PENERAPAN ALGORITMA AFFINE SCALING DAN ANALISIS SENSITIVITAS PADA OPTIMALISASI BIAYA TRANSPORTASI

(STUDI KASUS PADA UD. TIGA PUTRI MRANGGEN)

Skripsi

disusun sebagai salah satu syarat

untuk memperoleh gelar Sarjana Sains

Program Studi Matematika

oleh

Rhahmadani Susanti

4111413036

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2017

PERNYATAAN KEASLIAN

Saya menyatakan bahwa skripsi ini bebas plagiat, kecuali yang secara tertulis dirujuk dalam skripsi ni dan disebutkan dalam daftar pustaka. Apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, 10 Oktober 2017

Yang membuat pernyataan,

Rhahmadani Susanti

4111413036

PENGESAHAN

Skripsi yang berjudul

Penerapan Algoritma Affine Scalling dan Analisis Sensitivitas Pada Optimalisasi Biaya Transportasi (Studi Kasus Pada UD. Tiga Putri Mranggen)

disusun oleh

Rhahmadani Susanti

4111413036

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi Program Studi Matematika, FMIPA UNNES pada tanggal 18 September 2017.

Panitia:

Ketua

Prof. Dr. Zaenuri, S.E., M.Si., Akt.
NIP 196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si.
NIP196807221993031005

Ketua Penguji

Amidi, S.Si., M.Pd.
NIP 198703012014041001

Anggota Penguji/
Pembimbing I

Drs. Mashuri, M.Si.
NIP 196708101992031003

Anggota Penguji/
Pembimbing II

Dra. Rahayu Budhiati V., M.Si.
NIP 196406131988032002

MOTTO DAN PERSEMBAHAN

MOTTO

Man Jadda Wajada

“Sesungguhnya bersama kesulitan itu ada kemudahan.” (Q.S. Al-Insyirah: 6)

Belajarlah pada cahaya, seberapa besar tembok penghalang, tetap saja bisa mencari celah sempit dan ruang yang sebelumnya mungkin tak terpikir.

When everything goes to hell, the people who stand by you without flinching – they are your family (Jim Butcher).

Gagal hanya terjadi jika kita menyerah (BJ. Habibie).

PERSEMBAHAN

Untuk orang tua tersayang Bapak Sugiman dan Ibu Sri Sudarni (alm) selalu mendokan, memotivasi, dan mendukung disetiap langkah saya.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul Penerapan Algoritma Affine Scaling Dan Analisis Sensitivitas Pada Optimalisasi Biaya Transportasi (Studi Kasus Pada UD. Tiga Putri Mranggen). Skripsi ini disusun sebagai salah satu syarat meraih gelar Sarjana Sains pada Program Studi Matematika, Universitas Negeri Semarang. Sholawat serta salam selalu tercurah kepada junjungan kita Nabi Muhammad SAW, semoga mendapatkan syafaat-Nya di hari akhir nanti.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan dan bimbingan dari berbagai pihak. Untuk itu, penulis ingin menyampaikan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri Mastur, S.E., M.Si., Akt., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang.
4. Drs. Mashuri, M.Si., Dosen Pembimbing Utama yang telah memberikan bimbingan, arahan, dan saran kepada penulis dalam menyusun skripsi ini.

5. Dra. Rahayu Budhiati Veronica, M.Si., Dosen Pembimbing Pendamping yang telah memberikan bimbingan, arahan, dan saran kepada penulis dalam menyusun skripsi ini.
6. Amidi, S.Si., M.Pd., dosen penguji yang telah memberikan masukan pada penulis.
7. Drs. Sugiman, M.Si., dosen wali sekaligus inspirator yang memberikan dukungan dan motivasi dalam menyelesaikan skripsi ini.
8. Bapak dan Ibu dosen Jurusan Matematika, yang telah memberikan bimbingan dan ilmu kepada penulis selama menempuh pendidikan.
9. Bapak Didik, pemilik UD. Tiga Putri yang telah membantu terlaksananya penelitian.
10. Bapak Sugiman, terimakasih untuk doa, kasih sayang, dukungan finansial yang tak hingga.
11. Ibu Sri Sudarni (alm), sumber motivasi dan kekuatan tiada henti.
12. Kakak saya, Anik Margiani dan Wahyu Wibowo, yang telah memberikan dorongan, kasih sayang, dan semangat pantang menyerah dalam penyusunan skripsi ini.
13. Bapak Wagiman dan Ibu Sulastri (alm), terimakasih untuk doa dan motivasi.
14. Partner in crime (Agnes, Eka, Esta, Iin, Nana, Nunik), teman seperjuangan selama kuliah, terimakasih untuk *bully*-an, dukungan, motivasi, doa dan bantuan yang telah diberikan.
15. Grup Rempong (Tafrikha, Shera, Tyas, Eka, Novia), teman seperjuangan dari jaman putih abu-abu, terimakasih atas bantuan dan dukungan selama ini.

16. Vitis Vinifera (Iin, Dina, Nikki, Nunik, Rizky, Tika, Mbak Vintha, Ulfa, Huda, Mbak Aul, dan Santi) teman seperjuangan dalam suka dan duka, yang penuh akan motivasi, canda dan cerita tiada henti.
17. Saudara tanpa ikatan darah (Ibu Eniarti, Bapak Puji, Alifia, Bangkit, Erna, Sani, Zubair, Rahayu, Vitria, Dian), terimakasih untuk segala hal yang telah kalian berikan selama ini.
18. Grup WhatsApp TM Underground (Mas Hen, Mas Tri, Mas Mudi, Mas Alip, Mbak Tj, Mbak Okvi, Mbak Anna, Mbak Rita, Ica, Diantina, Ardhyani, Fahmi, Nando, Amin, Unggul, Syifa, Risyaf, dan Kamud) atas pelajaran dan hiburan , serta cacian yang merupakan dukungan untuk melangkah ke depan.
19. Teman-teman mahasiswa Prodi Matematika Universitas Negeri Semarang angkatan 2013, atas bantuan serta kerja sama dalam menempuh studi.
20. Teman-teman PKL BPS Kab. Kendal dan KKN Desa Sekopek yang selalu memberikan dukungan dan semangat.
21. Kementrian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia yang telah memberikan dana pendidikan bagi penulis sehingga penulis dapat menyelesaikan studi dengan lancar dan aman.
22. Semua pihak yang turut membantu dalam penyusunan skripsi ini.

Semoga skripsi ini dapat memberikan manfaat bagi penulis dan para pembaca. Terima kasih.

Semarang, September 2017

Penulis

ABSTRAK

Susanti, Rhahmadani. 2017. *Penerapan Algoritma Affine Scalling Dan Analisis Sensitivitas Pada Optimalisasi Biaya Transportasi (Studi Kasus pada UD. Tiga Putri Mranggen).* Skripsi, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang. Pembimbing I: Drs. Mashuri, M.Si, dan Pembimbing II: Dra. Rahayu Veronica Budhiati, M.Si.

Kata Kunci : Optimasi, Program Linear, Algoritma Affine Scaling, Analisis Sensitivitas.

Algoritma Affine Scaling merupakan salah satu bagian dari metode titik dalam. Algoritma Affine Scaling merupakan suatu metode titik dalam yang memotong atau menembus interior dari daerah layak untuk mencapai solusi optimum dengan bantuan transformasi Affine Scaling. UD. Tiga Putri merupakan salah satu perusahaan industri yang bergerak dalam bidang industri batu bata. UD. Tiga Putri dalam menentukan banyaknya pengiriman batu bata dari lokasi pembakaran ke toko bangunan langganan tidak menggunakan Algoritma Affine Scaling.

Penelitian ini bertujuan untuk mengetahui model masalah transportasi dalam bentuk program linear, solusi optimal biaya transportasi dengan Algoritma Affine Scaling dan hasil analisis sensitivitas dari hasil optimal yang sudah diperoleh. Data yang diperoleh dalam penelitian ini yaitu data sekunder. Pengumpulan data dilakukan dengan cara observasi pada UD. Tiga Putri dan melakukan wawancara dengan pihak UD. Tiga Putri.

Formula model transportasi UD. Tiga Putri adalah
$Min\ z = 24x_{11} + 27x_{12} + 28x_{13} + 24x_{14} + 30x_{15} + 32x_{16} + 27x_{17} + 25x_{21} + 28x_{22} + 29x_{23} + 25x_{24} + 31x_{25} + 33x_{26} + 28x_{27} + 23x_{31} + 26x_{32} + 27x_{33} + 23x_{34} + 29x_{35} + 31x_{36} + 26x_{37} + 23x_{41} + 26x_{42} + 27x_{43} + 23x_{44} + 29x_{45} + 31x_{46} + 26x_{47} + 23x_{51} + 26x_{52} + 27x_{53} + 23x_{54} + 29x_{55} + 31x_{56} + 26x_{57} + 38x_{61} + 41x_{62} + 41x_{63} + 38x_{64} + 44x_{65} + 45x_{66} + 41x_{67} + 60x_{71} + 63x_{72} + 64x_{73} + 60x_{74} + 66x_{75} + 68x_{76} + 63x_{77}$
dengan kendala $x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} = 90000; x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} = 80000; x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} = 75000; x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} = 70000; x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} = 77000; x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{66} + x_{67} = 65000; x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} + x_{77} = 60000; x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} = 72000; x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} = 60000; x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} + x_{73} = 64000; x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} + x_{74} = 84000; x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} + x_{75} = 80000; x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} + x_{76} = 75000; x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{67} + x_{77} = 82000; x_{ij} \geq 0; i =$ 1,2,3,4,5,6,7 dan $j =$ 1,2,3,4,5,6,7.

Biaya transportasi UD. Tiga Putri pada bulan April 2017 dengan menggunakan Algoritma Affine Scaling adalah sebesar Rp. 17.033.000,-. Sedangkan biaya transportasi UD. Tiga Putri sebelum menerapkan Algoritma Affine Scaling adalah sebesar Rp. 17.900.000,-. Dengan demikian, dapat dikatakan bahwa terjadi penurunan sebesar Rp. 867.000,- artinya dapat meningkatkan laba perusahaan sebesar 4,84 %.

Kemudian dihitung biaya transportasi UD. Tiga Putri pada bulan Mei 2017 dengan menggunakan analisis sensitivitas adalah sebesar Rp. 17.187.000,-. Hasil ini sudah optimal karena kenaikan jumlah permintaan pada bulan Mei dengan diikuti kenaikan biaya transportasi tetapi tidak melebihi biaya transportasi yang telah dikeluarkan UD. Tiga Putri pada bulan April 2017. Hal ini jelas menunjukkan bahwa terjadi peningkatan laba perusahaan.

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR LAMPIRAN

Lampiran Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Optimalisasi adalah proses untuk mencapai hasil yang ideal atau optimal. Optimalisasi sangat berpengaruh dalam pengambilan keputusan, baik di bidang teknik, bidang ekonomi manajemen maupun bidang-bidang lainnya. Salah satu cara menyelesaikan optimalisasi adalah dengan Program Linear. Program linear ditemukan oleh L.W. Kantorovich pada tahun 1939 dengan metode yang masih terbatas (Susanta, 1994:23). Masalah program linear pada umumnya yang sering dijumpai adalah tentang memaksimalkan laba atau meminimalkan biaya transportasi, dan pada skripsi ini lebih ditekankan pada permasalahan untuk meminimalkan biaya transportasi.

Permasalahan-permasalahan program linear pada umumnya dapat diselesaikan dengan metode simpleks, namun metode simpleks kurang mampu menyelesaikan permasalahan program linear dengan variabel yang banyak. Ada metode yang dapat digunakan untuk menyelesaikan permasalahan program linear dengan variabel yang banyak yaitu metode titik interior (Hiller and Lieberman, 2008:163). Metode titik interior terdiri dari Algoritma Khaciyan, Algoritma Affine Scaling dan Algoritma Karmarkar (Chong and Zak, 2000:340).

Algoritma Affine Scaling ditemukan pada tahun 1967 oleh seorang matematikawan bernama Dikin. Kelebihan dari algoritma ini adalah lebih efektif dalam memecahkan masalah dengann kendala besar, lebih cepat mencapai titik optimal untuk permasalahan kendala besar dan tingkat efisiensi akan nampak apabila menggunakan program komputer. Ide dasar dari algoritma ini adalah dimulai dengan memilih suatu titik dalam awal di dalam daerah fisibel. Kemudian dilanjutkan ke arah gradien yang diproyeksikan untuk memperoleh pemecahan baru. Sehingga diperoleh penyelesaian masalah program linear yang optimal.

Setelah diperoleh solusi optimal, seringkali digunakan suatu analisis untuk mengamati perubahan-perubahan yang terjadi pada parameter serta dampaknya terhadap optimalitas. Analisis tersebut dinamakan analisis sensitivitas. Pada permasalahan program linear, nilai ruas kanan fungsi kendala (jumlah permintaan dan kapasitas) yang dihadapi tidak selalu tetap dan bisa berubah-ubah dikarenakan beberapa faktor, antara lain musim, pemasaran distribusi, kualitas, persaingan pasar, dan lain-lain. Berdasarkan hal tersebut, maka dalam penelitian ini akan dibahas tentang analisis sensitivitas dengan parameter perubahan nilai ruas kanan fungsi kendala.

Algoritma Affine Scaling telah digunakan dalam beberapa penelitian, antara lain penelitian oleh Retnojiwati (2007) tentang pembuktian bahwa algoritma affine scaling dapat digunakan untuk menyelesaikan masalah program linear. Kemudian penelitian Utomo (2013) tentang penerapan algoritma affine scaling dalam optimalisasi laba dan hasil penelitiannya menunjukkan bahwa hasil perhitungan dengan algoritma affine scaling dapat meningkatkan laba sebesar 15,57% lebih banyak daripada perhitungan yang dilakukan oleh perusahaan, yang dalam penelitian ini yaitu UD. Sumber Padi, Kepung.

Selanjutnya penelitian oleh Paseru (2015) tentang penerapan algoritma affine scaling untuk meminimalkan biaya transportasi. Hasil penelitian oleh Kristina menujukkan bahwa penerapan algoritma affine scaling lebih optimal dari perhitungan yang tidak menggunakan algoritma affine scaling. Dengan penurunan biaya transportasi sebesar 18,53% berarti perusahaan dapat meningkatkan laba sebesar penurunan biaya transportasi.

Selain dibahas mengenai Algoritma Affine Scaling, juga akan dibahas mengenai penerapanya dalam menyelesaikan masalah transportasi dan hasil analisis sensitivitas perubahan nilai ruas kanan fungsi kendala dari hasil optimal pada UD. Tiga Putri. UD. Tiga Putri merupakan salah satu perusahaan industri yang bergerak dalam bidang industri batu bata. Perusahaan ini memiliki aktivitas usaha yaitu menjual dan mendistribusikan batu bata kepada toko yang membutuhkan produk tersebut dalam usahanya. UD. Tiga Putri berada di Desa Karangsono, Kecamatan Mranggen, Kabupaten Demak.

UD. Tiga Putri menyadari bahwa persaingan semakin kompetitif. Oleh karena itu, dibutuhkan strategi yang tepat untuk menghadapi persaingan tersebut. Karena UD. Tiga Putri adalah distributor, maka dari itu strategi untuk menghadapi persaingan adalah menekan biaya transportasi seminimal mungkin. Dengan demikian diperlukan analisis mengenai biaya transportasi yang tepat agar dapat menekan atau meminimumkan biaya transportasi.

Berdasarkan latar belakang masalah di atas, penulis bermaksud melakukan penelitian yaitu berupa riset tentang Algoritma Affine Scaling serta penggunaannya dalam menyelesaikan masalah optimalisasi biaya transportasi dan hasil analisis sensitivitas perubahan nilai ruas kanan fungsi kendala dari hasil optimal pada UD. Tiga Putri di Semarang. Adapun judul yang akan diambil dalam tugas akhir ini adalah “Penerapan Metode Affine Scaling dan Analisis Sensitivitas Pada Optimalisasi Biaya Transportasi (Studi Kasus pada UD. Tiga Putri Mranggen)”. Riset ini diharapkan dapat memberikan sumbangan pengetahuan khususnya untuk ilmu pengetahuan di bidang matematika.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang akan dibahas dalam penulisan skripsi ini adalah:

1. Bagaimana memodelkan masalah transportasi UD. Tiga Putri ke dalam bentuk program linear?
2. Bagaimana solusi optimal biaya transportasi di UD. Tiga Putri dengan menggunakan Algoritma Affine Scaling?
3. Bagaimana hasil analisis sensitivitas perubahan nilai ruas kanan fungsi kendala dari hasil optimal yang sudah diperoleh dengan Algoritma Affine Scaling?

1.3 Tujuan Penelitian

Tujuan dari penulisan skripsi ini adalah:

1. Mengetahui model masalah transportasi pada UD. Tiga Putri ke dalam bentuk program linear.
2. Mengetahui solusi optimal biaya transportasi di UD. Tiga Putri dengan menggunakan Algoritma Affine Scaling.

3. Mengetahui hasil analisis sensitivitas perubahan nilai ruas kanan fungsi kendala dari hasil optimal yang telah diperoleh dengan Algoritma Affine Scaling.

1.4 Batasan Masalah

Agar dalam pembahasan penelitian ini tidak terlalu meluas, maka penulis mencantumkan pembatasan masalah sebagai berikut.

1. Biaya transportasi yang dihitung adalah biaya transportasi untuk kurun waktu satu bulan produksi.
2. Data yang digunakan adalah data sekunder berdasarkan buku administrasi dan data primer melalui observasi dan wawancara di lapangan.
3. Metode yang digunakan dalam skripsi ini adalah Algoritma Affine Scaling.
4. Fungsi kendala yang dibahas adalah kapasitas permintaan dan persediaan di gudang.
5. Analisis sensitivitas yang akan dibahas adalah analisis sensitivitas dengan parameter perubahan nilai ruas kanan fungsi kendala.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan gambaran tentang model masalah transportasi di UD. Tiga Putri ke dalam bentuk Algoritma Affine Scaling.
2. Memberikan gambaran tentang solusi optimal biaya transportasi di UD. Tiga Putri menggunakan Algoritma Affine Scaling.

3. Memberikan gambaran tentang analisis sensitivitas perubahan nilai ruas kanan fungsi kendala dari hasil optimal yang diperoleh dengan Algoritma Affine Scaling.
4. Memberikan motivasi kepada para peneliti untuk lebih banyak mengembangkan Algoritma Affine Scaling dan Analisis Sensitivitas Perubahan Nilai Ruas Kanan Fungsi Kendala sehingga ilmu pengetahuan akan semakin maju.

1.6 Sistematika Penulisan

Skripsi ini terdiri dari tiga bagian, yaitu bagian pendahuluan skripsi, isi skripsi dan bagian akhir skripsi.

1. Bagian pendahuluan skripsi

   Bagian ini berisi halaman judul, halaman persetujuan, halaman pengesahan, halaman motto, halaman persembahan, prakata, daftar isi, dan lampiran.

2. Bagian isi skripsi Bagian ini terdiri dari :

   BAB I PENDAHULUAN

   Bab ini berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat Penelitian serta Sistematika Penulisan

   BAB II LANDASAN TEORI

   Dalam bab ini membahas landasan teori yang berhubungan dengan penulisan skripsi dan penelitian terdahulu.

BAB III METODE PENELITIAN

Bab ini berisi tentang objek penelitian, jenis data, teknik pengumpulan data dan langkah-langkah pengolahan data.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Berisi tentang hasil penelitian dan pembahasan

BAB V PENUTUP

Berisi tentang simpulan dan saran

3. Bagian penutup skripsi

Berisi daftar pustaka dan lampiran – lampiran.

BAB II

LANDASAN TEORI

2.1 Matriks

2.1.1 Definisi Matriks

Menurut Howard Anton (1995:22), matriks adalah susunan segi empat siku-siku dari bilangan-bilangan. Bilangan-bilangan dalam susunan tersebut dinamakan *entri* dari matriks. Matriks banyak digunakan untuk menyelesaikan permasalahan matematika terutama dalam pemecahan persoalan-persoalan yang terdiri dari lebih dari dua persamaan dengan beberapa variabel dan memudahkan di dalam pembuatan analisis-analisis yang mencakup hubungan antara variabel-variabel.

Sebuah matriks dinotasikan dengan simbol huruf besar seperti $A, X,$ atau Z dan sebagainya. Matriks mempunyai ukuran yang disebut ordo. Ordo matriks berbentuk mxn dengan m banyak baris dan n banyak kolom. Sebagai contoh matriks A dengan ordo mxn bisa ditulis sebagai berikut:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ a_{m1} & a_{m2} & \cdots & a_{mj} & \cdots & a_{mn} \end{bmatrix}$$

Keterangan :

$A = [a_{ij}], i = 1,2, \ldots, m$ dan $j = 1,2, \ldots, n$.

a_{ij} merupakan elemen matriks A dari baris *i* dan kolom *j, i* dan *j* dinamakan indeks (*subcript*), yaitu petunjuk letak (posisi) bagi setiap elemen. Elemen-elemen $a_{11}, a_{22}, \dots, a_{ij}, \dots, a_{mn}$ disebut diagonal pokok (*main diagonal*).

2.1.2 Jenis-jenis Matriks

Beberapa jenis matriks sebagai berikut:

1. Matriks kuadrat

   Matriks kuadrat adalah matriks yang memiliki baris dan kolom yang sama banyak. Dalam suatu matriks kuadrat, elemen-elemen $a_{11}, a_{22}, \dots, a_{nxn}$ disebut elemen diagonal utama.

$$A_{nxn} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

2. Matriks diagonal

   Matriks kuadrat $A = [a_{ij}]$ dinamakan matriks diagonal jika semua elemen di luar diagonal utama adalah nol, $a_{ij} = 0$ untuk $i \neq j$ dan paling tidak satu elemen pada diagonal pokok $a_{ij} \neq 0$ untuk $i = j$. Jumlah elemen-elemen diagonal utama suatu matriks kuadrat *A* disebut *trace A* ditulis *tr(A)*.

3. Matriks Identitas

   Matriks *A* disebut matriks identitas dan biasa diberi simbol *I*.

4. Matriks singular

   Matriks kuadrat $A = [a_{ij}]$ dikatakan singular jika semua elemen pada salah satu baris atau kolom adalah nol atau jika semua kofaktor dari elemen suatu baris atau kolom sama dengan nol. Untuk melihat kesingularan suatu matriks adalah dengan menghitung determinan matriks tersebut. Apabila determinannya sama dengan nol maka matriks tersebut singular.

5. Matriks orthogonal

   Matriks kuadrat $A = [a_{ij}]$ dikatakan dapat didiagonalisasi secara orthogonal jika terdapat matriks orthogonal *P* sehingga berlaku $P^{-1}AP = P'AP$. Matriks orthogonal didefinisikan sebagai matriks kuadrat yang inversnya sama dengan transposenya, sehingga $P^{-1} = P$ dan *P* adalah matriks orthogonal.

2.1.3 Operasi Matriks

Ada beberapa operasi dalam matriks, yaitu :

1. Perkalian matriks dengan skalar

   Jika $A = [a_{ij}]$ adalah matriks mxn dan k adalah suatu skalar, maka hasil kali *A* dengan *k* adalah $B = [b_{ij}]$ matriks *mxn* dengan $b_{ij} = ka_{ij}(1 \leq i \leq m, 1 \leq j \leq n)$.

2. Perkalian matriks dengan matriks

   Jika $A = [a_{ij}]$ dan $B = [b_{jk}]$ dengan $i = 1, 2, \dots, n$, $j = 1, 2, \dots, r$ dan $k = 1, 2, \dots, m$, maka perkalian matriks *A* dan *B* yang dinyatakan oleh $C = [c_{ij}]$ matriks ixj dengan $c_{ij} = \sum_{j=1}^{m} a_{ij}b_{jk}$ yang memenuhi syarat : banyak kolom *A* sama dengan banyak baris *B*.

3. Penjumlahan matriks

   Jika $A = [a_{ij}]$ adalah matriks *mxn* dan $B = [b_{ij}]$ adalah matriks *mxn* maka penjumlahan matriks *A* dan matriks *B* yang ditulis dengan $C = [c_{ij}]$ dengan:

   $c_{ij} = a_{ij} + b_{ij}\,; (i = 1,2, \dots, m; j = 1,2, \dots, n)$.

4. Transpose matriks

Jika $A = [a_{ij}]$ adalah matriks *mxn* maka matriks *nxm* dengan $A' = [a'_{ij}]$ dan $a'_{ij} = a_{ji}(1 \leq i \leq m, 1 \leq j \leq n$ disebut dengan transpose dari matriks *A*.

Matriks mxn yang umum dapat ditulis:

$$A_{mxn} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_{ij}]\,;\,;(i = 1,2,\dots,m; j = 1,2,\dots,n)$$

5. Invers matriks

Jika *A* matriks *nxn* disebut non singular apabila terdapat matriks *B* maka $AB = BA = I$. Matriks *B* disebut invers dari matriks *A*. Jika tidak terdapat matriks *B* maka matriks *A* disebut singular. Secara umum invers matriks *A* adalah:

$$A^{-1} = \frac{1}{\det(A)} Adj(A)$$

Sifat-sifat invers:

a. Jika *A* matriks non singular, maka A^{-1} adalah non singular dan $(A^{-1})^{-1} = A$.

b. Jika *A* dan *B* adalah matriks non singular, maka *AB* adalah non singular dan $(AB)^{-1} = B^{-1}A^{-1}$.

c. Jika *A* adalah matriks non singular, maka $(A^T)^{-1} = (A^{-1})^T$.

2.2 Program Linear

2.2.1 Pengertian Program Linear

Menurut Frederick S. Hiller dan Gerald J. Lieberman, pemrogram linear merupakan suatu model matematis untuk menggambarkan masalah yang dihadapi. Linear berarti bahwa semua fungsi matematis dalam model ini harus merupakan fungsi linear. Programming merupakan sinonim untuk kata perencanaan. Dengan demikian membuat rencana kegiatan—kegiatan untuk memperoleh hasil yang optimal, ialah suatu hasil untuk mencapai tujuan yang ditentukan dengan cara yang paling baik (sesuai dengan model matematis) di antara semua alternatif yang mungkin.

Model Pemrograman linear mempunyai tiga unsur utama, yaitu :

a) Variabel Keputusan adalah variabel persoalan yang akan mempengaruhi nilai tujuan yang hendak dicapai. Di dalam proses pemodelan, penemuan variabel keputusan tersebut harus dilakukan terlebih dahulu sebelum merumuskan fungsi tujuan dan kendalakendalanya.

b) Fungsi Tujuan. Dalam model pemrograman linear, tujuan yang hendak dicapai harus diwujudkan ke dalam sebuah fungsi matematika linear. Selanjutnya, fungsi ini dimaksimumkan atau diminumkan terhadap kendala-kendala yang ada. Beberapa contoh tujuan yang hendak dicapai di dalam pabrik manajemen adalah Pemaksimuman laba perusahaan, peminimuman biaya distribusi, dan lain sebagainya.

c) Kendala Kendala fungsional. Manajemen menghadapi berbagai kendala untuk mewujudkan tujuan-tujuannya.

2.2.2 Model Matematis

Secara umum model matematis untuk kondisi maksimasi dan minimasi terdapat perbedaan pada kendala. Untuk kasus maksimasi umumnya kendala berbentuk pertidaksamaan $\leq$, sedangkan kasus minimasi berbentuk pertidaksamaan $\geq$.

a. Kasus Maksimasi

Maksimum : $Z = C_1X_1 + C_2X_2 + \cdots + C_nX_n$

Kendala :

$a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n \leq b_1$

$a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n \leq b_2$

$\vdots$

$a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n \leq b_m$

$X_1, X_2, \dots, X_n \geq 0$

$b_1, b_2, \dots, b_m \geq 0$

b. Kasus Minimasi

Minimum : $Z = C_1X_1 + C_2X_2 + \cdots + C_nX_n$

Kendala :

$a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n \geq b_1$

$a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n \geq b_2$

$\vdots$

$a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n \geq b_m$

$X_1, X_2, \dots, X_n \geq 0$

$b_1, b_2, \dots, b_m \geq 0$

2.2.3 Asumsi-Asumsi dalam Program Linear

Menurut Frederick S. Hilter dan Gerald J. Lieberman, terdapat empat asumsi dalam program linear, yaitu :

a) Proporsionalitas, naik atau turunnya nilai Z dan penggunaan sumber daya yang tersedia akan berubah berbanding lurus dengan perubahan tingkat kegiatan (X).

b) Aditivitas, bahwa untuk setiap fungsi, nilai fungsi total dapat diperoleh dengan menjumlahkan kontribusi-kontribusi individual masing-masing kegiatan.
c) Divisibilitas, Kadang-kadang variabel-variabel keputusan yang dihasilkan oleh setiap kegiatan tidak selalu menghasilkan angka fisik yang bulat (integer) tetai juga dapat berupa bilangan pecahan (non-integer).
d) Kepastian, semua parameter model nilai-nilai (dalam program linear) merupakan konstanta-konstanta yang diketahui. Dalam praktek, asumsi ini jarang dipenuhi secara tepat. Model program linear biasanya dirumuskan untuk memilih tindakan dimasa yang akan datang, sedangkan kondisi yang akan datang itu sendiri membawa kepastian.

2.2.4 Langkah Pemodelan Program Linear

Dalam menformulasikan suatu masalah nyata ke dalam program linear, maka langkah berikut harus diperhatikan:

a) Memahami permasalahan.
b) Mengidentifikasi variabel-variabel keputusan.
c) Menyatakan fungsi tujuan sebagai kombinasi linear dari variabel-variabel keputusan.
d) Menyatakan kendala-kendala struktural sebagai kombinasi linear dari variabel-variabel keputusan.
e) Menyatakan kendala non negatif dari variabel-variabel keputusan.

2.3 Masalah Transportasi

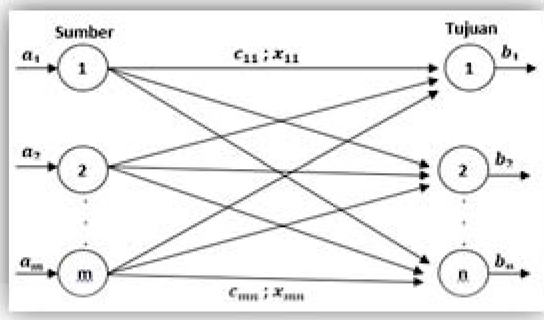
Masalah transportasi adalah bagian dari "*operation research*" yang membahas tentang minimasi biaya transportasi dari suatu tempat ke tempat lain.

Istilah transportasi atau distribusi terkandung makna bahwa mendistribusikan barang dari suatu tempat ke tempat atau beberapa tempat lain memerlukan alat dan biaya transportasi.

Permasalahan transportasi berkaitan dengan pendistribusian komoditas dari beberapa pusat penyedia, yang disebut sumber menuju ke beberapa pusat penerima yang disebut tujuan, dengan maksud untuk memperkecil total biaya distribusi (Hiller dan Lieberman, 2001:354).

Taha (1996:202) menyatakan bahwa model transportasi pada dasarnya merupakan sebuah program linear yang dapat dipecahkan dengan metode simpleks biasa. Namun mengingat metode simpleks kurang efisien dalam memecahkan masalah nyata yang mencakup banyak variabel dan kendala, sehingga digunakanlah metode transportasi yang dianggap lebih efisien dari pada metode simpleks.

Sehubungan dengan penerapan metode transportasi yang sudah banyak diterapkan dalam memecahkan persoalan transportasi, maka pada penelitian ini akan diterapkan Algoritma Affine Scaling yang juga diketahui lebih efisien dalam memecahkan masalah nyata yang mencakup banyak variabel dan kendala. Masalah transportasi dapat diselesaikan dengan algoritma Affine Scaling dengan memprosesnya terlebih dahulu ke dalam model transportasi dan formulasi program linier. Proses transportasi antara permintaan (*demand*) dan penawaran (*supply*) dapat dilihat pada Gambar 2.1.



Gambar 2.1 Transportasi Dari Sumber Ke Tujuan (Rangkuti, 2013).

Gambar 2.1 memperlihatkan sebuah model transportasi dari sebuah jaringan *m* sumber dan *n* tujuan. Sumber dan tujuan diwakili dengan node, dan rute pengiriman barang yang menghubungkan sumber ke tujuan diwakili dengan busur yaitu:

a) Masing-masing sumber mempunyai kapasitas $a_i, i = 1,2, \dots, m$.
b) Masing-masing tujuan mempunyai kapasitas $b_j, j = 1,2, \dots, n$.
c) x_{ij} merupakan jumlah satuan unit yang dikirim dari sumber *i* ke tujuan j.
d) c_{ij} merupakan ongkos pengiriman per unit dari sumber *i* ke tujuan j.

Secara matematis permasalahan transportasi dapat dimodelkan sebagai berikut:

Fungsi tujuan:

$$\text{Minimum } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}$$

dengan kendala:

$$\sum_{i=1}^{m} X_{ij} = a_i ; i = 1,2, \dots, m$$

$$\sum_{j=1}^{n} X_{ij} = b_j \; j = 1,2,\dots,n$$

Keterangan:

C_{ij} = biaya transportasi per unit barang dari sumber i ke tujuan j

X_{ij} = jumlah barang yang didistribusikan dari sumber i ke tujuan j

a_i = jumlah barang yang ditawarkan atau kapasitas dari sumber i

b_j = jumlah barang yang diminta atau dipesan oleh tujuan j

m = banyaknya sumber

n = banyaknya tujuan

Suatu masalah transportasi dikatakan seimbang apabila jumlah penawaran pada sumber i sama dengan jumlah permintaan pada tujuan j.

Dapat dituliskan:

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{n} b_j$$

Masalah transportasi dapat ditempatkan dalam suatu tabel khusus yang dinamakan tabel transportasi. Sumber ditulis dalam baris-baris dan tujuan dalam kolom-kolom. Dalam tabel transportasi terdapat $m\ x\ n$ kotak. Biaya transportasi per unit barang C_{ij} dicatat pada kotak kecil di bagian atas setiap kotak. Permintaan dari setiap tujuan terdapat pada baris paling bawah, sementara penawaran setiap sumber dicatat pada kolom palingkanan. Kotak pojok kiri bawah menunjukkan kenyataan bahwa penawaran atau *supply* (S_i) sama dengan permintaaan atau *demand* (d_i). Variabel X_{ij} pada setiap kotak menunjukkan jumlah barang yang diangkut dari sumber *i* ke tujuan *j*. Bentuk umum dari tabel transportasi dapat dilihat pada Tabel 2.1

Dari / Ke		Tujuan						*Supply*
		1	2	…	*j*	…	n	
Sumber	1	C_{11} X_{11}	C_{12} X_{12}		C_{1j} X_{1j}		C_{1n} X_{1n}	S_1
	2	C_{21} X_{21}	C_{22} X_{22}	…	C_{2j} X_{2j}		C_{2n} X_{2n}	S_2
	⋮	⋮	⋮		⋮		⋮	⋮
	i	C_{i1} X_{i1}	C_{i2} X_{i2}	…	C_{ij} X_{ij}	…	C_{in} X_{in}	S_i
	⋮	⋮	⋮		⋮		⋮	⋮
	m	C_{m1} X_{m1}	C_{m2} X_{m2}		C_{mj} X_{mj}		C_{mn} X_{mn}	S_m
Demand		D_1	D_2	…	D_j	…	D_n	$\sum S_i = \sum D_j$

Tabel 2.1 Tabel Transportasi

2.4 Algoritma Affine Scaling

2.4.1 Algoritma dasar

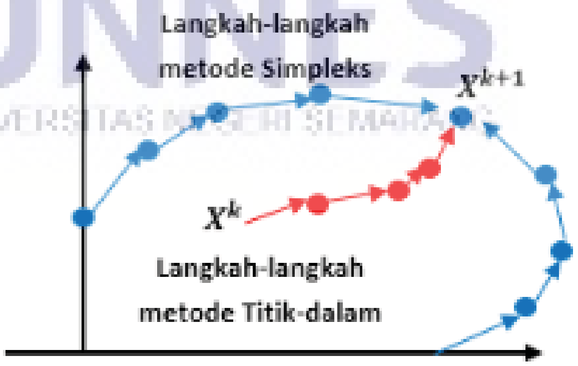
Algoritma Affine Scaling adalah sebuah metode titik interior. Secara umum, algoritma ini berbeda dengan metode simpleks dimana untuk mencapai titik optimal, Algoritma Affine Scaling bergerak di dalam daerah interior sementara metode simpleks bergerak pada titik ekstrimnya (Chong dan Ẑak, 2000).

Apabila program linear memuat variabel dan persamaan kendala yang banyak (lebih dari 100 persamaan kendala dan variabel) maka proses penyelesaian yang dilakukan dengan metode titik interior dapat lebih cepat dan efisien dibandingkan dengan metode simpleks (Avriel & Golany, (1996) dan Panik (1996)). Karena pada metode simpleks, apabila program linear memuat banyak variabel dan kendala, maka program linear tersebut juga akan memiliki banyak titik batas dan dibutuhkan proses yang panjang untuk mencapai titik optimal dibandingkan dengan metode titik interior.

Terdapat dua langkah yang diperlukan dari metode titik interior, yaitu:

1. Mencari arah fisibel yang memperbaiki nilai fungsi tujuan pada titik yang ditentukan dari setiap iterasi.
2. Menentukan *step length* yang berada pada daerah fisibel sesuai arah fisibel yang memperbaiki nilai fungsi tujuan.

Sebagai ilustrasi, perbedaan antara metode simpleks dan metode titik interior dapat dilihat pada Gambar 2.2.



Gambar 2.2. Perbandingan Metode Simpleks Dengan Metode Titik Interior (Retnojiwati, 2007).

Selanjutnya untuk memulai Algoritma Affine Scaling, akan diberikan masalah Program Linear

Minimalkan $z = c^T x$ (2.1)

dengan kendala $Ax = b$ (2.2)

$x \geq 0$,

dengan

A : Koefisien matriks kendala

b : Vektor suku tetap (batasan kendala)

c^T : Vektor ongkos

X : Kendala non negatif

Kendala-kendala fisibel (definisi) pada persamaan (2.5) adalah $Ax = b$ dan $x \geq 0$. Setelah bentuk masalah program linear sudah seperti pada persamaan (2.1) dan (2.2), langkah selanjutnya adalah mengandaikan bahwa terdapat titik fisibel $x^{(0)}$ yang memenuhi semua persamaan kendala. Titik baru $x^{(1)}$ (definisi) dapat diperoleh dengan mencari suatu arah $d^{(0)}$ yang menurunkan fungsi objektif. Dengan kata lain

$$x^{(1)} = x^{(0)} + \alpha_0 d^{(0)},$$

dengan

$x^{(0)}$: Penyelesaian fisibel pada iterasi ke-0

α_0 : *Step length* ($0 < \alpha < 1$)

$d^{(0)}$: Arah pencarian.

Berdasarkan pada metode gradien, digunakan gradien negatif pada fungsi objektif untuk mencari arah. Untuk masalah program linear, gradien negatif dari fungsi objektif adalah $-c$ (gradien). Dengan demikian, jika $d^{(0)} = -c$, maka titik $x^{(1)}$ mungkin tidak berada di dalam himpunan fisibel. Tetapi karena $x^{(1)}$ berada di dalam himpunan fisibel, maka $d^{(0)}$ harus merupakan sebuah vektor dalam nullspace A (nullspace artinya bahwa jika $x^{(1)}$ dan $x^{(0)}$ merupakan dua solusi fisibel pada persamaan (2.3) dan (2.4), yang jika $(x^{(1)} - x^{(0)})$ dikalikan dengan matriks Aakan menghasilkan vektor nol seperti pada persamaan (2.8)). Karena $x^{(0)}$ dan $x^{(1)}$ adalah fisibel, maka

$$Ax^{(0)} = b, \tag{2.3}$$

dan

$$Ax^{(1)} = b, \tag{2.4}$$

Dari persamaan (6) dan (7) diperoleh

$$Ax^{(0)} = Ax^{(1)}$$

$$Ax^{(1)} - Ax^{(0)} = 0$$

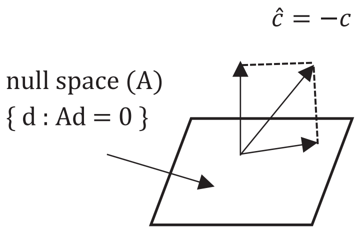
$$A(x^{(1)} - x^{(0)}) = 0$$

$$Ad^{(0)} = 0, \tag{2.5}$$

Selanjutnya, untuk memilih sebuah arah $d^{(0)}$ yang berada di nullspace A tetapi masih "dekat" ke $-c$, $-c$ secara ortogonal diproyeksikan ke atas nullspace A dan hasil proyeksi dijadikan sebagai $d^{(0)}$. Proyeksi ortogonal dari suatu vektor di atas nullspace A melibatkan perkalian dengan matriks P, yang disebut matriks proyeksi :

$$P = l_n - A^T(AA^T)^{-1}A. \tag{2.6}$$

Persamaan (2.6) diperoleh



Gambar 2.3 Arah Steepest Descent Yang Diproyeksikan (Shevade, 2012).

Dimisalkan bahwa

$$Ad = 0 \text{ dan } q = A^T\lambda.$$

Kemudian

$$\hat{c} = d + q$$

$$\hat{c} = d + A^T\lambda. \tag{2.7}$$

Masing-masing ruang pada persamaan (2.10) dikalikan dengan *A*, sehingga diperoleh

$$A\hat{c} = Ad + AA^T\lambda$$

$$A\hat{c} = 0 + AA^T\lambda$$

$$\lambda = (AA^T)^{-1}A\hat{c}. \tag{2.8}$$

Kemudian persamaan (2.8) disubstitusikan ke persamaan (2.7) diperoleh

$$\hat{c} = d + A^T(AA^T)^{-1}A\hat{c}$$

$$d = \hat{c} - A^T(AA^T)^{-1}A\hat{c}$$

$$d = \hat{c}(1 - A^T(AA^T)^{-1}A)$$

$$d = \underbrace{(1 - A^T(AA^T)^{-1}A)}_{P \text{ (Proyeksi Matriks)}}\hat{c}$$

$$d = P\hat{c}. \tag{2.9}$$

Persamaan (2.9) diposisikan sedemikian hingga agar benda di arah proyeksi orthogonal $-c$ di atas null space A yakni :

$$d^{(0)} = -Pc.$$

dengan $d^{(0)} = d$. Selanjutnya$APc = 0$ dapat dicek dengan mensubstitusikan $d^{(0)}$ ke persamaan (2.2) yakni :

$$A\left(x^{(1)} - x^{(0)}\right) = 0$$

$$A(-Pc) = 0$$

$$-APc = 0 \qquad \text{(masing-masing ruas dikalikan -1)}$$

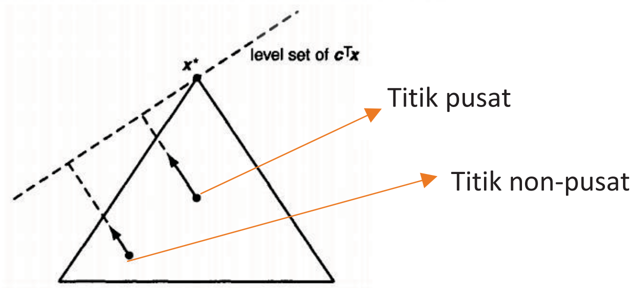
$$APc = 0.$$

Secara ringkas, diberikan titik $x^{(0)}$. Kemudian titik fisibel baru $x^{(1)}$ dapat diperoleh menggunakan rumus

$$x^{(1)} = x^{(0)} - \alpha_0 Pc. \tag{2.10}$$

Titik fisibel baru $x^{(1)}$ pada persamaan (2.10) menyatakan iterasi dari proyeksi gradien. Selanjutnya dipastikan bahwa titik $x^{(0)}$ yang dipilih dekat dengan pusat himpunan fisibel.

Berikut ini merupakan hasil langkah gradien yang diproyeksikan dari titik pusat dan non-pusat :



Gambar 2.4 Hasil Langkah Gradien Yang Diproyeksikan Dari Titik Pusat dan Non-Pusat (Chong & Żak, 2000).

Gambar 2.4 Memperlihatkan titik awal pusat dan non-pusat dalam mencapai titik optimal yaitu x^*. Dapat dilihat bahwa jika titik awal yang dipilih berada di pusat himpunan fisibel, maka langkah untuk menuju ke titik optimal akan semakin cepat. Demikian halnya jika titik awal yang dipilih jauh dari pusat, maka langkah untuk menuju ke titik optimal akan semakin lambat.

Misalkan dipilih suatu titik $x^{(0)}$ yang fisibel tetapi tidak berada dipusat. Maka langkah yang dapat dilaukan untuk membuat titiktersebut berada di pusat adalah mentransformasi titik$x^{(0)}$ menggunakan transformasi *Affine Scaling*. Secara sederhana, dimisalkan bahwa $A = [1,1,\dots,1]/n$ dan $b = [1]$. Dengan demikian dapat dilihat bahwa pusat dari himpunan fisibel ini adalah $\boldsymbol{e} = [1,\dots,1]^T$.

Untuk mentransformasikan $x^{(0)}$ ke pusat digunakan transformasi affine scaling yaitu :

$$\boldsymbol{e} = D_o^{-1}x^{(0)},$$

dengan D_o adalah matriks diagonal yang entri-entri diagonalnya merupakan komponen-komponen dari titik $x^{(0)}$, yakni :

$$D_o = diag\left[x_1^{(0)},\dots,x_n^{(0)}\right] = \begin{pmatrix} x_1^{(0)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_n^{(0)} \end{pmatrix}.$$

D_o adalah titik tunggal (*invertible*) karena telah diasumsikan bahwa $x^{(0)}$ adalah positif. A dan b juga perlu ditransformasi menggunakan transformasi Affine Scaling yang sama seperti di atas. Secara umum, mungkin tidak secara tepat berada di pusat himpunan fisibel, tetapi diharapkan bahwa titik yang ditransformasikan akan jadi "dekat" ke pusat. Minimal titik $\boldsymbol{e}$ berjarak sama dari batasan-batasan positif $\{x: x \geq 0\}$.

Sekali titik awal itu (atau dekat ke) pusat himpunan fisibel setelah melakukan transformasi Affine Scaling, maka dapat diproses lebih sebagaimana sebelumnya. Karena titik awal $x^{(0)}$ telah ditransformasi melalui pra-perkalian oleh D_0^{-1}, secara efektif hal itu merubah sistem kordinat, sehingga juga dibutuhkan masalah program linear awal di kordinat-kordinat baru.

Secara khusus, masalah program linear di kordinat-kordinat yang ditransformasi berbentuk :

$$\begin{aligned} &\text{Meminimalkan} && \bar{z} = \bar{c}_0^T \bar{x} \\ &\text{dengan kendala} && \bar{A}_0 \bar{x} = b \\ & && \bar{x} \geq 0. \end{aligned}$$

Dimisalkan bahwa

$$\bar{c}_0 = D_0 c$$

dan

$$\bar{A}_0 = A D_0.$$

Pada sistem kordinat baru ($\bar{x}$), proyeksi matriks dibentuk

$$\bar{P}_0 = l_n - \bar{A}_0^T (\bar{A}_0 \bar{A}_0^T)^{-1} \bar{A}_0,$$

Dan $\bar{d}^{(0)}$ diposisikan sedemikian hingga agar berada diarah proyeksi matriks dari $-\bar{c}_0$ pada null space $\bar{A}_0$:

$$\bar{d}^{(0)} = -\bar{P}_0 \bar{c}_0.$$

Selanjutnya $\bar{x}^{(1)}$ dihitung menggunakan

$$\bar{x}^{(1)} = \bar{x}^{(0)} - \alpha_0 \bar{P}_0 \bar{c}_0,$$

dengan

$$\bar{x}^{(0)} = D_0^{-1} x^{(0)}.$$

Untuk memperoleh titik baru, dilakukan transformasi yakni :

$$x^{(1)} = x^{(0)} + \alpha_0 d^{(0)}.$$

Secara iterasi, prosedur ini diulangi untuk menghasilkan urutan titik-titik $\{x^{(k)}\}$, dengan

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$$

dan

$$D_k = diag[x_1^{(k)}, \ldots, x_n^{(k)}$$

$$\bar{A}_k = AD_k$$

$$\bar{P}_k = l_n - \bar{A}_k^T (\bar{A}_k \bar{A}_k^T)^{-1} \bar{A}_k$$

$$\bar{x}^{(k)} = -D_k P_k D_k c.$$

Di setiap tahap dari algoritma ini, harus dipastikan bahwa titik $x^{(k)}$ bernilai positif. Catatan bahwa kondisi $Ax^{(k)} = b$ adalah terpenuhi secara otomatis di setiap tahap karena tergantung daricara memilih $d^{(k)}$. Meskipun demikian, juga dibutuhkan untuk menjamin bahwa $x_i^{(k)} > 0$ untuk $i = 1, \ldots, n$. Ini dapat dilakukan melalui pilihan tepat dari *step length* α_k.

Kriteria utama untuk memilih α_k adalah untuk membuat perhitungan lebih cepat sehingga sebagian komponen $x^{(k+1)}$ menjadi non-positif. Dipilih α_k sehingga

$$x_i^{(k+1)} = x^{(k)} + \alpha_k d_i^{(k)} > 0 \text{ untuk } i = 1, \ldots, n.$$

Untuk proses lebih lanjut, pertama didefinisikan

$$r_k = \min_{(i:d_i^{(k)}<0)} -\frac{x_i^{(k)}}{d_i^{(k)}},$$

r_k menyatakan nilai yang paling besar dari *step length* α_k dimana semua komponen dari $x^{(k+1)}$ non negatif. Untuk memastikan bahwa $x^{(k+1)}$ bernilai positif, digunakan *step length* daribentuk $\alpha_k = \alpha r_k$, dimana $\alpha \in (0,1)$. Nilai α untuk metode ini biasa $\alpha = 0,9$ atau $\alpha = 0,99$.

Tidak seperti dengan metode simpleks, Algoritma Affine Scaling tidak akan mencapai solusi optimal dalam beberapa tahap berhingga. Oleh karena itu, dibutuhkan kriteria untuk memberhentikan iterasi pada algoritma ini yaitu jika $c^T x^{(k+1)} > c^T x^{(k)}$ atau $|x^{(k+1)} s_k| < \varepsilon$ dan $\min(s_k) > -\varepsilon$, dimana ε adalah bilangan positif yang diberikan (Roos, Terlaky & Vial, 2005).

2.4.2 Metode Fase-Dua

Untuk mengimplementasikan Algoritma Affine Scaling yang dideskripsikan pada bagian 2.3.1, diperlukan solusi fisibel awal yang bernilai positif. Pada bagian ini, akan dibahas suatu metode untuk menemukan solusi awal. Setelah ditemukan solusi awal, berikutnya dapat dicari solusi optimal dari masalah yang ingin diselesaikan. Jadi metode Fase-Dua ini membutuhkan dua fase dimana pada fase I, akan ditentukan solusi fisibel awal yang positif. Selanjutnya pada fase II, hasil dari fase I digunakan untuk memulai Algoritma Affine Scaling dalam mencari solusi optimal.

Jadi Fase I dari Algoritma Affine Scaling adalah misalkan bahwa u adalah suatu vektor sembarang dengan semua komponennya bernilai positif, dengan

$$v = b - Au.$$

Dimisalkan bahwa $u = x^{(0)}$. Jika $v = 0$, maka u adalah titik fisibel yang positif. Dengan kata lain, jika $v \neq 0$, maka dibentuk masalah buatan sebagai berikut :

$$\begin{array}{ll} \text{Meminimalkan} & z = y \\ \text{dengan kendala} & [A, v\} \begin{bmatrix} x \\ y \end{bmatrix} = b \\ & \begin{bmatrix} x \\ y \end{bmatrix} \geq 0. \end{array}$$

dengan titik awal adalah

$$\begin{bmatrix} u \\ 1 \end{bmatrix}.$$

Dengan menggunakan titik $\begin{bmatrix} u \\ 1 \end{bmatrix}$ sebagai titik awal, maka Algoritma Affine Scaling dapat diterapkan untuk masalah buatan. Karena $y \geq 0$, maka Algoritma Affine Scaling akan dihentikan dengan beberapa solusi optimal.

2.4.3 Langkah-Langkah Algoritma Affine Scaling

Sebelum memulai Algoritma Affine Scaling, harus dipastikan bahwa bentuk masalah program linearnya adalah

$$\begin{array}{ll} \text{Meminimalkan} & z = \boldsymbol{c^T x} \\ \text{dengan kendala} & \boldsymbol{Ax = b} \\ & x \geq 0. \end{array}$$

Fase I :

Langkah 1 : Menentukan nilai *v*.

$$\boldsymbol{v} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(\mathbf{0})}.$$

Mengecek, apakah $\boldsymbol{v} = \mathbf{0}$?

1. Jika ya, maka $\boldsymbol{x}^{(\boldsymbol{k})} = \boldsymbol{x}^{(\mathbf{0})}$ kemudian lanjut ke Fase 2.
2. Jika tidak, masuk ke Langkah 2.

Langkah 2 : Membuat masalah buatan

$$\begin{aligned} &\text{Meminimalkan} && z = y \\ &\text{dengan kendala} && [A, v]\begin{bmatrix} x \\ y \end{bmatrix} = b \\ & && \begin{bmatrix} x \\ y \end{bmatrix} \geq 0. \end{aligned}$$

dengan solusi fisibel awal

$$\boldsymbol{x}^{(k)} = \begin{bmatrix} \boldsymbol{x}^{(\mathbf{0})} \\ 1 \end{bmatrix}.$$

Fase II :

Langkah 1 : Mendefinisikan matriks diagonal

$$X_k = diag\left(\boldsymbol{x}^{(\boldsymbol{k})}\right) = \begin{bmatrix} x_1^{(k)} & 0 & \cdots & 0 \\ 0 & x_2^{(k)} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & x_n^{(k)} \end{bmatrix}.$$

Langkah 2 : Menentukan A_k dan C_k

$$\boldsymbol{A_k} = \tilde{A}X_k$$

dan

$$\boldsymbol{C_k} = X_k\tilde{\boldsymbol{c}},$$

dengan $\widetilde{\boldsymbol{A}} = \boldsymbol{A}$ dan $\tilde{\boldsymbol{c}} = \boldsymbol{c}$.

Langkah 3 : Menentukan proyeksi matriks

$$P_k = l_n - A_k^T(A_k A_k^T)^{-1} A_k.$$

Langkah 4 : Menentukan arah pencarian

$$\boldsymbol{d_k} = -P_k \boldsymbol{C_k}.$$

Jika $\boldsymbol{dk} \ngtr \boldsymbol{0}$ artinya masalah tidak terbatas. Lanjut ke Langkah 5.

Langkah 5 : Menentukan vektor ongkos tereduksi

$$\boldsymbol{s_k} = -(X_k)^{-1} \boldsymbol{d_k}.$$

Langkah 6 : Menentukan *step length*

$$\alpha_k = \alpha \frac{1}{-\min(\boldsymbol{d_k})}$$

Langkah 7 : Menentukan titik baru

$$\boldsymbol{x^{(k+1)}} = \boldsymbol{x^{(k)}} + \alpha_k X_k \boldsymbol{d_k}.$$

Langkah 8 : Menguji Keoptimalan

Jika $\boldsymbol{c^T x^{(k+1)}} > \boldsymbol{c^T x^{(k)}}$ atau $\left|\left(\boldsymbol{x^{(k+1)}}\right)^{\boldsymbol{T}} \boldsymbol{s_k}\right| < \varepsilon$ dan $\min(\boldsymbol{s_k}) > -\varepsilon$, dengan ε adalah bilangan positif kecil yang diberikan, maka iterasi berhenti. Jika tidak, kembali ke Langkah 1 Fase II sampai kriteria pada Langkah 8 terpenuhi.

2.5 Analisis Sensitivitas

2.5.1 Pengertian Analisis Sensitivitas

Analisis sensitivitas merupakan suatu usaha untuk mempelajari nilai-nilai dari variabel-variabel pengambilan keputusan dalam suatu model matematika jika satu atau beberapa atau semua parameter model tersebut berubah atau menjelaskan pengaruh perubahan data terhadap penyelesaian optimal yang sudah ada. Tujuan utama dari analisis sensitivitas selain digunakan untuk pengecekan adalah untuk mengurangi perhitungan-perhitungan dan menghindari penghitungan ulang bila terjadi perubahan koefisien-koefisien pada model program linear setelah dicapai tahap optimal.

Analisis sensitivitas dapat dikelompokkan menjadi lima berdasarkan perubahan-perubahan parameter yang terjadi, yaitu:

1. Perubahan koefisien fungsi tujuan (***cj***),
2. Perubahan koefisien teknologi (***aij***) atau koefisien teknis,
3. Perubahan nilai ruas kanan fungsi kendala (***bi***),
4. Adanya tambahan fungsi kendala baru,
5. Adanya tambahan variabel pengambilan keputusan (***xj***) atau adanya penambahan kegiatan baru.

Perubahan yang dimaksud dalam penelitian ini adalah perubahan koefisien nilai ruas kanan fungsi kendala pada masalah program linear.

2.5.2 Perubahan Nilai Ruas Kanan Fungsi Kendala (b_i)

Perubahan nilai ruas kanan suatu fungsi kendala menunjukkan adanya pengetatan ataupun pelonggaran batasan tersebut. Pada penelitian ini perubahan konstanta ruas kanan terjadi apabila terjadi perubahan kapasitas permintaan dan persediaan yang ada. Misalkan solusi optimal dari suatu masalah telah didapatkan, lalu ubah b_i menjadi $b_i + \Delta b_i$ sehingga diperoleh masalah baru yang berbeda ruas kanannya dari yang semula. Perubahan pada b_i akan mempengaruhi nilai variabel basis pada solusi optimal dan optimalitas dari variabel basis tidak akan dipengaruhi dengan ketentuan bahwa perubahan b_i tidak membuat solusi basis menjadi tidak layak. Dengan demikian jika solusi basis yang baru adalah layak untuk konstanta ruas kanan yang baru, yaitu ketika

$$X'_B = B^{-1}(b + \Delta b) \geq 0 \tag{2.11}$$

maka basis optimal awal B juga optimal untuk masalah yang baru. Karena solusi optimal awalnya misalkan

$$X_B = \begin{bmatrix} x_1 \\ \dots \\ x_m \end{bmatrix} \tag{2.12}$$

dimana

$$X'_B = B^{-1}b \tag{2.13}$$

Maka Persamaan dapat diekspresikan sebagai berikut,

$$\widehat{x_i} = x_i + \sum_{j=1}^{m} \beta_{ij}\Delta b_j \ \geq 0 \tag{2.14}$$

dimana,

$$B^{-1} = [\beta_{ij}] \tag{2.15}$$

Dengan demikian basis optimal awal B tetap optimal dengan adanya perubahan pada b_i, Δb_i yang memenuhi persamaan (2.12). Perubahan nilai pada variabel basis optimal Δx_i berdasarkan perubahan pada b_i, diberikan sebagai berikut:

$$X'_B - X_b = \Delta X_B = B^{-1}\Delta b \tag{2.16}$$

Yaitu

$$\Delta x_i = \sum_{j=1}^{m} \beta_{ij}\Delta b_j, i = 1,2, \dots, m \tag{2.17}$$

Dengan demikian perubahan nilai fungsi objektif (ΔZ) berdasarkan perubahan Δb_i dapat diperoleh sebagai:

$$\Delta Z = c^T\Delta X_B = c^T B^{-1}\Delta b = \pi^T \Delta b = \sum_{j=1}^{m} \pi_j \Delta b_j \tag{2.18}$$

Berikut ini adalah contoh masalah program linear yang diselesaikan dengan Algoritma Affine Scaling dan Analisis Sensitivitas:

Contoh 2.1 :

Terdapat masalah program linear sebagai berikut :

$$\begin{aligned} &\text{Meminimalkan} && z = 3x_1 + 3x_2 \\ &\text{dengan kendala} && x_1 + 2x_2 \geq 20 && (*) \\ & && 2x_1 + x_2 \geq 16 && (**) \\ & && x \geq 0. \end{aligned}$$

Penyelesaian :

Sesuai dengan aturan metode Affine Scaling bahwa fungsi kendala harus $Ax = b$ dan $x \geq 0$, sehingga masalah program linear di atas perlu ditambahkan variabel *slack* (S_1 dan S_2) dan dapat ditulis sebagai berikut :

$$\begin{aligned} &\text{Meminimalkan} && z = 3x_1 + 3x_2 + 0S_1 + 0S_2 \\ &\text{dengan kendala} && x_1 + 2x_2 - S_1 = 20 \\ & && 2x_1 + x_2 - S_2 = 16 \\ & && x_1, x_2, S_1, S_2 \geq 0. \end{aligned}$$

dengan

$$A = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 20 \\ 16 \end{bmatrix}, c = [3 \quad 3 \quad 0 \quad 0]^T, \alpha = 0{,}99, \varepsilon = 10^{-5}.$$

<u>Fase I :</u>

Langkah 1 : Menentukan nilai awal

Dimisalkan : $x_1^{(0)} = 10$ dan $x_2^{(0)} = 10$.

Diperoleh :

$$x_1 + 2x_2 - S_1 = 30$$

$$S_1 = 10$$

$$2x_1 + x_2 - S_2 = 16$$

$$S_2 = 14$$

$$x^{(0)} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 14 \end{bmatrix}$$

Mengecek apakah $v = 0$ yaitu :

$$v = b - Ax^{(0)}$$

$$v = \begin{bmatrix} 20 \\ 16 \end{bmatrix} - \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 10 \\ 10 \\ 10 \\ 14 \end{bmatrix}$$

$$v = \begin{bmatrix} 20 \\ 16 \end{bmatrix} - \begin{bmatrix} 20 \\ 16 \end{bmatrix} = 0.$$

Karena $v = 0$, maka $x^{(1)} = x^{(0)}$, maka tidak melanjutkan ke Langkah 2. Langkah selanjutnya adalah masuk ke fase II.

Fase II :

Lnngkah 1 : Mendefinisikan matriks diagonal

$X_1 = diag(x^{(1)})$

$$X_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 14 \end{bmatrix}$$

Langkah 2 : Menentukan A_1 dan C_1

a) $A_1 = \hat{A}x_1$

$$A_1 = \begin{bmatrix} 10 & 20 & -10 & 0 \\ 20 & 10 & 0 & -14 \end{bmatrix}$$

dengan $\hat{A} = A$.

b) $C_1 = X_1\hat{c}$

$$C_1 = \begin{bmatrix} 30 \\ 30 \\ 0 \\ 0 \end{bmatrix}$$

Langkah 3 : Menentukan proyeksi matriks

$$P_1 = l_4 - A_1^T(A_1A_1^T)^{-1}A_1.$$

$$P_1 = \begin{bmatrix} 0,4193 & -0,2298 & -0,0404 & 0,4348 \\ -0,2298 & 0,3075 & 0,3851 & -0,1087 \\ -0,0404 & 0,3851 & 0,7298 & 0,2174 \\ 0,4348 & -0,1087 & 0,2174 & 0,5435 \end{bmatrix}$$

Langkah 4 : Menentukan arah pencarian

$$\boldsymbol{d_1} = -P_1\boldsymbol{C_1}.$$

$$d_1 = \begin{bmatrix} -5,6832 \\ -2,3292 \\ -10,3416 \\ -9,7826 \end{bmatrix}$$

Jika $\boldsymbol{d_1} \ngeq \boldsymbol{0}$ artinya masalah tidak terbatas. Lanjut ke Langkah 5.

Langkah 5 : Menentukan vektor ongkos tereduksi

$$\boldsymbol{s_1} = -(X_1)^{-1}\boldsymbol{d_1}.$$

$$s_1 = \begin{bmatrix} 0,5683 \\ 0,2329 \\ 1,0342 \\ 0,97826 \end{bmatrix}$$

Langkah 6 : Menentukan *step length*

$$\alpha_1 = \alpha\frac{1}{-\min(\boldsymbol{d_1})}$$

$$\alpha_1 = 0,0957$$

Langkah 7 : Menentukan titik baru

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} + \alpha_1X_1\boldsymbol{d_1}.$$

$$x^{(2)} = \begin{matrix} 4,5595 \\ 7,7703 \\ 0,1000 \\ 0,8892 \end{matrix}$$

Langkah 8 : Menguji Keoptimalan

Jika $c^T x^{(k+1)} > c^T x^{(k)}$ atau $|x^{(k+1)} s_k| < \varepsilon$ dan $\min(s_k) > -\varepsilon$.

Pertama,mengecek apakah

$$c^T x^{(2)} > c^T x^{(1)} \text{ ?}$$

$$36,9892 > 60 \text{ ?}$$

$$36,9892 < 60$$

Kemudian

$$|x^{(2)} s_1| < \varepsilon \text{ dan } \min(s_1) > -\varepsilon$$

a. $|x^{(2)} s_1| < \varepsilon$

$$5,374398 < 10^{-5} \text{ ?}$$

$$5,374398 > 10^{-5}.$$

b. $\min(s_1) > -\varepsilon$.

$$0,2329 > -10^{-5} \text{ ?}$$

$$0,2329 > -10^{-5}$$

Berdasarkan uraian pada Langkah 8, dapat dilihat bahwa titik baru ($x^{(2)}$) yang diperoleh pada Langkah 7 belum memenuhi kriteria, oleh karena itu proses diulangi dari Langkah 1 Fase II sampai kriteria pada Langkah 8 terpenuhi. Setelah melakukan iterasi sebanyak 6 kali, akhirnya diperoleh hasil iterasi pada Tabel 2.2.

Tabel 2.2 Hasil Iterasi Menggunakan Algoritma Affine Scaling

Iterasi	x_1	x_2	S_1	S_2	$\lvert x^{(k+)}s_k \rvert$	$\min(s_k)$
0	10	10	10	14	-	-
1	4,5595	7,7703	0,1	0,8892	5,374398108	0,23292
2	3,9764	8,0561	0,0885	0,0089	0,020424	-0,00324
3	4,005	7,9979	0,0009	0,008	0,000254659	-0,00006
4	3,9998	8,0005	0,0008	0,0001	0,000000152	0

Berdasarkan kriteria pada Langkah 8 Fase II, maka dapat dilihat pada Tabel 2.2 bahwa $\lvert x^{(k+1)}s_k \rvert < \varepsilon$ dan $\min(s_k) > -\varepsilon$ terpenuhi, oleh karena itu iterasi berhenti. Jadi

$$x_1 = 3{,}9998 \approx 4$$

$$x_2 = 8{,}0005 \approx 8$$

$$S_1 = 0{,}0008 \approx 0$$

$$S_2 = 0{,}0001 \approx 0.$$

dengan

$$z = 3x_1 + 3x_2$$

$$= 3(4) + 3(8)$$

$$= 36.$$

Nilai x_1 dan x_2 disubstitusikan ke persamaan kendala awal untuk menguji apakah variabel-variabel keputusan tersebut memenuhi syarat batas masalah, yaitu sebagai berikut :

$$x_1 + 2x_2 \geq 20 \qquad (*)$$

$$4 + 2(8) \geq 20$$

$$20 \geq 20$$

$$2x_1 + x_2 \geq 16 \qquad (**)$$

$$2(4) + 8 \geq 16$$

$$16 \geq 16$$

Dari pengecekan di atas, terlihat bahwa semua syarat batas pada persamaan kendala (*) dan (**) terpenuhi. Dengan demikian z disimpulkan sebagai solusi optimal dengan nilai $x_1 = 4$ dan $x_2 = 8$.

Contoh 2.2 :

Terdapat masalah program linear sebagai berikut :

Meminimalkan $z = 3x_1 + 3x_2$

dengan kendala $x_1 + 2x_2 \geq 24$ (*)

$2x_1 + x_2 \geq 18$ ` (**)

$x \geq 0.$

Penyelesaian :

Dapat dilihat pada contoh 2.1 dan 2.2 terjadi perubahan nilai ruas kanan fungsi kendala (b) tetapi koefisien fungsi tujuan (c_j) dan koefisien teknologi (a_{ij}) tidak mengalami perubahan. Selanjutnya, masalah program linear ini dapat diselesaikan dengan analisis sensitivitas sebagai berikut :

Langkah 1 : Tuliskan $A, b_{baru}, \Delta b, c\ dan\ X_B$. ($X_B$ merupakan variabel basis yang telah diperoleh dengan Algoritma Affine Scaling pada contoh 2.1)

$$A = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & -1 \end{bmatrix}, b_{baru} = \begin{bmatrix} 24 \\ 18 \end{bmatrix}, \Delta b = \begin{bmatrix} 4 \\ 2 \end{bmatrix}\ c = [3 \quad 3 \quad 0 \quad 0]^T, X_B = \begin{bmatrix} 4 \\ 8 \\ 0 \\ 0 \end{bmatrix}.$$

Langkah 2 : Menentukan matriks B^{-1}

Matriks B^{-1} merupakan matriks koefisien fungsi kendala dari variabel basis.

$$B^{-1} = [A]^{-1} = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} -0{,}1 & 0{,}4 \\ 0{,}4 & -0{,}1 \\ -0{,}3 & 0{,}2 \\ 0{,}2 & -0{,}3 \end{bmatrix}.$$

Langkah 3 : Menentukan nilai X'_B.

$$X'_B = X_B + B^{-1}\Delta b$$

$$X'_B = \begin{bmatrix} 4 \\ 8 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -0{,}1 & 0{,}4 \\ 0{,}4 & -0{,}1 \\ -0{,}3 & 0{,}2 \\ 0{,}2 & -0{,}3 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$X'_B = \begin{bmatrix} 4{,}4 \\ 9{,}4 \\ 0{,}8 \\ 0{,}2 \end{bmatrix}$$

Syarat solusi tetap optimal $X'_B \geq 0$.

Karena $X'_B \geq 0$, maka solusi tetap optimal.

Jadi

$$x_1 = 4{,}4 \approx 4$$

$$x_2 = 9{,}4 \approx 10$$

$$S_1 = 0{,}8 \approx 0$$

$$S_2 = 0{,}2 \approx 0.$$

dengan

$$\begin{aligned} z &= 3x_1 + 3x_2 \\ &= 3(4) + 3(10) \\ &= 42. \end{aligned}$$

Karena S_1 dan S_2 haruslah nol, maka nilai S_1 dibulatkan ke bawah menjadi nol dan nilai x_2 dibulatkan ke atas menjadi 10. Hal ini dilakukan agar nilai x_1 dan x_2 yang didapat memenuhi persamaan kendala awal yaitu sebagai berikut :

$$x_1 + 2x_2 \geq 24 \qquad (*)$$

$$4 + 2(10) \geq 24$$

$$24 \geq 24$$

$$2x_1 + x_2 \geq 18 \qquad (**)$$

$$2(4) + 10 \geq 18$$

$$18 \geq 18$$

Dari pengecekan di atas, terlihat bahwa semua syarat batas pada persamaan kendala (*) dan (**) terpenuhi. Dengan demikian z disimpulkan sebagai solusi optimal dengan nilai $x_1 = 4$ dan $x_2 = 10$.

2.6 Gambaran Umum UD. Tiga Putri

UD. Tiga Putri secara resmi berdiri pada tahun 2007 sebagai usaha keluarga. Pada tahun 2009, barulah UD. Tiga Putri ditetapkan statusnya secara yuridis. Kantor pusatnya di Desa Karangsono, Mranggen, Kabupaten Demak, dengan wilayah distribusi meliputi toko bangunan di daerah Kota Semarang.

UD. Tiga Putri merupakan salah satu perusahaan industri yang bergerak di bidang industri batu bata. Perusahaan ini memiliki aktivitas usaha yaitu menjual dan mendistribusikan batu bata kepada toko bangunan yang membutuhkan produk tersebut dalam kegiatan usahanya.

Batu bata yang telah diproduksi oleh beberapa lokasi pembakaran kemudian didistribusikan ke toko bangunan pelanggan tetap dari UD. Tiga Putri. Pengiriman produk batu bata dari UD. Tiga Putri menggunakan truk. Armada truk yang dimiliki oleh UD. Tiga Putri sebanyak 5 buah. Dalam satu bulan perusahaan melakukan pengiriman sebanyak 4 kali untuk setiap toko bangunan.

UD. Tiga Putri mempunyai langganan lokasi pembakaran batu bata di beberapa lokasi sebagai berikut:

1. Desa Karangsono, Kecamatan Mranggen, Kabupaten Demak.
2. Desa Karangboyo, Kecamatan Mranggen, Kabupaten Demak.
3. Desa Jawong, Kecamatan Mranggen, Kabupaten Demak.
4. Desa Kembangarum, Kecamatan Mranggen, Kabupaten Demak.
5. Desa Brawah, Kecamatan Mranggen, Kabupaten Demak.
6. Desa Wirosari, Kecamatan Wirosari, Kabupaten Grobogan.
7. Desa Ngawen, Kecamatan Ngawen, Kabupaten Blora.

Sedangkan untuk toko bangunan tetap dari UD. Tiga Putri yaitu:

1. Toko Bangunan Bintang Jaya, Pedurungan, Kota Semarang.
2. Toko Bangunan Bosan, Simongan, Kota Semarang.
3. Toko Bangunan Ada, Papandayan, Sampangan, Kota Semarang.
4. Toko Bangunan Makmur Jaya, Pucang Gading, Mranggen, Kabupaten Demak.
5. Toko Bangunan Rejeki, Banyumanik, Kota Semarang.
6. Toko Bangunan Sidodadi, Meteseh, Kota Semarang.
7. Toko Bangunan Acc, Jatingaleh, Kota Semarang.

UD. Tiga Putri awalnya hanya memiliki dua orang karyawan sebagai tenaga bongkar muat. Seiring berjalannya waktu, saat ini UD. Tiga Putri memilki jumlah karyawan sebanyak 24 orang dengan rincian 8 orang sopir dan 16 tenaga bongkar muat.

2.7 Pengantar Untuk Software *Matlab*

Matlab merupakan bahasa pemrograman *level* tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan (TheMathWorks, 2009). Matlab memungkinkan manipulasi matriks, pem-plot-an fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan pengantarmukaan dengan program dalam bahasa lainnya. Matlab adalah sebuah program untuk menganalisis dan mengkomputasi data numerik. Matlab merupakan suatu bahasa pemrograman matematika lanjutan, yang dibentuk dengan dasar pemikiran yang menggunakan sifat dan bentuk matriks.

Menurut Santoso (2008: 73) pemakaian *software* dalam menyelesaikan masalah optimasi sangatlah penting. Ini terutama bila sudah menyangkut persoalan skala besar dan melibatkan banyak iterasi dalam menemukan solusi optimal dari suatu persoalan. Persoalan sederhana mungkin bisa diselesaikan dengan suatu algoritma yang hanya memerlukan satu atau dua iterasi. Akan sangat membantu apabila algoritma atau metode yang dipakai bisa diprogramkan dengan bantuan *software*. Matlab merupakan salah satu *software* yang populer dan banyak dipakai untuk menyelesaikan masalah optimasi. *Matlab* mempunyai fungsi-fungsi yang sudah siap untuk menyelesaikan berbagai problem optimasi. Tugas kita sebagai *user* adalah memilih fungsi yang sesuai dengan persoalan yang kita punyai.

Kemudian kita perlu menuliskan persoalan optimasi kita dalam format Matlab. Di sisi lain kita juga bisa menuliskan sendiri fungsi/*script*/program yang belum tersedia dalam Matlab untuk menyelesaikan suatu persoalan optimasi. Dengan cara ini kita mempunyai keleluasaan untuk membuat tampilan, format input dan output dari *script* yang kita tulis. Masalah optimalisasi bisa kita kategorikan ke dalam dua kelas besar yaitu minimasi atau maksimasi. Minimasi berarti meminimalkan biaya produksi/transportasi agar didapat keuntungan yang optimal. Sedangkan maksimasi berarti meningkatkan laba perusahaan dengan memperkecil biaya produksi yang dikeluarkan.

2.7.1 Penyelesaian Persoalan Progrma Linear dengan Algoritma Affine Scaling menggunakan *Software Matlab*

Dengan menggunakan Contoh Soal 2.1, dituliskan kembali bentuk standar masalah program linear sebagai berikut :

Meminimalkan $z = 3x_1 + 3x_2 + 0S_1 + 0S_2$

dengan kendala $x_1 + 2x_2 - S_1 = 20$

$2x_1 + x_2 - S_2 = 16$

$x_1, x_2, S_1, S_2 \geq 0.$

dengan

$$A = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 20 \\ 16 \end{bmatrix}, c = [3 \quad 3 \quad 0 \quad 0]^T, \alpha = 0{,}99, \varepsilon = 10^{-5}.$$

Listing Program

```
function affine
clc;
clear;

x=input('masukan titik-dalam x=');
c=input('c=');
A=input('A=');
e=input('e=');
epsilon=input('epsilon=');

X=diag(x);
```

```
w=inv(A*X*X*A')*A*X*X*c;
r=c-A'*w;
p=e'*X*r;
d=-X*r;
norm_d=sqrt(d'*d);
d_min=min(d);
alpa_maks=min(0.99/-d_min);
x_baru=x+alpa_maks*X*d;


fprintf('\n w =');
fprintf('\n %3.4f',w);

fprintf('\n\n r =');
fprintf('\n %3.4f',r);

fprintf('\n\n p =');
fprintf('\n %3.4f',p);

fprintf('\n\n d =');
fprintf('\n %3.4f',d);

fprintf('\n\n d min=');
fprintf('\n %3.4f',d_min);

fprintf('\n\n norm_d =%3.4f',norm_d);

fprintf('\n\n alpa maks =%3.4f',alpa_maks);

fprintf('\n\n x_baru =');
fprintf('\n %3.4f',x_baru);


fprintf('\n------------------------');

while(r<0|p>epsilon);
   x=x_baru
   X=diag(x);
   w=inv(A*X*X*A')*A*X*X*c;
   r=c-A'*w;
   p=e'*X*r;
   d=-X*r;
   norm_d=sqrt(d'*d);
   d_min=min(d);
   alpa_maks=min(0.99/-d_min);
   x_baru=x+alpa_maks*X*d;
   z=c'*x_baru;
   fprintf('\n w =');
   fprintf('\n %3.4f',w);

   fprintf('\n\n r =');
   fprintf('\n %3.4f',r);
```

```
    fprintf('\n\n p =');
    fprintf('\n %3.4f',p);

    fprintf('\n\n d =');
    fprintf('\n %3.4f',d);

    fprintf('\n\n d min=');
    fprintf('\n %3.4f',d_min);

    fprintf('\n\n norm_d =%3.4f',norm_d);

    fprintf('\n\n alpa maks =%3.4f',alpa_maks);

    fprintf('\n\n x baru =');
    fprintf('\n %3.4f',x_baru);

    fprintf('\n------------------------');


    fprintf('\n\n f =%3.4f',z);
end
```

Output

```
masukan titik-dalam x=[10;10;10;14]
c=[3;3;0;0]
A=[1 2 -1 0;2 1 0 -1]
e=[1;1;1;1]

 x baru =
 4.0000
 8.0000
 0.0000
 0.0000
------------------------

 z =36.0000>>
```

Jadi, didapat titik optimumnya, yaitu $x = \begin{bmatrix} 4 \\ 8 \\ 0 \\ 0 \end{bmatrix}$

Nilai $z = 36$.

2.7.2 Penyelesaian Persoalan Program Linear dengan Analisis Sensitivitas menggunakan *Software Matlab*

Dengan menggunakan Contoh Soal 2.2, dituliskan kembali bentuk standar masalah program linear sebagai berikut :

Meminimalkan $z = 3x_1 + 3x_2 + 0S_1 + 0S_2$

dengan kendala $x_1 + 2x_2 - S_1 = 24$

$2x_1 + x_2 - S_2 = 18$

$x_1, x_2, S_1, S_2 \geq 0.$

dengan

$$A = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & -1 \end{bmatrix}, b_{baru} = \begin{bmatrix} 24 \\ 18 \end{bmatrix}, \Delta b = \begin{bmatrix} 4 \\ 2 \end{bmatrix} \; c = [3 \quad 3 \quad 0 \quad 0]^T, X_B = \begin{bmatrix} 4 \\ 8 \\ 0 \\ 0 \end{bmatrix}.$$

Listing Program

```
function sensitivity
clc;
clear;

x=input('masukkan x_b=');
a=input('masukkan a=');
b_lama=input('masukkan b_lama=');
b_baru=input('masukkan b_baru=');
c=input('masukkan c=');

delta_b=b_baru-b_lama;
b_invers=pinv(a);
v=b_invers*delta_b;
x_b=x+v;
z=c'*x_b;

fprintf('\n delta_b =');
fprintf('\n %3.4f',delta_b);

fprintf('\n\n b_invers =');
fprintf('\n %3.4f',b_invers);

fprintf('\n\n x_b=');
fprintf('\n %3.4f',x_b);
```

```
fprintf('\n------------------------');

fprintf('\n\n z=');
fprintf('\n %3.4f',z);

end
```

Output

```
masukkan x_b=[4;8;0;0]
masukkan a=[1 2 -1 0;2 1 0 -1]
masukkan b_lama=[20;16]
masukkan b_baru=[24;18]
masukkan c=[3;3;0;0]

 x_b=
 4.4000
 9.4000
 0.8000
 0.2000
------------------------>>
z=
 41.4000
```

Hasil yang didapat sama seperti yang sudah dihitung pada Contoh 2.2.

Jadi

$$x_1 = 4,4 \approx 4$$

$$x_2 = 9,4 \approx 10$$

$$S_1 = 0,8 \approx 0$$

$$S_2 = 0,2 \approx 0.$$

dengan

$$\begin{aligned} z &= 3x_1 + 3x_2 \\ &= 3(4) + 3(10) \\ &= 42. \end{aligned}$$

BAB V

PENUTUP

5.1 Simpulan

Dari pembahasan yang telah dilakukan, diperoleh simpulan sebagai berikut :

1. Model transportasi UD. Tiga Putri menggunakan program linear sebagai berikut :

   Meminimalkan $z = 24x_{11} + 27x_{12} + 28x_{13} + 24x_{14} + 30x_{15} + 32x_{16} + 27x_{17} + 25x_{21} + 28x_{22} + 29x_{23} + 25x_{24} + 31x_{25} + 33x_{26} + 28x_{27} + 23x_{31} + 26x_{32} + 27x_{33} + 23x_{34} + 29x_{35} + 31x_{36} + 26x_{37} + 23x_{41} + 26x_{42} + 27x_{43} + 23x_{44} + 29x_{45} + 31x_{46} + 26x_{47} + 23x_{51} + 26x_{52} + 27x_{53} + 23x_{54} + 29x_{55} + 31x_{56} + 26x_{57} + 38x_{61} + 41x_{62} + 41x_{63} + 38x_{64} + 44x_{65} + 45x_{66} + 41x_{67} + 60x_{71} + 63x_{72} + 64x_{73} + 60x_{74} + 66x_{75} + 68x_{76} + 63x_{77}$

   dengan kendala

   $$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} = 90000$$

   $$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} = 80000$$

   $$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} = 75000$$

   $$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} = 70000$$

   $$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} = 77000$$

   $$x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{66} + x_{67} = 65000$$

$$x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} + x_{77} = 60000$$

$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} = 72000$$

$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} = 60000$$

$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} + x_{73} = 64000$$

$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} + x_{74} = 84000$$

$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} + x_{75} = 80000$$

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} + x_{76} = 75000$$

$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{67} + x_{77} = 82000$$

$$x_{ij} \geq 0; i = 1,2,3,4,5,6,7 \text{ dan } j = 1,2,3,4,5,6,7.$$

2. Biaya transportasi UD. Tiga Putri pada bulan April 2017 dengan menggunakan Algoritma Affine Scaling adalah sebesar Rp. 17.033.000,-. Sedangkan biaya transportasi UD. Tiga Putri sebelum menerapkan Algoritma Affine Scaling adalah sebesar Rp. 17.900.000,-. Dengan demikian, dapat dikatakan bahwa terjadi penurunan sebesar Rp. 867.000,- artinya dapat meningkatkan laba perusahaan sebesar 4,84 %.
3. Biaya transportasi UD. Tiga Putri pada bulan Mei 2017 dengan menggunakan analisis sensitivitas adalah sebesar Rp. 17.187.000,-. Hasil ini sudah optimal karena kenaikan jumlah permintaan pada bulan Mei dengan diikuti kenaikan biaya transportasi tetapi tidak melebihi biaya transportasi yang telah dikeluarkan UD. Tiga Putri pada bulan April 2017. Hal ini jelas menunjukkan bahwa terjadi peningkatan laba perusahaan.

5.2 Saran

Dari penelitian yang telah dilakukan, maka saran yang sapat diberikan adalah :

1. Sebelum UD. Tiga Putri melakukan pendistribusian batu bata ke toko bangunan, sebaiknya persediaan batu bata pada lokasi pembakaran diperhatikan terlebih pada lokasi pembakaran dengan ongkos pengiriman yang murah agar dapat meminimalkan biaya transportasi dan meningkatkan laba perusahaan.
2. Diharapkan perusahaan UD. Tiga Putri dapat menerapkan Algoritma Affine Scaling dan Analisis Sensitivitas ini sebagai bahan pertimbangan dalam mengambil kebijakan dalam proses distribusi.
3. Dalam penelitian ini, telah digunakan salah satu dari metode titik interior dan analisis sensitivitas yaitu Algoritma Affine Scaling dan analisis sensitivitas dengan parameter perubahan nilai ruas kanan fungsi kendala. Sebagaimana diketahui bahwa metode titik interior terdiri dari 4 metode dan analisis sensitivitas terdiri dari 5 parameter, oleh karena itu untuk penelitian selanjutnya dengan inti permasalahan yang sama, disarankan untuk menggunakan algortima dan analisis sensitivitas dengan parameter yang lain.

DAFTAR PUSTAKA

Anton, Howard. 1995. *Aljabar Linear Elementer.* Jakarta: Penerbit Erlangga.

Chong, E.K.P. and H.Żak. 2000. *An Introduction to Optimization* (2^{nd} ed.). A Wiley-Interscience Publiction : John Wiley & Sons, Inc.

Dewi, I.K, Diah C., & Endang S. 2016. Masalah Penjualan Listrik dan Analisis Sensitivitas Menggunakan Pemrograman Linear. *Prosiding Seminar Nasional MIPA 2016. ISBN : 978-602-72216-1-1.* Bandung: Universitas Padjajaran.

Hiller, F.S. & G.J. Lieberman. 2008. *Introduction To Operations Research*. Amerika Serikat: Stanford University.

Ibnas, R. 2014. Optimalisasi Kasus Pemrograman Linear Dengan Metode Grafik dan Simpleks. *Jurnal MSA* Vol. 2 No. 1. Tersedia di journal.uin-alauddin.ac.id/index.php/msa/article/download/573/569 [diakses 23-03-2017].

Indriani, D.R., Suyitno, H., & Mashuri. 2013. Analisis Metode Karmarkar Untuk Menyelesaikan Masalah Program Linier. *Jurnal Mipa*, 36(1): 98-106.

Kuswardi, Y. 2012. *Program Linier.* Surakarta: UPT Penerbitan dan Percetakan UNS.

Mulyono, S. 2004. *Riset Operasi.* Jakarta: Fakultas Ekonomi Universitas Indonesia.

Pan, P. 2013. An Affine Scaling Pivot Algorithm For Linear Programming. A Journal of Mathematical Programming and Operations Research Vol. 62. Tersedia di www.tandfonline.com/doi/abs/10.1080/02331934.2011.606576 [diakses 22-01-2017].

Paseru, K. 2016. Penerepan Algoritma Affine Scaling Untuk Meminimalkan Biaya Transportasi. *Jurnal Matematika dan Ilmu Pengetahuan Alam.* Tersedia di *repository.unhas.ac.id/handle/123456789/19926 [diakses 26-01-2017]*

Retnojiwati, A. 2007. *Metode Primal Affine-Skaling Untuk Masalah Program Linier.* Skripsi. Yogyakarta, Indonesia.

Roos, C., T. Terlaky, and J.-Ph. Vial. 2005. *Interior Point Methods for Linear Optimization.* Springer, 2nd edition, Netherlands.

Santosa, B. 2008. *Matlab Untuk Statistika & Teknik Optimasi*. *Cetakan 1.* Yogyakarta: Graha Ilmu.

Shevade, Shirish. K. 2012. *Numerical Optimization.* Banglore, India.

Shintya Dewi, AA. 2014. Analisis Sensitivitas Dalam Optimalisasi Keuntungan Produksi Busana Dengan Metode Simpleks. Jurnal Matematika Vol. 4 No. 2., Desember 2014. ISSN : 1693-1394. Tersedia di ojs.unud.ac.id/index.php/jmat/article/viewFile/12553/8648 [diakses 20-01-2017].

Supranto, J. 2013. *Riset Operasi Untuk Pengambilan Keputusan.* PT Raja Grafindo. Jakarta.

Susanta, B. (1994). *Program Linear*. Yogyakarta: Departemen Pendidikan dan Kebudayaan Direktorat Jendral Pendidikan Tinggi.

Suyitno, H. 2014. *Program Linier*. Semarang: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.

Utomo, MIB. 2013. Model Optimasi Laba Dengan Algoritma Affine Scaling. Jurnal Mahasiswa Matematika Vol. 1 No.3. Tersedia di matematika.studentjournal.ub.ac.id/index.php/matematika/article/view/54 [diakses 22-01-2017].

Vanderbei, R.J. 1996. Linier Programming : Foundations and Extensions. Princeton, USA.