tentang penerapan algoritma *successive shortest path* dan algoritma *capacity scaling* dalam penyelesaian *minimum cost flow problem*.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada penelitian ini jika ditinjau dari latar belakang di atas adalah sebagai berikut:

- a. bagaimana penyelesaian *minimum cost flow problem* dengan menggunakan algoritma *successive shortest path* dan algoritma *capacity scaling*;
- b. bagaimana perbandingan hasil penyelesaian *minimum cost flow problem* menggunakan algoritma *successive shortest path* dengan algoritma *capacity scaling* berdasarkan total biaya minimum dan *running time*.

1.3 Tujuan

Berdasarkan latar belakang dan rumusan masalah di atas, adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

- a. mendapatkan penyelesaian *minimum cost flow problem* menggunakan algoritma *successive shortest path* dan algoritma *capacity scaling*;
- b. mengetahui perbandingan hasil penyelesaian *minimum cost flow problem* menggunakan algoritma *successive shortest path* dengan algoritma *capacity scaling* berdasarkan total biaya minimum dan *running time*.

1.4 Manfaat

Adapun manfaat yang diharapkan dari penulisan tugas akhir ini adalah:

- a. mendapatkan solusi *minimum cost flow problem* dengan menggunakan algoritma *successive shortest path* dan algoritma *capacity scaling*;
- b. memberikan referensi dan wawasan lebih dalam bagi pembaca tentang algoritma *successive shortest path* dan algoritma *capacity scaling* dalam penyelesaian *minimum cost flow problem*.

BAB 2. TINJAUAN PUSTAKA

2.1 Teori Graf

Teori graf merupakan pokok bahasan yang memiliki banyak terapan. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai bulatan atau titik. Sedangkan hubungan antara objek dinyatakan dengan garis (Munir, 2010).

2.1.1 Definisi Graf

Sebuah graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertexs* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul. Sebuah graf dimungkinkan tidak mempunyai sisi sama sekali, tetapi simpulnya harus ada. Simpul pada graf dapat dinomori dengan huruf, seperti a, b, c, \dots, V, Z atau dengan bilangan asli 1, 2, 3, atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau dinyatakan dengan lambang $e_1, e_2, e_3, \dots, e_n$. Dengan kata lain, jika e adalah sisi yang menghubungkan simpul u dengan simpul v , maka e dapat ditulis sebagai $e = (u, v)$.

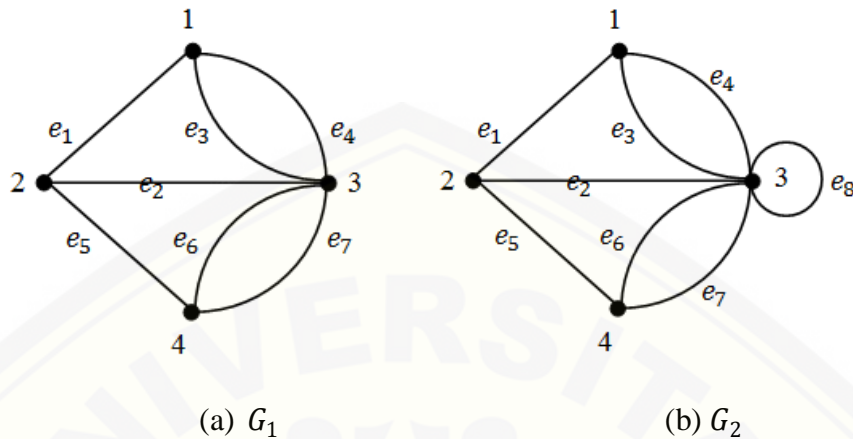
Secara geometri graf dapat digambarkan sebagai sekumpulan noktah (simpul) didalam bidang dwimatra yang dihubungkan dengan sekumpulan garis (sisi). Pada Gambar 2.1 Memperllihatkan 2 buah graf, G_1 dan G_2 . G_1 adalah graf dengan himpunan simpul V dan himpunan sisi E sebagai berikut

$$V = \{1, 2, 3, 4\}$$

$$E = \{(1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4)\} = \text{himpunan ganda} \\ = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$$

Pada G_1 sisi $e_3 = (1, 3)$ dan sisi $e_4 = (1, 3)$ dinamakan sisi ganda (*multiple edges*) karena kedua sisi ini menghubungkan dua buah simpul yang sama, yaitu

simpul 1 dan simpul 3. Pada G_2 sisi $e_8 = (3,3)$ dinamakan gelang atau kalung karena berawal dan berakhir pada simpul yang sama (Munir, 2010).



Gambar 2.1 Dua buah graf (a) (G_1) graf ganda, (b) (G_2) graf semu

2.1.2 Jenis-jenis Graf

Graf dapat dikelompokkan menjadi beberapa kategori (jenis) bergantung pada sudut pandang pengelompokannya.

Berdasarkan ada tidaknya sisi ganda, maka graf dikelompokkan menjadi dua jenis:

- Graf sederhana (*simple graph*), graf yang tidak mengandung sisi ganda graf.
- Graf tak sederhana (*unsimple graph*), graf yang mengandung sisi ganda.

Berdasarkan jumlah simpul pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis:

- Graf berhingga (*limited graph*), graf yang jumlah simpulnya n berhingga.
- Graf tak berhingga (*unlimited graph*), graf yang jumlah simpulnya n tidak berhingga banyaknya (Purwanto, 2006).

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas dua jenis :

- Graf tak berarah (*undirected graph*)

Graf yang sisinya tidak mempunyai orientasi arah disebut sebagai graf tak berarah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u, v) = (v, u)$ adalah sisi yang sama. Pada

jaringan telepon, sisi pada graf berarah menyatakan bahwa saluran telepon dapat beroperasi pada dua arah.

b. Graf berarah (*directed graph* atau *digraph*)

Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah. Sisi berarah biasa disebut dengan busur (*arc*). Pada graf berarah (u, v) dan (v, u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u, v) \neq (v, u)$ untuk busur (u, v) , simpul u dinamakan simpul asal (*initial vertex*) dan simpul v dinamakan simpul terminal (*terminal vertex*). Graf berarah sering dipakai untuk menggambarkan aliran proses, peta lalu lintas suatu kota (jalan searah atau dua arah). Pada graf berarah, gelang diperbolehkan tetapi sisi ganda tidak (Munir, 2010).

2.2 Network Flow

Network banyak dipakai dalam banyak hal untuk kegunaan yang berbeda-beda. Jaringan transportasi, jaringan listrik dan jaringan telekomunikasi adalah contoh-contoh dimana *network* ditemukan dalam kehidupan sehari-hari. Representasi *network* juga dipakai dalam produksi, distribusi, *project planning*, penempatan fasilitas, manajemen *resource* dan *financial planning*.

Network flow adalah sebuah graf berarah yang tiap sisinya memiliki kapasitas atau bobot dan pada sisi tersebut terdapat arus (*flow*) yang mengalir antara dua simpul yang mengapit sisi tersebut. Jumlah arus yang mengalir pada tiap sisi harus lebih kecil atau sama dengan kapasitas sisi tersebut. Pada aplikasinya, sebuah graf berarah sering disebut dengan *network*. Setiap arus (*flow*) yang ada dalam *network*, harus memenuhi sebuah batasannya itu arus yang masuk pada suatu simpul harus sama dengan arus yang keluar pada simpul tersebut, kecuali pada *source*, yang keluarannya lebih besar dari arus masuk, dan *sink*, yang arus masuknya lebih besar dari arus keluar sebuah *network* biasanya digunakan untuk memodelkan sistem lalu lintas, saluran pipa, atau sirkuit elektrik (Septiana, 2010).

Suatu *network* diperlukan karena memberi gambaran visual dan bantuan konseptual yang lebih jelas untuk memotret hubungan antar komponen dalam

sistem yang sering dijumpai dalam banyak kasus. Dengan berkembangnya ilmu komputer, memungkinkan penyelesaian *problem network flow* dengan bantuan *software*. Macam-macam aplikasi *network flow* antara lain:

- a. *shortest-path problem*,
- b. *minimum spanning tree problem*,
- c. *maximum flow problem*,
- d. *minimum cost flow problem* (Habibi, 2008).

2.3 Minimum Cost Flow Problem

Minimum cost flow problem merupakan model permasalahan pencarian biaya minimum yang digunakan untuk mendistribusikan suatu barang dari produsen menuju ke konsumen dalam suatu aliran jaringan. *Minimum cost flow problem* merupakan permasalahan yang mencari biaya minimum untuk mengirimkan suatu barang/objek dari suatu titik s ke titik t . Titik s disebut dengan titik sumber (*supply*), sedangkan titik t disebut titik tujuan (*demand*) (Rossen, 2000). *Minimum cost flow problem* adalah masalah penentuan arus distribusi agar biaya yang dikeluarkan minimum (Ahuja, 1993).

Misalkan $G = (N, A)$ adalah jaringan berarah dengan cost c_{ij} dan capacity u_{ij} untuk setiap sisi $(i, j) \in A$. Masing-masing titik $i \in N$ mempunyai bobot titik dan disimbolkan $b(i)$ yang menunjukkan titik tersebut titik pemberi (*supply*) atau titik penerima (*demand*) (Ahuja, 1993). Jika $b(i) > 0$, maka i adalah titik pemberi (*supply*), jika $b(i) < 0$ maka i adalah titik penerima (*demand*) (Rossen, 2000). *Minimum cost flow problem* dapat dinyatakan sebagai berikut:

Minimum

$$z = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.1)$$

dimana,

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = b(i), \quad \forall i \in A \quad (2.2)$$

$$0 \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in A \quad (2.3)$$

Simbol z merupakan biaya total yang diperoleh dari *minimum cost flow problem*. Biaya total yang dihasilkan adalah biaya yang minimum untuk

mengirimkan objek dari suatu titik *supply* ke titik *demand*. Biaya total ini diperoleh dari jumlah perkalian *cost* dengan kapasitas sisa yang berlawanan arah dengan sisi yang bersesuaian. Nilai-nilai pada aliran fisibel harus memenuhi kesetimbangan massa yaitu $b(i)$ /bobot titik harus sama dengan aliran fisibel yang keluar dari titik dikurangi dengan aliran fisibel yang masuk ke titik dan nilai aliran fisibel lebih besar atau sama dengan nol dan kurang dari atau sama dengan *capacity*.

Biaya (*cost*) adalah bilangan non negatif yang menyatakan besarnya biaya untuk memindahkan muatan dari satu titik ke titik yang lain. Kapasitas (*capacity*) dari suatu sisi adalah jumlah maksimum muatan yang dapat dikirim pada sisi-sisi. Jaringan sisa $G(x)$ dalam suatu aliran x didefinisikan dengan mengganti masing-masing sisi berarah $(i, j) \in A$ oleh dua sisi berarah (i, j) dan (j, i) . Sisi berarah (i, j) mempunyai biaya c_{ij} dan kapasitas sisa $r_{ij} = u_{ij} - x_{ij}$, dan sisi berarah (j, i) mempunyai biaya $-c_{ij}$ dan kapasitas sisa $r_{ij} = x_{ij}$ (Rosen, 2000). Pada jaringan sisa, jika kapasitas sisa suatu sisi bernilai nol, maka sisi tersebut dapat dihapus dari jaringan.

Potensial titik i adalah besarnya $\pi(i)$ yang bersesuaian dengan batas kesetimbangan umum pada titik i . Untuk himpunan potensial titik yang diberikan, *reduced cost* dari suatu sisi (i, j) pada jaringan sisa $G(x)$ adalah

$$c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j) \quad (2.4)$$

2.4 Algoritma *Successive Shortest Path*

Pada algoritma lintasan terpendek berulang (*successive shortest path*) memakai istilah ketidakseimbangan (*imbalance*) yang dinotasikan dengan $e(i)$, biaya tereduksi dinotasikan dengan c_{ij}^{π} , untuk *pseudoflow* dapat dinotasikan dengan x , serta potensial titik dapat dinotasikan dengan $\pi(i)$. Untuk setiap *pseudoflow* x didefinisikan *imbalance* dari titik i sebagai berikut,

$$e(i) = b(i) - \sum_{\{j | (i,j) \in A\}} x_{ij} + \sum_{\{j | (j,i) \in A\}} x_{ji}, \forall i \in N \quad (2.5)$$

Untuk c_{ij}^{π} dapat diperoleh dengan persamaan (2.4), sedangkan dalam algoritma lintasan terpendek berulang, $\pi(i)$ diperoleh dari,

$$\pi(i) = \pi(i) - d(i) + d(l) \quad (2.6)$$

dimana $d(i)$ merupakan panjang lintasan terpendek (*shortest path*) dari titik l ke titik i , dan titik l merupakan titik dengan $e(i) < 0$ (Ahuja,1993).

Untuk menyelesaikan masalah aliran biaya minimum (*Minimum cost flow*) menggunakan algoritma *successive shortest path* atau lintasan pendek berulang, dengan langkah-langkah sebagai berikut:

- a. menentukan nilai awal *pseudoflow* $x_{ij} = 0$ dan nilai awal $\pi(i) = 0, \forall i \in N$;
- b. menentukan nilai $e(i)$;
- c. menentukan biaya tereduksi awal c_{ij}^{π} ;
- d. menentukan himpunan E dan himpunan D ;
- e. memilih titik $k \in E$ dan $l \in D$;
- f. menentukan $d(i)$ bobot lintasan terpendek (*shortest path*) dari titik s ke semua titik di N berdasarkan biaya tereduksi;
- g. menentukan lintasan terpendek (*shortest path*) P dari titik k ke titik l berdasarkan biaya tereduksi;
- h. mengupdate $\pi(i)$ dan update nilai c_{ij}^{π} ;
- i. menentukan nilai δ ,

$$\delta = \min[e(k), -e(l), \min\{r_{ij} : (i, j) \in P\}] \quad (2.7)$$
- j. mengalirkan δ unit sepanjang lintasan P ;
- k. mengupdate $x, G(x), E, D$;
- l. melakukan perulangan tahap kedua hingga diperoleh $E = \emptyset$;
- m. setelah diperoleh $E = \emptyset$, dilakukan penghitungan total biaya minimum distribusi (Akbar, 2013).

2.5 Algoritma *Capacity Scaling*

Algoritma *capacity scaling* ini pertama kali ditemukan oleh Gabow pada tahun 1985, dan kemudian disempurnakan oleh Ahuja dan Orlin pada tahun 1991. Pada salah satu jurnal berjudul “*A Capacity Scaling for The Constrained Maximum Flow Problem*” yang ditulis oleh Ahuja dan Orlin (1993) menyebutkan bahwa algoritma *capacity scaling* dapat digunakan dalam

menyelesaikan permasalahan *maximum flow*. Ide pokok yang mendasari algoritma *capacity scaling* ini adalah menambahkan aliran sepanjang lintasan yang memiliki kapasitas sisa cukup besar. Algoritma *capacity scaling* ini baik digunakan untuk jaringan yang memiliki kapasitas sisi cukup besar.

Adapun langkah-langkah yang terdapat dalam buku *Network Flow* (Ahuja, dkk, 1993) yang harus dilakukan dalam algoritma *capacity scaling* ini adalah sebagai berikut:

Input: suatu *network* N yang telah diketahui bobot masing-masing titik $b(i)$, biaya dari suatu titik ke titik yang lain c_{ij} , dan kapasitas pengiriman dari suatu titik ke titik yang lain u_{ij} .

a. menentukan jaringan sisa, dengan mengalirkan aliran setiap sisinya $x_{ij} = 0$ dan potensial setiap titiknya adalah $\pi(i) = 0$. Sehingga dari jaringan sisa tersebut akan didapatkan ketakseimbangan titik $e(i)$, biaya sisa c_{ij}^π , dan kapasitas sisa r_{ij} ;

b. menentukan nilai Δ ,

$$\Delta = 2^{\lceil \log_2 U \rceil} \quad (2.8)$$

Δ nantinya akan menjadi banyaknya unit yang dialirkan pada setiap lintasan yang terpilih. Jika $\Delta \geq 1$ maka lanjut ke langkah 3, jika $\Delta < 1$ maka algoritma berhenti;

c. apabila kapasitas sisa lebih dari atau sama dengan Δ ($r_{ij} \geq \Delta$) dan biaya sisa kurang dari 0 ($c_{ij}^\pi < 0$) maka alirkan sebanyak r_{ij} dan update ketakseimbangan titik $e(i)$, aliran x_{ij} , dan kapasitas sisa x_{ij} dan lanjut ke langkah 4. Apabila tidak terjadi kasus tersebut, maka langsung ke langkah 4;

d. menentukan $S(\Delta)$ dan $T(\Delta)$ dengan ketentuan:

$S(\Delta)$ merupakan himpunan titik yang mempunyai ketakseimbangan titik sebanyak lebih dari atau sama dengan Δ atau merupakan himpunan titik pengirim. Sedangkan $T(\Delta)$ merupakan himpunan titik yang memiliki ketakseimbangan titik sebanyak kurang dari atau sama dengan $-\Delta$ atau merupakan himpunan titik penerima.

$$S(\Delta) = \{i \in N : e(i) \geq \Delta\}$$

$$T(\Delta) = \{i \in N : e(i) \leq -\Delta\}$$

Apabila $S(\Delta) \neq \emptyset$ dan $T(\Delta) \neq \emptyset$ maka lanjut ke langkah 5. Apabila $S(\Delta) \neq \emptyset$ dan $T(\Delta) = \emptyset$ maka langsung ke langkah 7;

e. ketika $S(\Delta) \neq \emptyset$ dan $T(\Delta) \neq \emptyset$ maka lakukan langkah berikut:

- 1) pilih $k \in S(\Delta)$ dan $l \in T(\Delta)$;
 - 2) update biaya sisa $c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j)$;
 - 3) update jarak $d(j) = d(i) + c_{ij}^\pi$;
 - 4) pilih P yang merupakan lintasan terpendek dari titik k ke titik l pada jaringan sisa Δ .
Pemilihan lintasan dilakukan dengan cara melihat $d(l)$ yaitu dengan mencari lintasan dengan biaya sisa minimum yang sama dengan $d(l)$;
 - 5) update potensial titik $\pi(i) = \pi(i) - d(i)$;
 - 6) alirkan sebanyak Δ pada lintasan terpilih P ;
 - 7) update ketidakseimbangan titik $e(i)$;
 - 8) update kapasitas sisa $r_{ji} = r_{ij} - \Delta$. Apabila pada lintasan ada sisi (j, i) , maka $r_{ij} = r_{ij} + \Delta$;
 - 9) update $x_{ij} = x_{ij} + \Delta$. Apabila pada lintasan ada sisi (j, i) , maka $x_{ij} = x_{ij} - \Delta$;
- f. lakukan langkah 5 sampai $T(\Delta)$ terpilih semua;
- g. apabila $T(\Delta)$ sudah terpilih semua, update Δ dengan $\Delta = \frac{\Delta}{2}$;
- h. kembali ke langkah 3;

Output: jumlah dari aliran x_{ij} yang masing-masing dikalikan dengan biaya sisinya c_{ij} , $\forall (i, j) \in A$ (Saputri, 2013).

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan adalah data primer yang diperoleh dari PT. Amita Bara Sejahtera Jember, yang merupakan salah satu usaha dagang yang bergerak di bidang perdagangan gas LPG 3 Kg dan melakukan pengiriman barang ke toko dalam lingkup kota Jember. Data yang digunakan sebagai berikut:

- a. jumlah toko tujuan, adapun jumlah toko yang digunakan untuk penelitian ini yaitu 49 toko tujuan dengan 1 Depo sebagai toko pengirim;
- b. banyaknya permintaan setiap toko tujuan (Lampiran A);
- c. kapasitas maksimum angkutan (Lampiran A);
- d. biaya distribusi (Lampiran A).

3.2 Langkah-langkah Penelitian

Pada penelitian ini dibahas tentang penggunaan algoritma *successive shortest path* dan algoritma *capacity scaling* dalam mencari *minimum cost flow problem*. Adapun langkah-langkah yang dilakukan dalam penelitian ini dapat dilihat secara skematis pada Gambar 3.1.

a. Studi literatur

Penelitian ini diawali dengan mempelajari buku-buku, jurnal, artikel dan sumber lain yang membahas tentang *minimum cost flow problem*, algoritma *successive shortest path* dan algoritma *capacity scaling*.

b. Pengambilan data

Pengambilan data tentang pendistribusian barang dari PT. Amita Bara Sejahtera Jember berupa daftar toko tujuan, banyaknya permintaan setiap toko tujuan, kapasitas maksimum angkutan, dan biaya distribusi.

c. Mempresentasikan data dalam bentuk graf berbobot dan berarah.

Data yang telah dikumpulkan pada poin b akan dipresentasikan dalam bentuk graf berbobot dan berarah, dimana titik menyatakan toko, sisi menyatakan jalan

yang menghubungkan antar toko, bobot titik menyatakan banyaknya permintaan dan bobot sisi menyatakan besarnya biaya distribusi dan kapasitas maksimum angkutan.

d. Mengolah data dengan algoritma *successive shortest path*.

Langkah-langkah mengolah data menggunakan algoritma *successive shortest path* sebagai berikut:

- 1) menentukan nilai awal *pseudoflow* $x_{ij} = 0$ dan nilai awal $\pi(i) = 0, \forall i \in N$;
- 2) menentukan nilai $e(i)$;
- 3) menentukan biaya tereduksi awal c_{ij}^{π} ;
- 4) menentukan himpunan E dan himpunan D ;
- 5) memilih titik $k \in E$ dan $l \in D$;
- 6) menentukan $d(i)$ bobot lintasan terpendek (*shortest path*) dari titik s ke semua titik di N berdasarkan biaya tereduksi;
- 7) menentukan lintasan terpendek (*shortest path*) P dari titik k ke titik l berdasarkan biaya tereduksi;
- 8) mengupdate $\pi(i)$ dan update nilai c_{ij}^{π} ;
- 9) menentukan nilai δ ;
- 10) mengalirkan δ unit sepanjang lintasan P ;
- 11) mengupdate $x, G(x), E, D$;
- 12) melakukan perulangan tahap kedua hingga diperoleh $E = \emptyset$;
- 13) setelah diperoleh $E = \emptyset$, dilakukan penghitungan total biaya minimum distribusi.

e. Mengolah data dengan algoritma *capacity scaling*.

Langkah-langkah mengolah data menggunakan algoritma *capacity scaling* sebagai berikut :

Input: suatu *network* N yang telah diketahui bobot masing-masing titik $b(i)$, biaya dari suatu titik ke titik yang lain c_{ij} , dan kapasitas pengiriman dari suatu titik ke titik yang lain u_{ij} .

- 1) menentukan jaringan sisa, dengan mengalirkan aliran setiap sisinya $x_{ij} = 0$ dan potensial setiap titiknya adalah $\pi(i) = 0$. Sehingga dari jaringan sisa tersebut akan didapatkan ketakseimbangan titik $e(i)$, biaya sisa c_{ij}^π , dan kapasitas sisa r_{ij} ;
- 2) menentukan nilai Δ , Δ nantinya akan menjadi banyaknya unit yang pada setiap lintasan yang terpilih. Jika $\Delta \geq 1$ maka lanjut ke langkah 3, jika $\Delta < 1$ maka algoritma berhenti;
- 3) apabila kapasitas sisa lebih dari atau sama dengan Δ ($r_{ij} \geq \Delta$) dan biaya sisa kurang dari 0 ($c_{ij}^\pi < 0$) maka alirkan sebanyak r_{ij} dan update ketakseimbangan titik $e(i)$, aliran x_{ij} , dan kapasitas sisa r_{ij} dan lanjut ke langkah 4. Apabila tidak terjadi kasus tersebut, maka langsung ke langkah 4;
- 4) menentukan $S(\Delta)$ dan $T(\Delta)$ dengan ketentuan:
 $S(\Delta)$ merupakan himpunan titik yang mempunyai ketakseimbangan titik sebanyak lebih dari atau sama dengan Δ atau merupakan himpunan titik pengirim. Sedangkan $T(\Delta)$ merupakan himpunan titik yang memiliki ketakseimbangan titik sebanyak kurang dari atau sama dengan $-\Delta$ atau merupakan himpunan titik penerima. Apabila $S(\Delta) \neq \emptyset$ dan $T(\Delta) \neq \emptyset$ maka lanjut ke langkah 5. Apabila $S(\Delta) = \emptyset$ dan $T(\Delta) = \emptyset$ maka langsung ke langkah 7;
- 5) ketika $S(\Delta) \neq \emptyset$ dan $T(\Delta) \neq \emptyset$ maka lakukan langkah berikut:
 - a) pilih $k \in S(\Delta)$ dan $l \in T(\Delta)$;
 - b) update biaya sisa $c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j)$;
 - c) update jarak $d(j) = d(i) + c_{ij}^\pi$;
 - d) pilih P yang merupakan lintasan terpendek dari titik k ke titik l pada jaringan sisa $-\Delta$. Pemilihan lintasan dilakukan dengan cara melihat $d(l)$ yaitu dengan mencari lintasan dengan biaya sisa minimum yang sama dengan $d(l)$;
 - e) update potensial titik $\pi(i) = \pi(i) - d(i)$;
 - f) alirkan sebanyak Δ pada lintasan terpilih P ;

- g) update ketidakseimbangan titik $e(i)$;
- h) update kapasitas sisa $x_{ji} = x_{ij} - \Delta$. Apabila pada lintasan ada sisi (j, i) , maka $r_{ij} = r_{ij} + \Delta$;
- i) update $x_{ij} = x_{ij} + \Delta$. Apabila pada lintasan ada sisi (j, i) , maka $x_{ij} = x_{ij} - \Delta$;

6) lakukan langkah 5 sampai $T(\Delta)$ terpilih semua;

7) apabila $T(\Delta)$ sudah terpilih semua, update Δ dengan $\Delta = \frac{\Delta}{2}$;

8) kembali ke langkah 3;

Output: jumlah dari aliran x_{ij} yang masing-masing dikalikan dengan biaya sisinya $c_{ij}, \forall (i, j) \in A$ (Saputri, 2013).

f. Membuat program

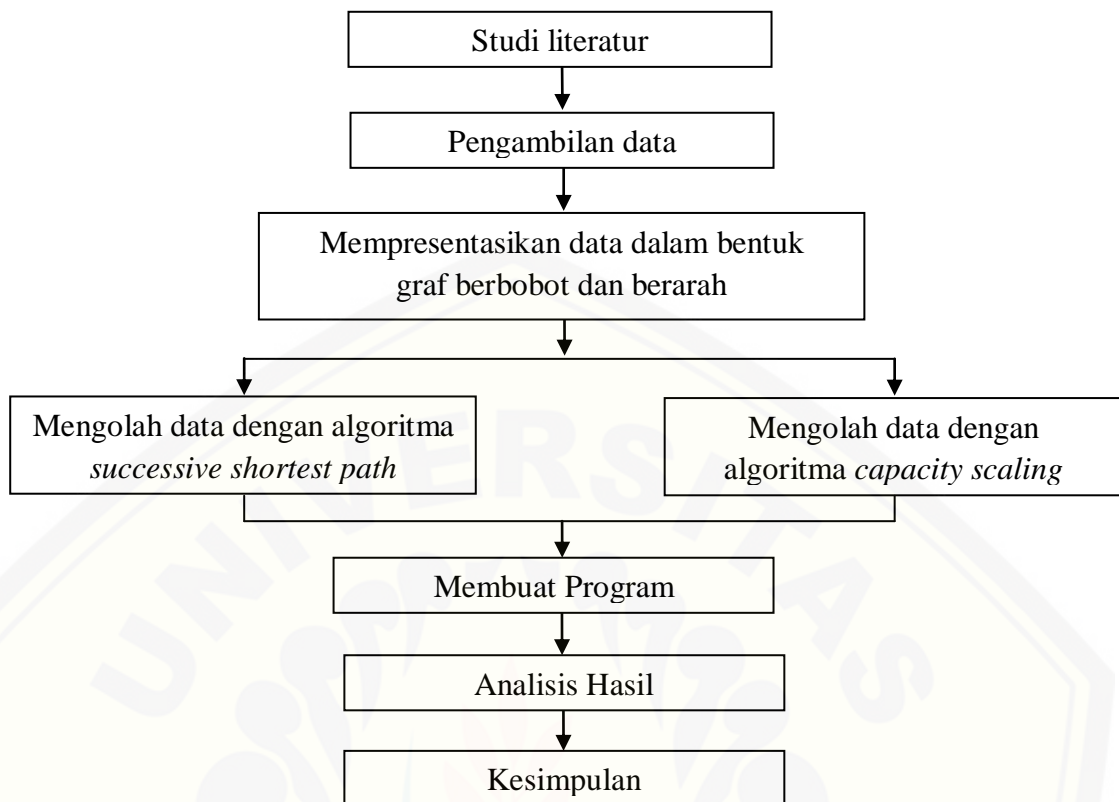
Pembuatan program menggunakan *software* MATLAB, pada langkah ini dibuat skrip program dan desain tampilan GUI sesuai dengan algoritma yang digunakan.

g. Analisis hasil

Peneliti akan membandingkan hasil *output* program antara algoritma *successive shortest path* dan algoritma *capacity scaling* berdasarkan total biaya paling minimum.

h. Kesimpulan

Kesimpulan yang diambil berdasarkan analisis yang telah diperoleh.



Gambar 3.1 Skema metode penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang diperoleh dapat disimpulkan beberapa hal sebagai berikut:

- a. Penyelesaian *minimum cost flow problem* dengan menggunakan algoritma *successive shortest path* dan algoritma *capacity scaling* pada data pendistribusian gas LPG 3 Kg PT. Amita Bara Sejahtera Jember menghasilkan total biaya minimum algoritma *successive shortest path* sebesar Rp 326.206,-, dan total biaya minimum algoritma *capacity scaling* sebesar Rp 326.525,-.
- b. Penerapan algoritma *successive shortest path* dan algoritma *capacity scaling* pada data pendistribusian gas LPG 3 Kg PT. Amita Bara Sejahtera Jember menunjukkan bahwa total biaya minimum yang dihasilkan algoritma *successive shortest path* relatif lebih kecil dari pada algoritma *capacity scaling*.
- c. *Running time* algoritma *capacity scaling* lebih baik dari pada algoritma *successive shortest path* karena *running time* yang dihasilkan algoritma *capacity scaling* lebih kecil dari pada algoritma *successive shortest path*.

5.2 Saran

Berdasarkan kesimpulan di atas, penulis menyarankan kepada peneliti lain untuk mengembangkan dan menerapkan algoritma lain untuk menyelesaikan *minimum cost flow problem*. Peneliti selanjutnya juga dapat mengembangkan algoritma *successive shortest path* dan algoritma *capacity scaling* dalam penentuan lintasan terpendek.

DAFTAR PUSTAKA

- Ahuja, R.K., Magnanti, T.L., dan Orlin, J.B. 1993. *Network Flow Theory, Algorithms, and Applications*. New Jersey: Prentice Hall, Inc.
- Ahuja, R.K, dan Orlin, J.B. 1993. *A Capacity Scaling for The Constrained Maximum Flow Problem*.
- Akbar, F.R., dan Widodo, A. 2013. “Optimasi Biaya Distribusi Pupuk Menggunakan Successive Shortest Algorithm dalam Masalah Minimum Cost Flow Problem (Studi Kasus KUD Lemahbang, Sumber Rejo, Sukorejo, Pasuruan)”. Tidak Diterbitkan. Skripsi. Malang: Universitas Brawijaya.
- Prabowo, F., 2012. “Penerapan Algoritma Cost Scaling pada Permasalahan Minimum Cost Flow Problem dan Implementasinya pada Program”. Tidak Diterbitkan. Skripsi. Malang: Universitas Negeri Malang.
- Habibi, L. 2008. “Pemodelan Network Flow Analysis Sistem Distribusi Air Menggunakan Algoritma Genetika – Metode Newton”. Tidak Ditebitkan. Tesis. Bandung: ITB.
- Johsohnbaugh, R. 2001. *Discrete Mathematisc*. New Jersey : Prentice Hall, Inc.
- Munir, R. 2010. *Matematika Diskrit*. Bandung: Informatika Bandung.
- Purwanto. 1998. *Teori Graph*. Malang: FPMIPA IKIP Malang.
- Rossen, K.H. 2000. *Handbook of Discrete and Combinatorial Mathematics*. Washington, D. C.: CRC Press.
- Saputri, R.D. 2013. “Algoritma Capacity Scaling dalam Menyelesaikan Minimum Cost Flow Problem dan Implementasi Programnya”. Tidak Diterbitkan. Skripsi. Malang: Universitas Negeri Malang.
- Septiana, E.P., dan Ketut, E. 2010. *Pengembangan Simulasi Aliran Air Pada Saluran Drainase Kota Menggunakan Pemodelan Network Flow*. Surabaya: ITS.
- Woodward, H.F. 1972. *Manajemen Transpor*. Terjemahan oleh Ny. P. Hadinoto. 1982. Jakarta: Pustaka Binaman Pressindo.

LAMPIRAN

LAMPIRAN A. Data Permintaan, Biaya Distribusi dan Kapasitas Maksimum Angkutan

Lampiran A.1 Data Permintaan Toko Tujuan

Kode Titik	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
Jumlah																										
Permintaan	1067	10	10	51	20	37	23	9	25	66	10	9	10	15	15	7	25	24	7	60	9	14	20	52	11	
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50		
13	7	11	5	15	10	15	20	12	10	9	14	148	18	22	10	6	18	25	54	20	30	13	11	12		

LAMPIRAN B. Data Alamat Depo dan Toko Tujuan

Kode Titik	Nama Toko	Alamat	Kelurahan	Kecamatan
1	Depo	Jl. Tawangmangu No. 06		
2	Abdollah / Iva	DUSUN KRAJAN BARAT RT/RW 01/01	CANDIJATI	ARJASA
3	Adim Dr. Subandi	JL. Dr. SUBANDI	PATRANG	
4	Adip	DUSUN TENGIR TIMUR RT/RW 004/001	TENGIR TIMUR	JELBUK
5	Ali G. Mada	JL. GAJAH MADA		KALIWATES
6	Anton	JL. SUMATRA 2/2 RT/RW 01/05	SUMBERSARI	SUMBER SARI
7	Ar	DUSUN KOJUK RT/RW 01/08	SUKOKERTO	SUKOWONO
8	Ayu / Mayang	MAYANG		
9	Barokah	DUSUN KRAJAN	SUMBER WRINGIN	SUKOWONO
10	Baru Ajung / Hartono	JL. MH. THAMRIN RT/RW 1/7	AJUNG	KALISAT
11	Bintang Mas	JL. RAYA RAUNG RT/RW 01/01	TEGALREJO	MAYANG
12	Candra PR. Kebon Agung	PERUM KEBON AGUNG		
13	Doremon Trunojoyo	JL. TRUNOJOYO		
14	Eko	JL. TAWANGMANGU RT/RW 001/008	ANTIROGO	SUMBERSARI
15	Erfan	JL BANYUWANGI	SUMBER KEJAYAN	MAYANG
16	Erna	DUSUN KRAJAN RT/RW 05/05	SUMBER WRINGIN	SUKOWONO
17	Evi	Jl. Kasuari Lingk. Kedawung Lor RT/RW 002/008	GEBANG	PATRANG
18	Fuji Ayu	JL. TAPAK SIRING	ANTIROGO	SUMBERSARI
19	Husen H.	SUMBERWRINGIN RT/RW 007/004		
20	Hariyanto	DUSUN KLAYU RT/RW 03/02	TEGAL WARU	MAYANG
21	Hotija			
22	Ido	DESA TEGAL WARU DUSUN WUYU		
23	Ismail L. Suprpto	JL. LETJEN SUPRAPTO		
24	Ita	DUSUN TENGGIR TIMUR	JELBUK	JELBUK
25	Jamik	DUSUN KRAJAN	PATEMON	PAKUSARI

UTARA				
26	Kedawung Kaswari	JL. KASUARI ,LINGK.KEDAWUNG KIDUL RT/RW 003/001	GEBANG	PATRANG
27	Komariyah	DUSUN ONJUR RT/RW 004/008 SEMPOLAN		
28	Like	DUSUN KRAJAN	SEMPOLAN	SILO
29	Maryam Cempaka	JL. KACA PIRING	GEBANG	PATRANG
30	Narto	BARATAN		
31	Nora	DUSUN KRAJAN SUMBERJATI SILO		
32	Novi/Tk.Robi	BARATAN		
33	Putra Jaya			
34	Rais Kedawung	KEDAWUNG	GEBANG	PATRANG
35	Rejeki Baru	JL. PANGANDARAN RT/RW 2/2	SUMBERSARI	SUMBER SARI
36	Ribut			
37	Ri			
38	Riro	JL SLAMET RIYADI GG SENTAL	BARATAN	PATRANG
39	Slamet Kaswari	JL CEMPAKA	GEBANG	PATRANG
40	Sudarmono			
41	Tatik	DUSUN SUMBER TENGAH RT/RW 002/001 PANDUMAN		
42	Liana	DUSUN LANCENG LEDOKOMBO LEMBENGAN RT/RW 002/007		
43	Tila	JL. PANGANDARAN RT/RW 2/2	ANTIROGO	SUMBERSARI
44	Titi Manggar	JL. MANGGAR LINGK. GEBANG PORENG		
45	Tres	JL. MANGGAR LINGK. GEBANG PORENG		
46	Wili Srikoyo	JL. SRIKOYO GG. TEGAL BATU		PATRANG
47	Wiwik			
48	Yahya (Asak)	JL DIPONEGORO	KALISAT	KALISAT
49	Yanto Anggur	JL. ANGGUR		PATRANG
50	Yanto Sempolan	JL. PLALANGAN RT/RW 002/001	SEMPOLAN	SILO